

Lucent Technologies
Bell Labs Innovations



Conversant

VIS V5.0 Interface (CLEO)
LINKix Documentation Set

585-350-912

Addenda to Release Notes

LINKix 3270

Read this section first!

Although the greatest care and attention to detail go into the preparation of the documentation, the following corrections and amendments should be noted. The information that follows supersedes existing documentation or provides supplemental information that was not originally included in the documentation.

Please, read all of this material before you install or use the product.

Technical Support

If you still have questions about installing and using this product after reading these Release Notes, the installation instructions and the troubleshooting information, contact Interface Systems Technical Support between the hours of 8:30 A.M. and 5:00 P.M. (EST/EDT) at: (313) 769-5900.



Copyright © 1995 CLEO Communications
A Division of Interface Systems, Inc.

February 8, 1995

Changes in This Release

This release of the LINKix 3270 feature package contains the following changes.

Installation Fails if ar is Not Present

Description: If the UNIX *ar* command is not available at installation time, the installation routine exits after removing the product. Typically, *ar* is installed on SCO with the development system, and on UnixWare it is part of the Personal Edition. Not all customers have *ar* available on their systems.

Correction: If the UNIX *ar* command is not available at installation time, the installation does not exit. Instead an error message is printed telling the user that the *ar* command failed and that the library *libhapi.o* has been copied into */usr/lib/linkix*. The installation then proceeds and fully installs the product.

If HLLAPI support is needed, you must archive *libhapi.o* into *libcom.a* in */usr/lib* using a copy of *ar* provided with the operating system or a compatible shareware/freeware utility.

Slow lp Printing

Description: Under certain circumstances the 3270 emulator *com3270* would exhibit very slow *lp* printing. After starting *com3270* the first one or two print jobs get spooled out to the printer in a timely fashion, however, subsequent print jobs would take a long time to print. The slow *lp* printing was more likely to occur with slow lines.

Correction: The fix corrects a problem in *com3270* which caused unnecessary calls to *wait ()* to be executed.

HLLAPI CONNECT Hangs When File Transfer in Progress

Description: The HLLAPI CONNECT verb hangs on the semaphore while a SEND FILE or RECEIVE FILE verb is in process on the same session id.

Correction: The fix causes the CONNECT verb to return HARC_UNAVAILABLE to the application, instead of making the application wait until the SEND FILE or

RECEIVE FILE completes. The application should retry the CONNECT verb later if HARC_UNAVAILABLE is returned.

HLLAPI NOTIFY EVENT Not Properly Cancelled

Description: The HLLAPI NOTIFY EVENT verb was not being properly cancelled.

Correction: A HLLAPI NOTIFY EVENT verb is now cancelled by the RESET SYSTEM verb and when the HLLAPI application terminates.

HLLAPI Program Not Notified of Event

Description: While a NOTIFY EVENT verb was in effect, the HLLAPI application was not informed of a change in the status line during the login sequence when it should have been.

Correction: This problem has been corrected.

Addenda to Release Notes

LINKix 3.0.2.X

Read this section first!

Although the greatest care and attention to detail go into the preparation of the documentation, the following corrections and amendments should be noted. The information that follows supersedes existing documentation or provides supplemental information that was not originally included in the documentation.

Please, read all of this material before you install or use the product.

Technical Support

If you still have questions about installing and using this product after reading these Release Notes, the installation instructions and the troubleshooting information, contact Interface Systems Technical Support between the hours of 8:30 A.M. and 5:00 P.M. (EST/EDT) at: (313) 769-5900.



© 1994 Cleo Communications
A Division of Interface Systems, Inc.

September 7, 1994

Trademark Acknowledgments

Cleo Communications has made every effort to accurately acknowledge all trademarks that appear in the documentation that follows. Cleo Communications, however, cannot attest to the accuracy of this information.

The trademarks listed below were obtained from a variety of sources.

LINKix™ is a trademark and the Cleo logo is a registered trademark of Cleo Communications.

IBM®, IBM PC®, IBM PC/XT®, IBM PC/AT®, Personal System/2®, PS/2®, OS/2®, Micro Channel® and RISC System/6000® are registered trademarks of International Business Machines Corporation.

ShortCUT™ is a trademark and ICOT® is a registered trademark of ICOT Corporation.

Microsoft® is a registered trademark of Microsoft Corporation.

Novell® and NetWare® are registered trademarks of Novell, Inc.

Open Desktop®, SCO®, SCOTerm® and The Santa Cruz Operation® are registered trademarks of The Santa Cruz Operation, Inc.

UNIX® is a registered trademark and UnixWare™ is a trademark of UNIX System Laboratories, Inc., a wholly-owned subsidiary of Novell, Inc.

Xsight® is a registered trademark of Locus Computing Corporation.

All other trademarks are the property of their respective owners.

Known Problems and Incompatibilities

The following information details problems and incompatibilities with other hardware and software products and any known work arounds to these problems.

PUINFO Does Not Prompt to Redial Modem

If you execute the *puinfo* utility with a modem, the first execution goes correctly. Subsequent executions, however, do not prompt you to dial.

Workaround: After *puinfo* displays the prompt “Opening port 1. Please assure that the cable is connected to the port so the open will complete,” dial the number and the call will go through.

No Warning When Overwriting Defined Keystroke

When assigning a keystroke record sequence to a PF key that is already in use, no warning beep is issued and the previous sequence is overwritten.

Workaround: Be careful not to enter a PF key which has a sequence associated with it, unless you want to replace that sequence.

Color Support in SCO OPEN DESKTOP

When running the management utilities, *commanage* and *combrowse*, in SCO OPEN DESKTOP, the default colors may make the screen unreadable (or just ugly).

Workaround: This problem may be corrected by using scripts which we have provided. For *commanage*, use the *cmanage* script. For *combrowse*, use the *cbrowse* script.

If the script does not correct the problem, you should check your SCO OPEN DESKTOP *Graphical Environment Administrator's Guide* for details on changing video system settings (see the section on running **mkdev graphics**).

LINKix 3270 Color Support on SCO OPEN DESKTOP

If LINKix 3270 returns the error message:

You have loaded a style file for a color terminal, but this terminal does not support color. LINKix will use monochrome terminal settings.

when you are using a color monitor, you should check your SCO OPEN DESKTOP *Graphical Environment Administrator's Guide* for details unchanging video system settings (see the section on running **mkdev graphics**).

Token Ring: Disconnected Cable Causes Problems

After further investigation, we've discovered that a disconnected Token Ring cable will require the following actions to reactivate the connection:

- ◆ Unixware 1.1.0: You must reboot the system
- ◆ Unixware 1.1.1: ??? (not tested)
- ◆ Unixware 1.1.2: You must stop and restart the Token Ring link
- ◆ SCO UNIX (ODT 3.0) with Token Ring LLI driver version 3.1.lh: You must reboot the system

Best way to avoid problems with a disconnected Token Ring cable: Before starting LINKix, make sure that the Token Ring cable is properly attached to the system, and that it is secure so it won't accidentally be disconnected.

General Notes

Token Ring Driver Requirements for SCO UNIX

If you intend on using a Token Ring communications adapter board on a machine that's running SCO UNIX, make sure that the SCO UNIX Token Ring LLI driver is one of the following versions: 3.1.0j, 3.1.0k, 3.1.1h.

The LINKix Token Ring link level package will only install if one of these versions is present.

Note: The Token Ring driver patch, which is applied during installation, only activates SNA source routing for a machine with a local SAP of 04 (this is the local SAP address in the default configuration file).

Note: The SCO 3.3.0 LLI driver does not support SNA source routing and cannot be used with LINKix. You must use one of the earlier drivers (which are properly patched at install time).

Release Notes

LINKix 3.0.2.0

Read this section first!

Although the greatest care and attention to detail go into the preparation of the documentation, the following corrections and amendments should be noted. The information that follows supersedes existing documentation or provides supplemental information that was not originally included in the documentation.

Please, read all of this material before you install or use the product.

Technical Support

If you still have questions about installing and using this product after reading these Release Notes, the installation instructions and the troubleshooting information contact Interface Systems Technical Support between the hours of 8:30 A.M. and 5:30 P.M. (EST/EDT) at: (313) 769-5900.



INTERFACE
S Y S T E M S . I N C .

© 1994 Cleo Communications
A Division of Interface Systems, Inc.

July 1, 1994

Trademark Acknowledgments

Cleo Communications has made every effort to accurately acknowledge all trademarks that appear in the documentation that follows. Cleo Communications, however, cannot attest to the accuracy of this information.

The trademarks listed below were obtained from a variety of sources.

LINKix™ is a trademark and the Cleo logo is a registered trademark of Cleo Communications.

IBM®, IBM PC®, IBM PC/XT®, IBM PC/AT®, Personal System/2®, PS/2®, OS/2®, Micro Channel® and RISC System/6000® are registered trademarks of International Business Machines Corporation.

ShortCUT™ is a trademark and ICOT® is a registered trademark of ICOT Corporation.

Microsoft® is a registered trademark of Microsoft Corporation.

Novell® and NetWare® are registered trademarks of Novell, Inc.

Open Desktop®, SCO®, SCOTerm® and The Santa Cruz Operation® are registered trademarks of The Santa Cruz Operation, Inc.

UNIX® is a registered trademark and UnixWare™ is a trademark of UNIX System Laboratories, Inc., a wholly-owned subsidiary of Novell, Inc.

Xsight® is a registered trademark of Locus Computing Corporation.

All other trademarks are the property of their respective owners.

Known Problems and Incompatibilities

The following information details problems and incompatibilities with other hardware and software products and any known work arounds to these problems.

Token Ring: Disconnected Cable Causes Problems

We've discovered that a disconnected Token Ring cable will cause the LINKix system to respond unpredictably. In most instances, the system has to be rebooted if this problem occurs.

Workaround: Before starting LJNKix, make sure that the Token Ring cable is properly attached to the system, and that it is secure so it won't accidentally be disconnected.

3770: No Error if Spooler Exhausts Available Space

We've discovered when a job is spooled to a RJE workstation and there is not sufficient disk space for the file to be spooled, the spooler will not report a failure. This causes a problem when trying to run the work-station. To correct this problem, go into the spool directory under the /usr/lib/linkix/rje/<wkstaname>. Find the "STA_..." file that has a length of zero and the "SPL_..." file that has the same ending letters as the STA file. Both files must be removed.

Workaround: Check to ensure that the /usr file system has sufficient space to spool the job(s).

BackTab (Shift-Tab) Doesn't Work

The **BACKTAB** key does not move the cursor backward through fields in the LINKix interface, it moves forward instead. Also, because the 3270 BACK TAB key is mapped to **BACKTAB**, BACK TAB does not work.

Workaround: In the LINKix interface screens, use **CTRL+B** to move backward. In 3270, remap the BACK TAB key to **CTRL+B** and use that key combination instead.

Coprocessor Does Not Support Large Frame Sizes

The LINKix coprocessor software (linkix_coproc) does not currently support SDLC frame sizes larger than 265 (256+9).

Workaround: If you are using this product, do not change the default LINKix SDLC frame size and make sure that the MAXDATA value at the host is no larger than 265.

Screen Redraw When Returning from Temporary Shell

When returning from a temporary UNIX shell (TEMPEXIT), the host screen is redrawn over the existing screen.

Workaround: Press Ctrl+R (Redraw) after returning from the temporary shell.

Error Messages Improperly Displayed

Error messages which appear while 3270 emulation is running may not be properly displayed — they appear “folded.”

Workaround: Press Ctrl+R once or twice and the screen will redraw properly. You can also use the -s option when starting the daemon (*comstart*) to prevent messages from appearing on the screen.

RJE Console Program Command Line

When exiting from a help screen for the RJE Console program, *snarjecon*, the screen is not drawn correctly (a dashed line appears instead).

Workaround: Type in anything and the command line returns.

Connection Failure Message

The following error message may be reported shortly after the control daemon is started (*comstart*):

Connection connection-name failed - no link service available

Workaround: If you use the -s switch when starting the daemon, the message will not appear on the system console, but it will be logged in the

audit/error log file. If this message appears at the console or is logged to the audit/error log file shortly after a comstart, you should ignore it and continue with processing as normal.

APPC/CPI-C Configuration File Changes

The sample configuration files provided with the APPC/CPI-C packages require several changes before the sample applications can be successfully used.

Workaround: Make the following changes (refer to chapters 5 and 7 of the *LINKix Administration Guide* for details):

- ◆ In *cpic2.cfg* (or *cpic.cfg*) you need to change the invocable program name from *csample2* to *tp2*.
- ◆ In *cpic2.cfg* you need to add the Symbolic Destination Name CPICTEST in order to run the CPI-C sample application.
- ◆ In *appc2.cfg* and *cpic2.cfg* change the Token Ring connection from active “initially” to active “on-demand.” A peer-to-peer connection via Token Ring cannot be active initially.

Screen Colors Are Not Restored

After running the configurator (*comconfig*), 3270 emulation (*com3270*) or manage utilities, the previous screen colors are not restored on exit.

Last Character in 3270 Status Line is Not Drawn

The last character in the 3270 status line isn’t always drawn. This problem will be corrected in a future release.

Wyse 60 Arrow Keys

When using a Wyse 60 terminal, the down and left arrow keys don’t work. This problem may be addressed in a future release.

Screen Size Reset to 24 by 80

We've found that *com3270* improperly resets the screen to 24 by 80 (regardless of the defined screen size) when the CLEAR key is pressed if both the primary screen size and alternate screen size are defined by the BIND. The problem only occurs when the host sends a WRITE command.

Workaround: If you are experiencing this problem, have the host send an ERASE WRITE or WRITE ALTERNATE, instead of a WRITE.

Restart Link After Changing Configuration

We've found that the LINKix configurator, *comconfig*, does not always notify when the Link must be restarted (when making changes to a running configuration file).

Workaround: If you make any changes to the Link or Connection information in a running configuration file, you should restart both the Link and the Node to ensure that the changes take effect.

Command Line File Transfer From Temp Shell

When using the 3270 command line file transfer utilities, *comsend* and *comreceive*, from a temp shell, no progress messages are displayed and the file transfer is performed at a slow rate.

Workaround: Execute the command line file transfer utilities from a separate virtual terminal instead of a temp shell.

Command Line File Transfer and Host Errors

The 3270 command line file transfer utilities, *comsend* and *comreceive*, do not currently process errors returned from the host during a file transfer (e.g., disk full).

Workaround: Use the 3270 menu interface for performing file transfers (host errors are correctly processed by menu-driven file transfers). If using the command line file transfer utilities, you must abort the transfer from the File Transfer dialog. (See also, **File Transfer: Exceeding Maximum Number of Extents** in these Release Notes.)

Running fifodiag On Ports Greater Than 2

The fifodiag hardware diagnostic utility currently panics the system when executed against a port greater than 2. This problem is currently being investigated and will be corrected as soon as possible.

Workaround: Do not run fifodiag against port 3 or 4.

Specifying an Invalid Port Number Panics System

Specifying an invalid port number when configuring a link can cause the system to panic. Also, executing the *puinfo* utility with an invalid port number can cause the system to panic.

Workaround: When configuring a link or running *puinfo* make sure that you specify a valid port number.

3270 User Notes

Model 3, 4 and 5 Support on UnixWare 1.1

When using LINKix 3270 with UnixWare 1.1, the `c3270 -m` switch for setting up a model 3, 4 or 5 screen is only supported from the desktop. On an ASCII terminal, you must configure the terminal for the correct number of rows and columns before starting 3270 emulation with `com3270`. In addition, the correct model size must be specified in the 3270 configuration and the “Model Can Be Overridden” check box should not be checked.

If you cannot get Model 5 because the monitor resolution isn’t fine enough (although the terminal is capable of the necessary quality) use `/usr/X/addr.setvgamode`. Refer to the UnixWare documentation for details on this utility.

3270 QUIT Key on UnixWare 1.1

The default key mapping for the 3270 QUIT key (**CTRL+T**) conflicts with a UnixWare feature. On UnixWare, **CTRL+T** is used for printing international characters so it is absorbed by the operating system before it can be used as the QUIT character for the 3270 product. To make use of the 3270 QUIT character, you must remap it to another keystroke sequence (for example, **ESC T**).

File Transfer: Exceeding Maximum Number of Extents

When sending a large file from the UNIX computer to the host, the following sequence can occur if the storage space (the number of extents) reserved at the host is not large enough to hold the file

1. The file transfer stops prematurely (the number of bytes sent to the host, as displayed on the screen, stops at a number less than the size of the file.
2. Returning to the 3270 emulation screen, you see the message:
Exceeded maximum number of extents for dataset
3. To cancel the interrupted transfer, from the File Transfer dialog select the session on which the file transfer is in progress and choose `< Abort transfer >`. See Stopping a File Transfer in chapter 4 of the 3270 User’s Guide for details.

“Cannot create shared memory for session id 65” Error

If you start *com3270* twice using the same style file which contains HLLAPI session IDs assigned, the following messages will be written in the LINKix *com.err* file:

```
Cannot create shared memory for session id number.  
Failed to initialize HLLAPI.
```

The *number* indicates the session id, in decimal, assigned to the first session in the style file (most often “A”, which is ASCII 65). Because shared memory has already been created for this session id, it is an error to attempt to create shared memory for the same session id a second time by the same user. When *com3270* detects this error, it does not continue the shared memory creation for the other sessions, therefore this audit message only appears for the first session.

LUs in Pools Must be Activated by Host

If a Communications Check Code of 923 (No Free LU in Pool) is returned when a user attempts to activate a session that’s configured to use a pool, either all the LUs in the pool have been activated and are in use or some of the LUs in the pool have not been activated by the host (and therefore are not available for use).

If all LUs have been activated and are in use, users can remap their sessions to use different LUs or pools (if their configuration does not specifically prevent them from remapping sessions). On the other hand, if some LUs in the pool have not been activated by the host, contact your host personnel to activate the remaining LUs.

3770 User Notes

Record Sizes Larger than 128 Bytes

LINKix 3770 customers who require record sizes larger than the default size of 80 bytes, or the supported exchange data size of 128 bytes, should take note of the new -m option available for use with the snajobsub command. This option replaces the previously undocumented -1 option. If you have used the -1 option with a previous version of the software, you should now use -m. See *snajobsub* in chapter 3 of the *3770 User's Guide* for complete details.

Console Program: Entering Host Commands

The following keystroke sequences do not currently function as they are described in the 3770 User's Guide. These inconsistencies in the console program will be corrected shortly.

The documentation states that the Delete key deletes forward; however, it currently deletes backward (the same as Backspace). Use Ctrl + Y to delete forward.

General Notes

Upgrading From an Earlier Release?

If you are upgrading from a LINKix release earlier than 3.0.0.0 and wish to use configuration files from the earlier release, you' 11 have to convert the files to the newer format first.

1. Change to the directory that contains the configuration files.
2. Because the conversion utility overwrites the existing files, you may want to copy the existing files to another area before continuing.
3. Convert the files with the following command:

```
comconvcfg <cfgfile> [<cfgfile2> <cfgfile3> . . . ]
```

4. After the conversion utility has converted the configuration files (overwriting the existing files), you are ready to continue with configuration.

Token Ring Driver Requirements

If you are using a Token Ring communications adapter board, make sure that the board is supported by the Token Ring device driver software provided with the UnixWare operating system. If you are using an IBM Token Ring card, you must have installed the necessary update to UnixWare 1.1.1 (referred to as release 6, but not officially released as of 7/1/94). This update should be obtained from your UnixWare distributor or directly from Univel/Novell.

Configuring Local Nodes

Please take note that only one node may be defined at each LINKix server. This node, however, may service multiple links.

Restarting Link and Local Node

When making changes to a running configuration file that affect a link or connection, you should restart the link and the local node. See **Starting/Stopping LINKix Services** in chapter 11 of the *LINKix Administration Guide* for details.

Coprocessor and APPC on Dial-up Connection

If you are running an APPC application with the coprocessor version of LINKix on a dial-up connection, you must ensure that one of the following is true:

- ◆ The DSR signal from your modem or modem eliminator is set high for the duration of the session (refer to the documentation that accompanied your modem or modem eliminator for details).
- ◆ The secondary APPC node must be readily available, such that the primary APPC node can make a viable connection (reaching the point of exchanging receiver ready frames between nodes).

If neither case is true, a “flurry” of messages will be displayed on the screen when the primary node is activated, indicating that the connection cannot be established. Allowing the APPC primary node to continue its attempt to make a connection may eventually lead to a “heap exhausted” message. At this point, you must perform a comstop and comstart sequence to reset the LINKix software.

User and Session Limits

This release of LINKix allows for a maximum number of concurrent users (32, 64, 128 or 254, depending on which version you’ve purchased) and a maximum number of actively bound sessions. If the user tries to perform an action that would cause either limit to be exceeded, an error is returned. If you suspect that you’re exceeding one of these limits, inspect the *com.aud* or *com.err* files for the appropriate message. If your installation’s needs consistently exceed the user or session limits, contact Cleo Communications for availability and pricing of upgrades.

User Count

The user count is the total number of

- ◆ all copies of *com3270* that are currently running
- ◆ all copies of *snarjewks* that are currently running
- ◆ all APPC TPs that are currently running
- ◆ all LUA applications that are currently running

Session Count

The session count is the total number of:

- ◆ all 3270 LUSs that are connected to the host
- ◆ all 3770 LUs that are connected to the host
- ◆ all APPC sessions allocated between the local LU and partner LU
- ◆ all LUA LUs that are connected to the host

MAXDATA Host Parameter

Note that the MAXDATA value at the host, must match the SDLC Frame Size that you specified during software installation. If these values do not match, LINKix will not operate correctly. To change the LINKix SDLC Frame Size, use the *LZNKix.cfg* utility (see chapter 3 of the *Administration Guide*, **Software Installation**).

Warning When Compiling nmapitest.c Under UnixWare

The following warning message, which appears when compiling the *nmapitest.c* sample program under UnixWare:

"nmapitest . c" , line 720: warning: argument is incompatible with prototype: arg #2
can be ignored. This warning appears even when the program compiles correctly.

Minimum Data Rate: 2400 baud

Take note that LINKix supports data communication rates of 2400 baud or higher. Slower speeds are currently not supported.

Removing Advanced NetView Supplemental Package

If you've purchased and installed the LINKix Advanced NetView supplemental product and wish to remove it at a later time, make sure you remove the NetView connection from the LINKix configuration (see chapter 10 of the *Administration Guide* for details) before removing the supplemental package. If you remove the Advanced NetView supplement without first removing the NetView connection, the control daemon will subsequently fail to start, returning a message that the NetView management daemon (netman) could not be initialized. To correct the problem you must reinstall the Advanced NetView supplement, remove the NetView connection from the LINKix configuration and then remove the Advanced NetView package.

Administration Guide

December 1995



December 13, 1995

Administration Guide

December 1995

Interface Systems reserves the right to, without notice, modify or revise all or part of this document and/or change product features or specifications and shall not be responsible for any loss, cost or damage, including consequential damage, caused by reliance on these materials.

This document may not be reproduced, stored in a retrieval system or transmitted, in whole or in part, in any form or by any means (electronic, mechanical, photocopied or otherwise) without the prior written permission of Interface Systems.

GOVERNMENT RESTRICTED RIGHTS

Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Use, reproduction or disclosure is subject to 52.227-19 (a) through (d) and restrictions set forth in the accompanying end user agreement.

GOVERNMENT LIMITED RIGHTS

Limited rights shall be effective indefinitely and are not subject to expiration as set forth in paragraph (3) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Copyright © 1995 Interface Systems, Inc. – All rights reserved.

Portions:

Copyright © 1989-1992 Data Connection Limited

Copyright © 1989, 1990 Digital Communications Associates, Inc.

Copyright © 1989, 1990 Microsoft Corporation

Table of Contents

About the Documentation

Documentation Conventions	xv
About This Manual	xvi
Related Publications	xviii

1. Introduction

LINKix 3270	1 - 4
LINKix 3770 Remote Job Entry (RJE)	1 - 7
Application Program Interfaces (APS)	1 - 9
Client/Server Architecture	1-11

2. The User Interface

Terminal Requirements	2 - 2
Required Functions	2 - 2
Recommended Functions	2 - 2
Menu Bar	2 - 4
Menus	2 - 6
Dialogs	2 - 8
Pushbuttons	2 - 8
Check Boxes	2-10
Radio Groups	2-11
Edit Boxes	2-12
List Boxes	2-13
Elevator Bar	2-14
List Boxes with Associated Edit Boxes	2-14
Dual List Boxes	2-15
Accelerator Keys	2-17

Table of Contents

Other Interface Features	2-18
Disabled (Grayed) Items	2-18
On-line Help Screens	2-18
Message Boxes	2-20
Screen Redrawing	2-21
Fast Exit from Menu Interface Programs	2-22
 3. Software Installation	
LINKix Packages	3 - 2
UnixWare	3 - 4
Installing LINKix	3 - 4
Removing LINKix	3 - 7
SCO UNIX System V/386	3 - 9
Installing LINKix	3 - 9
Removing LINKix	3-12
Additional Setup Requirements	3-13
Copy a Configuration and Security File	3-13
Copy the Style File	3-14
Changing Hardware Configuration With <i>LINKix.cfg</i>	3-15
Limiting User Access to Functionality	3-22
Restricting All Functions to a Specific Group of Users	3-22
Restricting System Administrator Functions to Single User	3-22
 4. Configuration Overview	
Overview	4 - 2
Starting the Configuration Program	4 - 5
Saving a Configuration and Exiting	4 - 8
Change File Permissions	4-10
Print Configuration File Contents	4-11

5. Configuring Services

Configuring Links	5 - 2
SDLC Link	5 - 4
Token Ring Link	5 - 6
Vendor X.25 Link	5 - 8
Configuring Connections	5 - 9
SDLC Connection	5-11
Token Ring Connection	5-19
X.25 Connection	5-25
SVC Connection	5-30
PVC Connection	5-33
Configuring Local Nodes	5-34
SDLC Parameters	5-39
Token Ring Parameters	5-40
X.25 Parameters	5-41
Links: Adding, Deleting, Renaming	5-42
Connections: Adding, Deleting, Renaming	5-45
Local Nodes: Adding, Deleting, Renaming	5-48
Where to Next?	5-52

6. Configuring for 3270

Overview of 3270 Configuration	6 - 2
Configuring 3270 LUs	6 - 3
Adding 3270 LUs	6 - 4
Adding a Single LU	6 - 4
Adding a Range of LUs	6 - 5
Editing a 3270 LU Configuration	6 - 9
Deleting an LU	6-10
Renaming an LU	6-10
Configuring a 3270 LU	6-11

Table of Contents

Configuring 3270 LU Pools	6-13
Adding a 3270 LU Pool	6-14
Editing an LU Pool Configuration	6-15
Deleting an LU Pool	6-15
Renaming an LU Pool	6-16
Configuring an LU Pool	6-16
Configuring 3270 Users	6-18
Adding a User	6-20
Editing a 3270 User Configuration	6-21
Deleting a User	6-21
Renaming a User	6-21
Configuring a 3270 User	6-22
Configuring RTM Settings	6-27
Configuring NetView User Alerts	6-30
Style File	6-33

7. Configuring for APPC and CPI-C

Overview of APPC Configuration	7 - 2
Mode Configuration	7 - 2
LU Confirmation	7 - 2
TP Configuration	7 - 3
Implicit Modes	7 - 3
Implicit Partner LUs	7 - 4
Host System Restrictions	7 - 4
Overview of CPI-C Configuration	7 - 6
Configuration Requirements	7 - 7
Two TPs on the Local Computer	7 - 7
TPs on Different Computers	7 - 7
Configuring APPC Modes	7 - 9
Adding a Mode	7-10
Editing an APPC Mode Configuration	7-11
Deleting a Mode	7-11
Renaming a Mode	7-13

Table of Contents

Configuring a Mode	7-13
Session Limits and Contention Winner Sessions	7-16
Supporting Parallel Sessions	7-17
Defining RU Lengths	7-17
Configuring Remote APPC LUs	7-19
Adding a Remote LU	7-20
Editing a Remote LU Configuration	7-21
Deleting a Remote LU	7-22
Renaming a Remote LU	7-23
Configuring a Remote LU	7-23
Configuring Implicit Remote LUs	7-26
Configuring Local APPC LUs	7-27
Adding a Local APPC LU	7-28
Editing a Local LU Configuration	7-29
Deleting a Local LU	7-30
Renaming a Local LU	7-30
Configuring a Local LU	7-31
Configuring for Implicit Remote LUs	7-35
Associating a Mode with an LU Pair	7-35
Configuring Implicit Modes	7-37
Configuring Invocable TPs	7-38
File Ownership of Invocable TPs	7-38
Adding an Invocable TP	7-39
Modifying an Existing Invocable TP	7-40
Renaming an Existing Invocable TP	7-41
Deleting an Invocable TP	7-41
Configuring an Invocable TP	7-41
Configuring TP Security	7-46
Adding a New User ID	7-48
Deleting a User ID	7-49
Specifying a Password	7-49
Configuring CPI-C Symbolic Destination Names	7-51

Table of Contents

8. Configuring for LUA

Overview of LUA Configuration	8 - 2
Configuring LUA LUs	8 - 3
Adding LUA LUs	8 - 4
Adding a Single LU	8 - 4
Adding a Range of LUA LUs	8 - 5
Editing an LUA LU Configuration	8 - 8
Deleting an LUA LU	8 - 9
Renaming an LUA LU	8 - 9
Configuring an LUA LU	8 - 9
Configuring LUA LU pools	8 - 12
Adding a New LUA LU Pool	8 - 13
Editing an LUA LU Pool Configuration	8 - 14
Deleting an LUA LU Pool	8 - 14
Renaming an LUA LU Pool	8 - 14
Configuring an LUA LU Pool	8 - 15

9. Configuring for 3770 RJE

Overview of RJE Configuration	9 - 2
Configuring RJE LUs	9 - 3
Adding RJE LUs	9 - 4
Adding a Single LU	9 - 4
Adding a Range of RJE LUs	9 - 5
Editing an RJE LU Configuration	9 - 8
Deleting an RJE LU	9 - 9
Renaming an RJE LU	9 - 10
Configuring an RJE LU	9 - 10
Configuring RJE FCBs	9 - 12
Adding a New FCB	9 - 13
Editing an RJE FCB Configuration	9 - 14
Deleting an Existing FCB	9 - 14
Renaming an Existing FCB	9 - 15
Configuring an RJE FCB	9 - 15

Table of Contents

Configuration RJE Workstations	9 - 1 7
Adding a New RJE Workstation	9 - 1 8
Editing an RJE Workstation Configuration	9 - 1 9
Deleting an Existing Workstation	9 - 1 9
Renaming an Existing Workstation	9 - 1 9
Configuring an RJE Workstation	9 - 2 0
Configuring Printers for an RJE Workstation	9 - 2 3
Adding Printers	9 - 2 3
Editing a Printer's Configuration	9 - 2 4
Deleting Printers	9 - 2 4
Configuring Workstation Printers	9 - 2 4
Configuring Punches for an RJE Workstation	9 - 2 8
Adding Punches	9 - 2 9
Editing a Punch's Configuration	9 - 2 9
Deleting Punches	9 - 2 9
Configuring Workstation Punches	9 - 3 0
 10. Configuring Diagnostics	
Network Management Connection	1 0 - 2
UCF Security	1 0 - 5
Configuring Audit and Error Logs	1 0 - 6
Resetting the Audit and Error Log Files	1 0 - 8
 11. Managing LINKix Components	
Starting/Stopping the Control Daemon	1 1 - 2
Starting the Control Daemon	1 1 - 2
Stopping the Control Daemon	1 1 - 4
Starting/Stopping LINKix Services	1 1 - 5
Starting Services	1 1 - 5
Stopping Services	1 1 - 6
Daemon Startup Problems	1 1 - 7
Managing Services from the Command Line	1 1 - 8
Tracing Options	1 1 - 9
Administration Guide	ix

Table of Contents

Examples	11-11
V.25bis Dialer	11-12
Using <i>v25dial</i>	11-12
Example Commands	11-13
The <i>dial</i> Script	11-14
Using <i>v25cmd</i>	11-14
Some Common V.25bis Commands	11-15
Return Codes	11-16
 12. Troubleshooting	
Troubleshooting LINKix 3270	12 - 2
3270 Emulation Problems	12 - 2
3270 Session Problems	12 - 3
Troubleshooting LINKix 3770	12 - 7
It Was Working Just Fine	12-11
 13. Advanced Diagnostics	
Diagnostics Support	13 - 2
Trace Files	13 - 2
Audit and Error Log Files	13 - 2
Tracing	13 - 4
Tracing User Space	13 - 5
SNATRC	13 - 5
SNAMSG	13 - 6
SNATRUNC	13 - 7
SNACTL	13 - 7
Controlling Tracing from within an Application	13 - 8
Tracing Kernel Space	13 - 9
Specifying Trace File Names	13 - 9
Enabling Tracing at Initial Start	13 - 9
Trace File Formats	13-10
Process ID	13-10
Message Data	13-10

Table of Contents

APPC Trace File Format	13-11
CPI-C Trace File Format	13-13
LUA and CSV Trace File Format	13-16
HLLAPI Trace File Format	13-19
Kernel Trace File Formats	13-20
Kernel Trace Abbreviations	13-20
SNA Message Trace	13-21
Internal Message Trace	13-22
coptool	13-26
Problem Determination	13-30
Incorrect or Conflicting I/O Address	13-30
Incorrect or Conflicting Memory Address	13-30
Determine if the Board is Running	13-31
Incorrect or Conflicting IRQ	13-31
Correct PU Address Configured	13-32

A. Configuration Site Guides

B. Sample VTAM Definition

C. Kernel Tuning

D. Exit Codes

Table of Contents

About the Documentation

All CLEO products come with a comprehensive library of documentation that consists of several manuals and an on-line help system.

Administration Guide

The *Administration Guide* explains the procedures for installing and configuring the product, suggested troubleshooting of installation problems, setting up system files, as well as monitoring and tracing program execution. A chapter on diagnostics explains audit and error log files and possible solutions to problems. There's also information on NetView alerts. This manual is provided with all products.

3270 User's Guide

The *3270 User's Guide* describes LINKix 3270 product features, architecture, connectivity, configuration options, user interface, how to start and stop 3270 emulation, transferring files, using customization features such as remapping the keyboard and display colors, interpreting status-line information and viewing response times. An appendix in this manual contains a glossary of terms used throughout the manuals.

HLLAPI Programmer's Guide

The *HLLAPI Programmer's Guide* contains complete reference information on the LINKix 3270 High-Level Language Application program Interface (HLLAPI). It includes conceptual and detailed reference information, as well as sample application program source code.

APPC Programmer's Guide

The *APPC Programmer's Guide* contains complete reference information on CLEO'S implementation of the APPC programmer's interface. It includes conceptual and detailed reference information. Sample application program source code is also provided.

LUA Programmer's Guide

The *LUA Programmer's Guide* contains complete reference information on CLEO'S implementation of the LUA programmer's interface. It includes conceptual and detailed reference information. Sample application program source code is also provided.

3770 User's Guide

The *3770 User's Guide* describes product features and functions of LINKix 3770 RJE, explains how a job is submitted and processed, explains the RJE commands used to start and stop an RJE workstation daemon and explains how to use the RJE console program.

Common Service Verbs Programmer's Guide

The *Common Service Verbs Programmer's Guide* contains complete reference information to CLEO's implementation of the Common Service Verbs (CSVs) programmer's interface. It includes conceptual and detailed reference information.

On-Line Help

On-line help is available in the following programs whenever the menu interface is active:

- ◆ Configuration program
- ◆ 3270 emulation program
- ◆ 3770 console program

Pressing **F1** at any point in the menu interface initiates the help system. A help screen appears with information about the options that are available at this point. While a help screen is displayed, you can press **F1** again to receive information about the help system. This explains the type of information you will receive and how to move around in the help screen system.

Documentation Conventions

The following documentation conventions are used in this manual:

<i>italics</i>	Italicized words are used for emphasis, to identify the titles of manuals and to indicate file and directory names.
bold	Bold type is used for cross-references to other chapters within the manual or to other sections within a chapter.
KEYNAME	A key name in this format, for example ENTER, refers to a specific key on your terminal's keyboard. Some keyboard commands require that you press one key and hold it down while you press one or more other keys. These are also called combination keystrokes. To indicate this, key names are shown together with a + sign between them (example: CTRL + D). While holding down the first key, press the other key(s).
KEYNAME	A key name in uppercase letters, for example ENTER , refers to a 3270 key. Appendix A lists the default UNIX terminal keys used to represent these 3270 keys. If you remap your keyboard, you can always find the UNIX terminal keystroke used to represent a particular 3270 key by using the View and Define Key Definitions dialog.
Oxnn	A number preceded by 0x (for example 0x20) indicates a hexadecimal value.

About This Manual

This *Administration Guide* is organized as follows:

Part I: Introduction and Installation

Chapter 1, **Introduction**, provides an overview to CLEO'S LINKix 3270, LINKix 3770 and our Application Program Interfaces (APIs).

Chapter 2, The **User Interface**, explains how to use the CUA compliant user interface and its components.

Chapter 3, **Software Installation**, explains how to install, setup and remove the LINKix software.

Part II: Configuration

Chapter 4, **Configuration Overview**, explains how to start the configuration program and how to open and save a configuration file.

Chapter 5, **Configuring Services**, explains how to configure the Link and Connection services.

Chapter 6, **Configuring for 3270 Emulation**, explains how to configure the features of LINKix 3270 to communicate with host computers in an SNA network.

Chapter 7, **Configuring for APPC**, explains how to configure the Advanced program-to-program Communication (APPC) to communicate with other computers in an SNA network.

Chapter 8, **Configuring for LUA**, explains how to configure your system for applications using the LUA interface.

Chapter 9, **Configuring for 3770 RJE**, explains how to configure your system for RJE operation, by configuring one or more RJE Workstations.

Chapter 10, **Configuring Diagnostics**, explains how to configure advanced features such as Response Time Monitor settings, diagnostics and NetView user alerts.

Part III: System Management and Troubleshooting

Chapter 11, **Managing LINKix Components**, describes how to start and manage various LINKix components, including the drivers, the control daemon, the command line manage utility and other components. This chapter also describes and explains various messages that can be output by the LINKix components.

Chapter 12, **Troubleshooting**, describes some general troubleshooting strategies for LINKix 3270 and LINKix 3770.

Chapter 13, **Advanced Diagnostics**, explains the tracing facilities, including the different types of tracing available, how to control tracing on each software component and how to interpret the trace files produced.

Appendices

Appendix A, **Configuration Site Guides**, provides a series of configuration forms that can aid you in gathering the necessary data you'll need when configuring the product.

Appendix B, **Sample VTAM Definition**, shows a VTAM definition and explains how this definition maps to your LINKix 3270 configuration.

Appendix C, **Kernel Tuning**, describes the UNIX kernel tuning parameters which affect the function of LINKix 3270 and 3770 LINKix and explains how to modify them.

Related Publications

The documentation that accompanies this product does not attempt to fully describe IBM's Systems Network Architecture (SNA), any of the 3270 information display system products, APPC, LUA, RJE or the UNIX operating system. You may, therefore, find one or more of the following documents useful.

SNA

For more information on SNA, see the following publications:

GC30-3073	<i>IBM Systems Network Architecture: Technical Overview</i>
GC30-3072	<i>IBM Systems Network Architecture: Concepts and Products</i>
GA27-3136	<i>IBM Systems Network Architecture: Formats</i>
SC30-3112	<i>IBM SNA Format and Protocol Reference Manual: Architectural Logic</i>
GC20-1 869	<i>IBM Systems Network Architecture: Introduction to Sessions between Logical Units</i>
GG24-1557	<i>IBM Advanced Communications Function Products Installation Guide</i>

Common User Access

For information on Common User Access (CUA), refer to:

SC26-4582	<i>IBM Systems Application Architecture Common User Access: Advanced Interface Design Guide</i>

3270 Display Systems

For more information on 3270 information display system products, see the following publications:

GA23-0061	<i>IBM 3270 Information Display System: 3274 Control Unit Description and Programmer's Guide</i>

GA33-3056	<i>IBM 3270 Information Display System: Color and Programmed Symbols</i>
GA23-0023	<i>IBM 3270 Information Display System: 3274 Control Unit Display Station: Operator's Guide</i>
GA23-0218	<i>IBM 3270 Information Display System: 3174 Control Unit Functional Description</i>

APPC/LU 6.2

For information about APPC (LU 6.2), refer to these documents:

GG24-1584	<i>IBM Systems Network Architecture: Introduction to APPC</i>
GC30-3084	<i>IBM Systems Network Architecture: Transaction Programmer's Reference Manual for LU Type 6.2</i>
SC30-3269	<i>IBM Systems Network Architecture: Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2</i>
SC30-6808	<i>IBM Systems Network Architecture LU6.2 Reference: Peer Protocols</i>
SC30-3400	<i>IBM Systems Network Architecture: VTAM Programming for LU Type 6.2</i>

You may also want to refer to the list of general SNA documents.

LUA

The following SNA documentation provides important information that can be of use when programming with the LUA API:

GA27-3136	<i>IBM Systems Network Architecture: Formats</i>
SC30-3112	<i>IBM Systems Network Architecture Format and Protocol Reference Manual: Architectural Logic</i>
GC20-1869	<i>IBM Systems Network Architecture: Introduction to Sessions between Logical Units</i>

About the Documentation

RJE

For information about RJE, refer to these documents:

SC23-0046 *System Programming Library: JES2 Initialization and Tuning*

SC23-0064 *MVS/Extended Architecture Operations: JES2 Commands*

GC28-1352 *MVS/Extended Architecture JCL Reference*

UNIX

For information on the UNIX operating system and its utilities, refer to the user documentation provided with your UNIX system.

1. Introduction

The product you've purchased is part of CLEO's complete family of UNIX-based software products that permit a UNIX computer to communicate with other computers using SNA communications. LINKix emulations (3270 and 3770) and all related APIs (HLLAPI, APPC/LU6.2, LUA/LUO and CSV) adhere to the latest standards and protocols in the industry. In addition, our products provide complete file transfer capabilities, as well as many other advanced features.

The CLEO product that you've purchased incorporates one or more of the following features and functions:

3270 Emulation

LINKix 3270 provides users with most of the standard features of an IBM 3270 Information Display System, such as extended attribute byte support for color and extended highlighting. In addition, the emulation program provides file transfer capabilities and allows you to develop and run HLLAPI transaction programs.

3770 RJE

LINKix 3770 enables your UNIX systems to emulate IBM 3776 and 3777 RJE (Remote Job Entry) workstations for fast, easy batch transfer operations through SNA communications. You can use LINKix 3770 alone or in combination with our 3270 software for SNA. It supports the latest UNIX System V releases from AT&T, SCO, Interactive Systems and others.

HLLAPI

The LINKix 3270 HLLAPI programmer's API provides developers with the tool they need to develop application programs that interact with a host 3270 application as if it were a 3270 user. A HLLAPI application can enter data, process information sent by the host and so on. HLLAPI programs can also be used to automate start-up operations for communicating with a host application or to access host applications which require larger screen models than your terminal supports. HLLAPI is a standard feature provided with all versions of LINKix 3270.

APPC/LU62

APPC is a program-to-program API with which you can develop distributed applications that communicate across an SNA network using IBM's LU 6.2 protocol.

Introduction

With APPC you can communicate on a program-to-program basis with any system which provides APPC, such as an IBM host, an OS/2 PC or another UNIX computer. The APPC API is optionally available with all remote SNA products.

APPC applications can also communicate with CPI-C applications, since the format of the data transmitted between applications is the same for both interfaces. Neither application needs to know whether its partner is using APPC or CPI-C.

CPI-C

CPI-C is a program-to-program API that enables the development of distributed applications across an SNA network. It provides the same functions as APPC, but with a different programming interface.

CLEO's CPI-C implementation is compatible with the implementation provided by the DCA/Microsoft Comm Server and Comm Workstation products, and with that provided by IBM's Networking Services/2, version 1.0.

CPI-C applications can also communicate with APPC applications, since the format of the data transmitted between applications is the same for both interfaces. Neither application needs to know whether its partner is using APPC or CPI-C.

LUA/LUO

LUA is a programmer's API that enables the development of applications that communicate with IBM host computers across an SNA network. LUA uses the LU types 0, 1,2 and 3. You can develop new applications using LUA to communicate with host applications, or you can adapt existing PC-based LUA applications (without any need to modify the host applications). The LUA API is optionally available with all remote SNA products.

Common Service Verbs (CSV)

CSV is an additional API that provides general services for application programs. CSV verbs can be used within HLLAPI, APPC and LUA applications, or they can be used independently. CSV capabilities include character-set conversion, trace control, message logging and transmission of alerts to IBM's NetView. CSV is a standard feature provided with all versions of the LINK.ix product line.

NetView/DM Client Support

Support for NetView/DM (an IBM host system application) is provided through CLEO's NetView/DM Client software. This product provides the UNIX computer with the ability to support NetView/DM or a similar host application, Distributed

System Executive (DSX). NetView/DM support is an optional feature available with all remote SNA products.

Client/Server Architecture

LINKix offers a client/server package that allows users to perform distributed cooperative processing. With our client/server software, LINKix components are installed on separate client and server nodes which cooperate to perform the desired functions. The client/server feature can be purchased and installed as part of the initial LINKix installation or as a subsequent upgrade.

LINKix 3270

CLEO's LINKix 3270 provides complete 3270 emulation for UNIX via SNA. With LINKix 3270, your UNIX computer emulates an IBM 3270 series terminal. “

LINKix 3270

Standard Features

- 3278/79 terminal emulation
- 3287 printer emulation
- Complete Model 2 through 5 full-screen support*
- Complete extended attribute byte (EAB) support*
- Color remapping*
- Support for international language versions of EBCDIC at the host*
- Keyboard remapping
- File transfer compatible with IBM's IND\$FILE host file transfer program
- HLLAPI Programmer's API
- Common Service Verbs (CSV) API
- Response Time Monitor (RTM)**
- 3270 user alerts (through IBM's NetView)**

Optional Features

- APPC and CPI-C Programmer's APIs**
- LUA/LUO Programmer's API**
- NetView/DM client support**
- QLLC interface kit**

* Support for screen models other than model 2, EAB, color remapping and international operation is dependent on your UNIX computer and its terminals.

** Available in the SNA Remote version only.

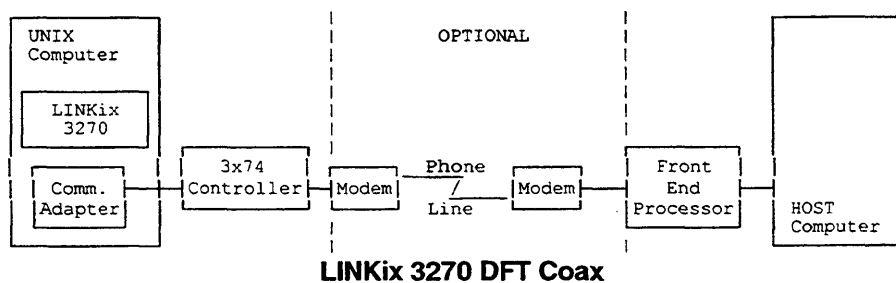
LINKix 3270 was developed using the UNIX STREAMS architecture. It supports the latest UNIX System V releases from AT&T, SCO, Interactive Systems, AIX and others, and it runs on a 386/486-based PC/XT/AT, PS/2 or compatible system.

LINKix 3270 can be purchased in one of several different versions: DFT Coax, Remote or NetWare Systems compatible.

LINKix 3270 DFT Coax

LINKix 3270 DFT Coax connects to an IBM 3x74 controller (through a Type A adapter) via our coax board. LINKix 3270 DFT Coax allows up to five simultaneous sessions which can be assigned as any combination of terminal and printer emulation.

With LINKix 3270 DFT Coax, the communications adapter board is connected to a 3x74 controller which in turn maybe attached to a modem which communicates over a phone line to another modem that's attached to a front end processor that's connected to the host computer, or the 3x74 controller may be directly connected to the host system's front end processor.

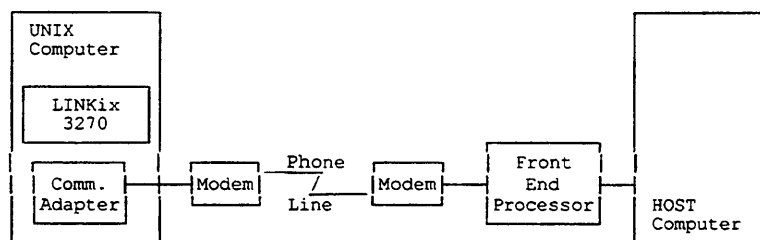


LINKix 3270 Remote

LINKix 3270 Remote provides multi-user UNIX-to-mainframe connectivity without controllers or coax boards. Because it also emulates the functions of the 3x74 controller, it links directly to synchronous modems or modem eliminators through a synchronous interface board or our coprocessor board. With the coprocessor board, the 3270 communication tasks are offloaded from your UNIX system to the coprocessor, thus freeing your UNIX computer for applications processing. The coprocessor board is your best solution for HLLAPI applications that require a large number of sessions, maximum throughput and efficient use of memory.

In an SNA or X.25 link, the communications adapter board is connected to a modem which communicates over a phone line to another modem which is attached to a front end processor that's connected to the host computer.

Introduction



LINKix 3270 Remote

LINKix 3270 Remote can be purchased with support for up to 254 sessions in SNA. The remote SNA version provides support for IBM's NetView, as well as NetView/DM client services. It's also available with APPC/LU6.2 and LUA/LUO capabilities. We also offer a QLLC interface kit for connecting to SNA hosts through X.25 networks.

LINKix 3770 Remote Job Entry (RJE)

LINKix 3770 provides complete 3770 RJE for UNIX via SNA. LINKix 3770 enables your UNIX systems to emulate IBM 3776 and 3777 RJE (Remote Job Entry) workstations for fast, easy batch transfer through SNA communications.

You can use LINKix 3770 alone or in combination with our LINKix 3270 software for SNA. It supports the latest UNIX System V releases from AT&T, SCO, Interactive Systems and others.

LINKix 3770

Standard Features

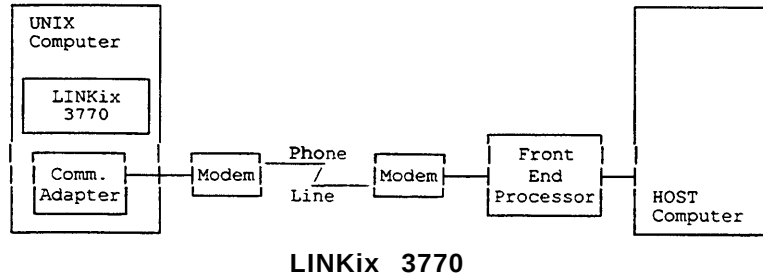
- 3776/77 RJE workstation emulation
- RJE scripting language
- Printer Forms Control
- SCS (SNA Character Stream) support
- ASCII-EBCDIC translation
- Menu-driven configuration
- Multiple sessions
- Function Management Interface (FMI)
- Common Service Verbs (CSV) API

Optional Features

- APPC and CPI-C Programmer's APIs
- LUA/LUO Programmer's API
- NetView/DM client support
- QLLC interface kit

Introduction

LINK.ix 3770 operates with our high-performance coprocessor boards as well as our economical synchronous interface boards, providing remote SNA connectivity via synchronous modems or modem eliminators.



Common RJE Applications

Electronic Data Interchange (EDI)
Medical Claims Filing
Electronic Tax Filing
Credit Card Verification
Stock Market Exchange Data

Point-of-Sale
Electronic Funds Transfer (EFT)
Check Clearing and Deposits
Customs Automated Broker Interface

Application Program Interfaces (APIs)

In addition to our 3270 and 3770 emulation products, CLEO offers four Application Program Interfaces (APIs): HLLAPI, APPC, CPI-C and LUA. HLLAPI is the standard API provided with all versions of LINKix 3270, while APPC, CPI-C and LUA are optionally available with any of our remote SNA products.

HLLAPI

HLLAPI programs provide UNIX users with easy access to mainframe applications. With CLEO's LINKix 3270 HLLAPI, you can write programs that enable your UNIX systems to interact efficiently with your mainframe 3270 applications.

HLLAPI programs can automate repetitive tasks, such as log-on and data-transfer procedures. They can also help mask complicated mainframe operations from your UNIX users.

LINKix 3270 HLLAPI is compatible with IBM's OS/2 Extended Edition High-Level Language Application program Interface (EHLLAPI).

APPC/LU6.2 and CPI-C

With CLEO's APPC/LU6.2 package, your UNIX systems can perform program-to-program communications with your IBM mainframe, or you can bypass the mainframe altogether to communicate with other computers in your SNA network.

APPC/LU6.2 allows program-to-program communications between UNIX systems and other computers in your SNA network through IBM's LU Type 6.2.

With APPC, separate programs running on the same or different systems can communicate with each other to accomplish a single task, such as querying a remote database, copying a remote file or sending and receiving electronic mail.

CLEO's APPC provides a full implementation of IBM's Advanced Program-to-Program Communications for mapped and basic conversations. It also supports many APPC option sets. You can use APPC in combination with any of our remote SNA products.

Introduction

CPI-C is an API which provides the same program-to-program communications features as APPC, but with a simpler interface. The actual data transmitted between programs is the same, though.

A CPI-C application can communicate with another CPI-C application or with an APPC transaction program, using the same local LUs, remote LUs and modes as APPC. This makes the process for establishing communications between two CPI-C applications the same as for APPC.

The main differences between the APPC and CPI-C are as follows:

- ◆ Each CPI-C verb is a separate function call.
- ◆ Instead of setting up parameters in a verb control block (VCB) before making the call to APPC, a CPI-C application passes the information required for each function call as parameters to the call.
- ◆ Instead of setting parameters such as the conversation type, receive type and send type for each verb, you can set these parameters once using a separate CPI-C function; they then retain their settings until you change them. For example, you can set the conversation type to Mapped and all subsequent CPI-C verbs are treated as mapped conversation verbs.
- ◆ A CPI-C application does not need to specify its partner's TP name, LU name, mode name and security information when allocating a conversation. Instead, it can specify a symbolic destination name, which corresponds to an entry in the LINKix configuration file; all the information about the partner application is taken from this configuration file entry.

LUA/LU0

CLEO's LUA is a flexible, low-level API that enables you to write applications for your UNIX systems using LU Type 0. It's a great way to integrate your highly specialized and widely distributed UNIX applications into your SNA network.

With LUA, your applications can send and receive a broad range of SNA commands, giving you full control over the interface. Your application programs appear to the host as SNA logical units — either devices or programs, as you require.

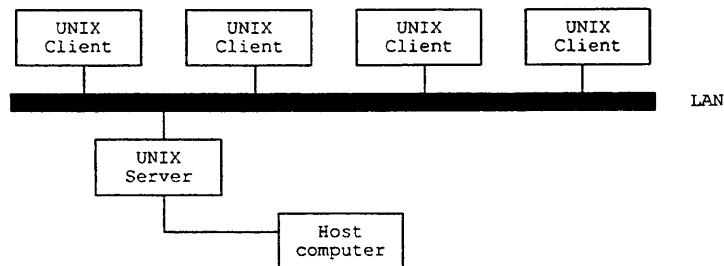
Client/Server Architecture

LINKix offers a client/server package that allows users to perform distributed cooperative processing. With our client/server software, LINKix components are installed on separate client and server nodes which cooperate to perform the desired functions. The communications between the LINKix client(s) and the LINKix server(s) is typically carried out over a suitable local area network (LAN).

Client/Server Structure

The computers on a LINKix LAN are of two types: servers and clients. A server contains one or more of the SNA service components required for communications with remote systems. A client, on the other hand, does not contain these components but accesses them on the server by means of the LAN. Servers and clients communicate across the LAN using TCP/IP sockets.

The diagram below shows a simple LAN setup for LINKix client/server.



The LINKix client/server product maybe used on a simple LAN with only one server and several clients, or on a larger LAN with several servers; additional servers and clients may be added as necessary.

Benefits of Client/Server

The client/server architecture provides the following benefits:

- ◆ A single connection to a remote SNA computer may be shared by many computers on the LAN
- ◆ Hardware and software resources on a server can be accessed by other servers and clients, reducing the need for duplication
- ◆ Storage requirements on clients can be minimized
- ◆ Multiple servers can provide load balancing and fault tolerance (clients can be routed to different servers according to available resources)
- ◆ Existing equipment and LINKix installations can be retained while taking advantage of client/server features

System Requirements

In order to install and use the LINKix client/server features, you must have the following software installed:

- ◆ A supported UNIX operating system
- ◆ TCP/IP
This software must be installed *and configured* before you install the LINKix client/server feature.

2. The User Interface

The user interface that appears in the configuration program complies with the SAA guidelines of CUA (Common User Access). This chapter describes how to use the interface and its components. Specifically, the following topics are covered:

- ◆ Terminal requirements
- ◆ nu bars
- ◆ Menus
- ◆ Dialogs
- ◆ Disabled (grayed) items
- ◆ Help screens
- ◆ Screen redrawing

Note that the example screens shown in this chapter do not show all video features such as highlighted characters or reverse video.

Terminal Requirements

In order to use the menu interface, your terminal should have certain functions defined in the UNIX computer's terminfo database. These functions fall into two categories:

- ◆ **Required.** Only terminals with these functions can display the menu interfaces. (Terminals without these functions can be used, however, for transaction programs operating under one of the LINKix AIPs.)
- ◆ **Recommended.** These functions make the screen interfaces easier to use, but can be used without them.

Required Functions

Screen Size

The screen size defined in terminfo must be at least 80 columns by 24 rows.

Cursor Addressability

Menus and dialogs cannot be drawn without this capability.

Alternate Character Mode

These characters are used to draw the menus and dialogs. If alternate character mode is not supported, the boxes do not appear correctly on the screen.

F1 Function Key

This key is used to invoke the on-line help facility.

Up and Down Arrow Keys

These keys are used to move within list boxes and to select radio buttons.

Recommended Functions

Left and Right Arrow Keys

These keys are used for moving within edit boxes and for selecting radio buttons.

PageUp and Page Down

These keys are used to move up and down within list boxes.

Home and End

These keys are used to move to the first or last entry in a list box. The **UP** and **DOWN** arrow keys or **PageUp** and **Page Down** can be used instead, but are slower.

BackTab

This key is useful for moving backward through the sequence of items in a dialog. If it is not available, the keystroke **Ctrl + B** can be used instead.

Insert, Delete and Backspace

The **Insert**, **Delete** and **Backspace** keys are useful for modifying information in an edit box.

Color and Attribute Support

While this product can be run without color or attribute support, these features can be put to good use if your terminal supports them - attribute support refers to features such as intense text, blinking text and underlining. (This applies particularly to the 3270 emulation program when communicating with host programs which support color and highlighting attributes.)

Menu Bar

The menu bar is the starting point for the user interface. When you start the configuration program the menu bar is displayed at the top of the main screen.

The example below shows the configuration program's main screen and menu bar.

```
File Services 3270 APPC LUA RJE Diagnostics          F1=Help
LINKix Configurator 3.0
-----
                                     Configuration file : /usr/lib/linkix/com.cfg loaded
                                     Copyright (C) 1993 CLEO Communications
Press Return to select a menu
```

The menu bar contains a number of menu names, each of which has one letter (the accelerator key) highlighted. One of the menu names may appear in reverse video, indicating that it's currently selected.

At the top right of the screen, F1=Help is displayed as a reminder that help screens are always available by pressing **F1**.

Choosing a Menu from the Menu Bar

To choose a particular menu from the menu bar, perform the following steps.

- ◆ Press the highlighted letter (accelerator key) from the name of the menu you want to open.

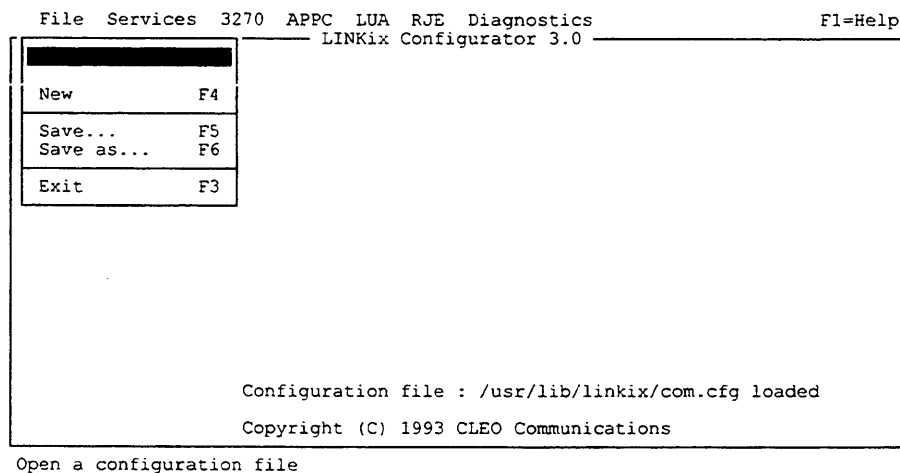
- ◆ Alternatively, press **Enter** to activate the reverse-video selector, which will highlight the first menu name. Use the **Right** and **Left** cursor keys to select the name of the menu you want, then use either the **Up** or **Down** cursor keys or **Enter** to pull down the menu.

The menu appears below the name on the menu bar.

Menus

A menu bar contains a number of menus (also called pulldowns or pulldown menus).

When you select an option from a menu bar, the menu is displayed. The following is an example of a menu.



Each menu contains a number of menu options (also called menu items) which indicate the functions you can access from the menu. One menu option appears in reverse video, the currently selected option. The bottom line of the screen displays information on the currently selected option.

Note that one or more options in the menu may be disabled (grayed) and cannot currently be selected (for more information on how these items operate, see **Disabled (Grayed) Items** later in this chapter).

Choosing a Menu Option

To choose a particular option from a menu, do one of the following:

Use the **Up** and **Down** cursor keys to move to the menu option you want (watch the reverse-video selector), then press **Enter**.

Press the accelerator key (the highlighted letter) of the menu option you want.

Press the direct accelerator key (this key or key combination is to the right of the menu option). Direct accelerator keys are available for the most commonly used menu options, for example the Exit option in the File menu uses the **F3** key as a direct accelerator key. Direct accelerator keys can be used from the main screen of the program, without first having to select the appropriate menu.

If a menu option is followed by an ellipsis (...), a dialog is displayed when you choose the menu option. For example, choosing Open from the File menu causes a dialog to appear that prompts you for the name of a configuration file to open.

```

File  Services  3270  APPC  LUA  RJE  Diagnostics  F1=Help
                        LINKix Configurator 3.0

```

Open

() Open running configuration file
(o) Open a specific configuration file

Configuration file name {com.....}.cfg
Configuration file path {/usr/lib/linkix/.....}

< OK > <Cancel>

Configuration file : /usr/lib/linkix/com.cfg loaded
Copyright (C) 1993 CLEO Communications

If a menu option is not followed by an ellipsis, an action is performed when you choose the menu option. For example, when you choose New from the File menu, a new configuration file is created.

Dialogs

Dialogs are used within the user interface for entering responses and making choices.

Dialogs contain one or more fields (areas where information is entered or choices are made). The fields in dialogs may be pushbuttons, check boxes, radio groups, edit boxes, list boxes and dual list boxes.

The **TAB** key (or **CTRL + 1**) moves you from one field in a dialog to the next; **BACKTAB** (or **CTRL + B**) moves you from the current field to the previous field. In some cases you can move to another field by using accelerator keys; the use of this feature depends on the type of field.

The cursor movement keys do not move you through the fields in the dialog, but rather perform specific actions that depend on the type of field the cursor is in. The **UP** and **DOWN** arrow keys allow you to move within list boxes and to select radio buttons. The **Left** and **Right** arrow keys are used to move within edit boxes and for selecting radio buttons.

Pushbuttons

Pushbuttons are controls that when chosen cause an action to occur. In the example below, **< Zoom >**, **< Remap >**, **< OK >** and **< Cancel >** are pushbuttons.

```
File  Services  3270  APPC  LUA  RJE  Diagnostics  F1=Help
                        Configure 3270 User <DEFAULT>

Description . . . . . [Generic User.....]
Max enabled sessions . . . (o) No limit  ( ) Limit of [..]

Session  LU/pool  Permissions for user:
number   name     Type
-----
1        CRTLU    D ^
2        PRTLUI   P #
3        [UNUSED] #
4        [UNUSED] #
5        [UNUSED] #
6        [UNUSED] #
7        [UNUSED] #
8        [UNUSED] #
9        [UNUSED] #
10       [UNUSED] v

Initial style
[NONE] ^ [com3270.]
[User Defined] #
com3270 #
v

< Zoom > < Remap > < OK > < Cancel >
```

Pushbutton names are enclosed in angle brackets. They're always displayed at the bottom of dialogs, with one exception: < Move>, which appears with Dual list boxes.

Choosing a Pushbutton

To choose a pushbutton, do one of the following:

- ◆ Use the **TAB** key to move the cursor to the pushbutton (this highlights the angle brackets of the pushbutton) and press **ENTER**.
- ◆ Use the accelerator key of the pushbutton.

On each dialog, there's always a default pushbutton (indicated with highlighted angle brackets). Pressing the **ENTER** key anywhere in the dialog chooses this pushbutton.

Pushbuttons which appear in the lower right-hand corner of a dialog are used to end the dialog. These pushbuttons do not have an accelerator key shown, although the **Esc** key can be used as an accelerator key for two of them. There's always one or more of the following:

<OK>

Choosing < OK > indicates that you've finished making changes to the information in the dialog and you want the program to accept the information. The information you've entered is checked. If it's correct the dialog is ended. If any of the information is incorrect, a warning or error message is displayed and you're returned to the dialog to correct the information or cancel the input.

< Cancel >

Choosing < Cancel> abandons any changes you've made to the information in this dialog and ends the dialog. The **Esc** key can be used as an accelerator key for this pushbutton.

< Exit >

This pushbutton appears in dialogs for which no information needs to be entered; for example, on dialogs which allow you to view information on particular items. Choosing this pushbutton causes you to exit from the dialog. The **Esc** key can be used as an accelerator key for this pushbutton, since it never appears in combination with the < Cancel > pushbutton.

The User Interface

Check Boxes

Check boxes are used to specify options which are either on or off. They are shown as a pair of square brackets [], with or without an X between the brackets. The dialog below contains a check box that specifies whether or not the link service uses a constant carrier.

```
File  Services  3270  APPC  LUA  RJE  Diagnostics  F1=Help
                                Links
----- Configure SDLC Link SDLC LINK -----
Description . . . . . [SDLC Link Service.....]
Device name . . . . . [sna_SDLC]
Port number . . . . . [001]
Line type:
  (o) Leased
  ( ) Switched auto-dial (Hayes AT)
  ( ) Switched manual dial
[ ] Constant carrier
                                     < OK > <Cancel>
Copyright (C) 1993 CLEO Communications
```

When an X is displayed in the brackets, the item is selected. When the brackets are empty, the item is not selected. In the example above, the check box is empty and the option is not selected (the link service does not use a constant carrier).

Choosing a Check Box

To choose a check box from the keyboard, do the following:

1. Press **TAB** to move to the check box.
2. Press **SPACE** to toggle the box on or off. Alternately, the **LEFT** and **UP** cursor keys can be used to toggle the box on and the **RIGHT** and **DOWN** cursor keys to toggle the box off.

You can also use the accelerator key to move to a check box. This automatically toggles the state of the check box.

Radio Groups

Radio groups provide a set of two or more options, of which one and only one can be selected at any one time. Each option is indicated by a radio button, which appears on the screen as a pair of parentheses. In the example below, the items under Line type (Leased, Switched auto-dial and Switched manual dial) are a radio group.

```

File  Services  3270  APPC  LUA  RJE  Diagnostics  Fl=Help
                                Links
                                Configure SDLC Link SDLCLINK
Description . . . . . [SDLC Link Service.....]
Device name . . . . . [sna_SDLC]
Port number . . . . . [001]
Line type:
  ( ) Leased
  ( ) Switched auto-dial (Hayes AT)
  (o) Switched manual dial
  [ ] Constant carrier
                                < OK > <Cancel>
                                Copyright (C) 1993 CLEO Communications

```

A circle within the parentheses indicates which radio button in the group is selected. Since one and only one radio button can be selected at one time, when one button is selected any other selected button within the group is automatically deselected. In the Line type radio group above, Switched manual dial is selected.

Choosing a Radio Button

To choose a radio button from the keyboard, do the following:

1. Press TAB to move to the radio group.
2. Use the cursor keys to select the button you want DOWN or RIGHT move forward and UP or LEFT move backward. If you move forward while on the last button in the group, you move to the first radio button. In similar fashion, moving backward while positioned on the first button moves you to the last radio button.

The User Interface

You can also use the accelerator key to move to a radio button. This automatically selects that radio button.

Edit Boxes

Edit boxes are areas in a dialog where you can type information. They appear as a pair of square brackets separated by some amount of space, in which you enter data. They often contain default information, which you can either accept or replace. In the example below, the Configuration file name and Configuration file path fields are both edit boxes.

File Services 3270 APPC LUA R.JE Diagnostics F1=Help
LINKix Configurator 3.0

Save

() Update running configuration file
(o) Save as a specific filename

Configuration file name [tom.....].cfg
Configuration file path [/usr/lib/linkix/.]

< OK > <Cancel>

Configuration file : /usr/lib/linkix/com.cfg loaded
Copyright (C) 1993 CLEO Communications

In some edit boxes, a limited number of characters are shown in the edit box region, but the typing area scrolls when you type in more characters than the space allotted.

Typing Information in an Edit Box

To add or change information in an edit box, do the following:

1. Press **TAB** or use the accelerator key to move to the edit box. Existing information in the edit box is shown in reverse video.

- 2a. If you begin typing while the existing text is in reverse video, the highlighted text is all replaced with the first keystroke.
- 2b. If you use the LEFT or RIGHT cursor keys or the HOME or END keys to position the cursor within the edit box before you edit the existing text, the highlighting disappears and you manually insert and delete characters as needed.

List Boxes

A list box is a control that contains a list of items, from which you choose one item. When an item is chosen from the list, you might view or specify further information on that particular item by selecting a pushbutton such as < Zoom > or < Edit >. The following example shows a dialog which contains a single control, a list box.

File Services 3270 APPC LUA RJE Diagnostics			Fl=Help
Name	Type	Description	
[NEW]	SDLC		L
[NEW]	TR		#
[NEW]	X.25		#
[NEW]	VX25		#
SDLC LINK	SDLC	SDLC Link Service	#
			#
			:
			#
			V

< Add >
< Edit >
< Delete >
< Rename >
< Exit >

Copyright (C) 1993 CLEO Communications

Also see **Dual List Boxes** for another way in which list boxes are used.

Selecting an Item from the List Box

To choose an item from the list box, you first need to move to the list box (unless it is the only control in the dialog or it's selected by default when the dialog appears) by using the TAB key or the accelerator key shown in the list box title. One item in the list box (usually the first) is shown in reverse video, indicating that it's selected.

The User Interface

To select a different item within the list box, do one of the following:

- ◆ Use the UP and DOWN cursor keys to position the cursor on the item you want.
- ◆ Press the key corresponding to the first character in the item name; if more than one item has this character, press the key again until the cursor reaches the one you want.

Once the item is selected, you typically carry out some action by choosing a pushbutton.

A list box may contain more items than the box can show at one time. To view items which are not currently shown, do one of the following:

- ◆ Use the UP or DOWN cursor keys to move the cursor past the top or bottom of the list box; the items in the box scroll down or up
- ◆ Use the HOME or END keys to move the cursor to the first or last item in the list box
- ◆ Use the PAGEUP or PAGEDOWN keys to move up or down through the list box items

If the item is not currently visible in the list box, you can still select it by using the key corresponding to the first letter of its name. (In the example on the previous page, you could enter the Session Number.)

Elevator Bar

The elevator bar (shown on the right-hand edge of the list box) shows your current position in the list. If the list box is currently displaying the first few items in the list (or all the items, if the list is short enough for all the items to appear at once), the marker (a reverse-video space character) appears at the top of the right-hand edge of the box. As you scroll down the list box, the marker moves downward to indicate your approximate position in the list.

List Boxes with Associated Edit Boxes

In the example below, the list box oUser namei has an edit box associated with it.

File Services 3270 APPC LUA RJE Diagnostics Fl=Help

3270 Users

Add a 3270 User

User Name		
[User Defined]	^	[.....]
gerhard	█	
nancyg	#	
tina	#	
davev	#	
nancyw	#	
daven	#	
chris	#	
TBD	v	

< OK > <Cancel>

Copyright (C) 1993 CLEO Communications

Where a list box has an associated edit box, you can specify an entry either by selecting it from the list box or by typing it into the edit box. In this example, you would normally specify the name of a user by selecting it from the list box. However, you could also select [User Defined] from the list box and type the user name into the edit box.

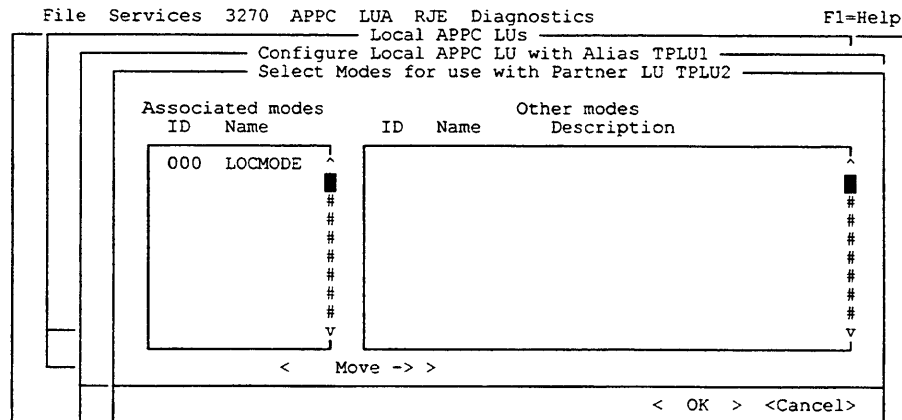
To select an entry from the list box, follow the same steps as for a normal list box. The data shown in the edit box corresponds to the selected entry in the list box, and is disabled (grayed) to indicate that you cannot alter it.

To type an entry in the edit box, follow these steps:

1. Select the entry [User Defined] that appears at the top of the list box. The edit box is cleared, and is enabled (not grayed) to indicate that you can type into it.
2. Use the TAB key to move to the edit box.
3. Type the information you want in the edit box.

Dual List Boxes

Dual list boxes allow you to select one or more items from a list. They appear as two list boxes from which you select an item and move it from one list box to the other by choosing the < Move > pushbutton that appears below or between them. In the example below, Associated modes and Other modes are a pair of dual list boxes.



1. Use the **TAB** key (or the accelerator key shown in the list box title) to move to the list box which contains the item you want to move.
2. Select the item you want to move from this list box (see the previous section for details of how to select an item from a list box). The arrow next to the word **Move** points in the direction in which you want to move the item (away from the list box containing the selected item and toward the list box to which you want to move it).
3. **TAB** to the **<Move>** pushbutton. (You cannot use the accelerator key because you're in a list box.)
4. Press **ENTER**, or the accelerator key of the **< Move>** pushbutton, to move the item. It disappears from one list box and appears in the other.

Accelerator Keys

Each field in a dialog has an accelerator key, which is indicated by a highlighted character in the name of the field. You can move directly to another field in a dialog by pressing the key which corresponds to this character.

However, there are two situations in which accelerator keys cannot be used:

- ◆ When the cursor is in an edit box, since any character key you press is interpreted as text to be entered into the edit box.
- ◆ When the cursor is in a list box, since any character key you press is used to move the cursor to the next item in the list box that begins with that character.

To use accelerator keys in these situations, you must first use the **TAB** or **BACKTAB** keys to move the cursor to a field other than an edit box or list box.

There are also two other accelerator keys which are not indicated by highlighted characters:

- ◆ The **ENTER** key chooses the current default pushbutton (the one with the highlighted angle brackets)
- ◆ The **Esc** key is an accelerator for either of two pushbuttons < Cancel > or < Done >. (These two pushbuttons never appear on the same dialog.)

Other Interface Features

Disabled (Grayed) Items

Sometimes menu options or dialog items are disabled and cannot be selected. An option or item is disabled when it is not applicable for the current situation or when it cannot be changed. For instance, dialog items may not be available because of previous selections you made, such as selecting a particular check box, and some options may not be available in certain product packages.

Disabled options appear grayed (or dimmed), except on a monochrome screen which cannot show grayed or dimmed text. If you are using a monochrome screen, however, items which are disabled (grayed) have no accelerator keys showing.

On-line Help Screens

On-line help screens are available whenever the menu interface is active. A help screen is provided for each of the following:

- ◆ The main screen of each program – describing the function of the program and the menus available from the main screen.
- ◆ Each menu in the main screen's menu bar – describing the overall function of the menu and a summary of the options available in this menu.
- ◆ Each menu option – giving more details of the functions provided by this menu option.
- ◆ Each dialog – explaining the overall purpose of the dialog, each field in the dialog and the valid responses to each field.

To invoke the help screen for a particular dialog or menu item, press the **F1** key. The help screens are context-specific; that is, the information displayed relates to the particular dialog or menu item on which the **F1** key was pressed.

The example below shows the help screen for the File menu.

File	Ser	File		elp						
Use the File menu to open new or existing configuration files, to save changes made to configuration files, or to exit the configuration program. The following options are available from the File Menu: <table> <tr> <td>Open</td> <td>Open an existing configuration file. Use this option to modify an existing configuration.</td> </tr> <tr> <td>New</td> <td>Open a new configuration file. Use this option to create a new configuration.</td> </tr> <tr> <td>Save</td> <td>Save changes to the current configuration file.</td> </tr> </table>					Open	Open an existing configuration file. Use this option to modify an existing configuration.	New	Open a new configuration file. Use this option to create a new configuration.	Save	Save changes to the current configuration file.
Open	Open an existing configuration file. Use this option to modify an existing configuration.									
New	Open a new configuration file. Use this option to create a new configuration.									
Save	Save changes to the current configuration file.									
		< Next >	< Previous >	< UI Info >						
		< OK >								

Perform file operations

When there's more help information than can redisplayed on the screen at one time, the <Next> and <Previous> pushbuttons can be used to display the next or previous page of information. To return to the dialog from which the help screen was called, choose <OK> or press the **Esc** key.

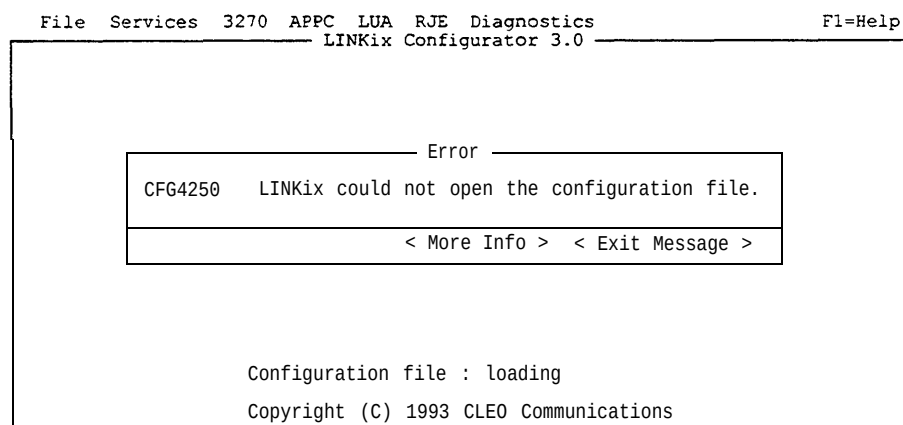
The <UI Info> pushbutton can be used to display help information on the LINKix user interface. As with the other help screens, you can page up and down through this information using the <Next> and <Previous> pushbuttons. To return to the original help screen, press the <OK> pushbutton or the **Esc** key.

While a help screen is displayed, you can press **F1** again to display a screen which gives more information about the help facility. Choosing <OK> or pressing the **Esc** key returns you to the help screen you were previously viewing.

The User Interface

Message Boxes

Message boxes display various messages as you use the menu interface. A typical message box is shown below.



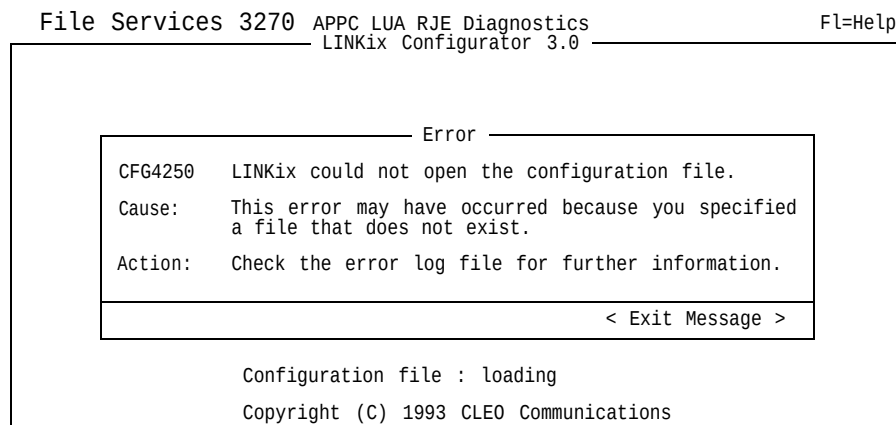
There are three different types of message boxes, as follows:

Error	Error messages indicate a condition that requires action. This may be an error detected by LINKix, for example in a file you were attempting to open. It may also be an error that you made, such as typing invalid data in an edit box and choosing <OK>; you must return to the edit box and correct the entry, or choose <Cancel> to abandon the dialog.
Warning	Warning messages indicate a condition that may require action. For example, if you load a 3270 style file that uses color when your terminal does not support it, a warning message informs you that the color mappings will be reset to their default values; you can either continue with the default mappings, or load another style file.
Information	Information messages provide conflation that an action has been accepted (such as starting a file transfer process), or provide

information related to the previous action that may be useful to you. No action is required.

The title of a message box indicates whether it is an Error, Warning or Information message. For all types of message box, choose the <Exit Message> pushbutton to acknowledge the message and continue with the program.

For Error and Warning message boxes, there is also a <More Info> pushbutton. If you choose this pushbutton, the message box is replaced by a larger box, such as the example below:



This box includes the message text that was shown in the original message box, and also more detailed information on the condition that caused the message and the action you should take. Again, choose the <Exit Message> pushbutton to acknowledge the message and continue with the program.

Screen Redrawing

While using the menu interface, your screen may become corrupted if other processes write data to it or if characters are lost due to line transmission problems. To redraw the screen, you can use the keystroke CTRL + R or CTRL + L from any menu or dialog box. This does not affect data you have entered.

Fast Exit from Menu Interface Programs

While using any of the LINKix menu interface programs, you can use the **F3** key to perform a fast exit from any point in the menu interface, without having to close any displayed displayed and return to the main screen.

You are prompted to save the configuration file or the style file before exiting. You can choose to save and exit to exit without saving or to abandon the exit.

If you choose to save and exit, the exit process stops if it encounters a condition that requires a pop-up message (Error, Warning or Information); you can then take any action required, before pressing **F3** again to continue the exit.

If you were creating a new configuration file or style file, the Save as dialog appears, to let you specify the name and path for the file. The program then saves the file and exits to the UNIX command prompt.

For the 3270 emulation program, note that the **F3** fast exit applies only within the menu interface, and cannot be remapped. From 3270 emulation, use the 3270 key EXIT (default keystroke **CTRL + X**).

3. Software Installation

The information that follows explains how to install and remove the LINKix software package. Information on setting certain hardware settings (IRQ and I/O address) is also included.

Perform Installation and Removal in Single-UserMode

Before you install or remove the software, we strongly recommend that your UNIX system be running in single-user mode (also known as system maintenance mode) - failure to do so may cause corruption of vital UNIX system files. If the UNIX computer is currently running in multi-user mode (or you're unsure of the mode) perform an orderly shutdown (e.g., `/etc/shutdown`) and restart the system in single-user mode.

Installing and Configuring Multiple FIFO/SIB Adapter Boards

If you are installing multiple FIFO/SIB adapter boards, keep the following in mind:

- ◆ All FIFO/SIB boards must be configured to use *the same* IRQ - the hardware configuration script enforces this requirement, automatically setting the IRQ for all boards to that specified for board one.
- ◆ Each FIFO/SIB must be configured to use *a different* I/O address – the hardware configuration script enforces this requirement, allowing you to only specify an I/O address that is not in use by another driver.
- ◆ The board number assigned during hardware configuration must be used when specifying the port number of the link (see **Configuring Links** in chapter 5 of the *Administration Guide*) – if you're unsure which I/O address is being used by a board, use the `LINKix.cfg` utility to display the current configuration.

TIP As you configure each board in a multiple board installation, make a note of the board number and configured I/O address. Then attach a label which contains this information to the face of the adapter bracket that is visible from the rear panel of the computer's chassis.

LINKix Packages

LINKix is installed as a number of separate packages.

LINK Level

The link level package, which is the first package to be installed, consists of one of the following:

linkix_coproc	Support for Coprocessor Hardware
linkix_sdlca	Support for ADAX SDLC
linkix_sdlci	Support for IBM PS/2 MP/A Hardware
linkix_sib	Support for SIB Hardware
linkix_qllc	Support for X.25 Hardware
linkix_tkrn	Support for Token Ring
linkix_vhmod	Support for “other” Hardware

Note: More than one link level package can be installed on a single system. Additional link levels can be added and removed from the system at any time.

SNA Level

The SNA level package, which can only be installed after the link level package, consists of one of the following:

linkix_sna_321u	Support for 32 LUS
linkix_sna_641u	Support for 64 LUS
linkix_sna_1281u	Support for 128 LUS
linkix_sna_2541u	Support for 254 LUS

Note: Only one of the SNA levels above can be installed on the system at any time.

Feature Level 1

The feature level 1 packages, which can only be installed after the SNA level package, consist of one or more of the following:

linkix_3270	LINKix 3270 (3270 emulation)
linkix_3770	LINKix 3770 (3770 emulation)
linkix_appc_dev	APPC Development System
linkix_appc_rt	APPC Run-time System
linkix_lua_dev	LUA/LUO Development System

linkix_lua_rt	LUAILUO Run-time System
linkix.mgmt	Management Utilities
linkix.netman	NetView Package
linkix_slim	Client/Server Package

Note: More than one feature level 1 package can be installed on a single system.

Feature Level 2

The feature level 2 packages, which can only be installed after a particular feature level 1 package, consist of one or more of the following:

linkix_ndm	NetView DM Client (requires linkix_lua_rt)
linkix_supt_link_m	SupportLINK Master (requires linkix_appc_rt)
linkix_supt_link_s	SupportLINK Slave (requires linkix_appc_rt)
linkix_hte	HLLAPI Terminal Emulator (requires linkix_3270)

Note: More than one feature level 2 package can be installed on a single system (with the exception of the SupportLINK packages, which are mutually exclusive).

Installation Sequence

If you try to install a package before the prerequisite software is installed, an error message is displayed which lists all the software prerequisites for the package.

UnixWare

This section describes the process of installing LINKix on a system that is running UnixWare version 1.1. The following description assumes that the default installation administration file (*/var/sadm/install/admin/default*) is being used - no checking is done except to see if there is enough room to install the package and if there are dependencies on other packages. You may see minor differences in this procedure if a different installation administration file is being used or if the default file has been changed.

Installing Supplements: If one or more Supplement disks are included with the product, install the Supplement(s) *after* you have completed the initial installation of all LINKix packages.

Installing LINKix

1. Log in as root.
2. Insert the “Link level” disk into the appropriate diskette drive.
3. At the command prompt type:

```
pkgadd -d device
```

where *device* is a device identifier (such as */dev/rdisk/f 0 t*) or a device alias (such as *diskette1*). These are defined in the device table */etc/device.tab*.

The packages that are available on the diskette are displayed, and you are prompted to select the package(s) you wish to install.

4. Press ENTER to specify all.

The installation process begins. Several messages are displayed, followed by the rights legend.

5. After reading the rights legend, answer yes to continue or no to terminate the installation process.

If you answer yes to continue, the software is installed from the diskette.

After all the Link level files are installed, the installation script makes sure that the required streams drivers are installed – STREAMS is a standardized UNIX communication technology that is used by the LINKix software.

6. The installation process is now ready to add new drivers and build a new kernel. If you wish to continue press “c”, otherwise press “t” to terminate the installation process.

If you continue, the driver configurations are added to kernel. This step may take several minutes.

7. At this point in the installation process, hardware configuration information may be required. If so, LINKix will ask for several items of information (all of the following may not be needed, depending on the hardware you have installed):

- ◆ Board type
- ◆ Number of boards installed
- ◆ IRQ (interrupt vector)
- ◆ I/O address
- ◆ Shared-memory address

If the installation script prompts for any of this information, read the instructions displayed on the screen and provide the appropriate answer. This hardware specific information is then added to the kernel configuration.

8. The fired steps of the Link level installation are performed and, if all goes well, you are notified that the installation of the package was successful.

Multiple Link Levels: More than one link level package can be installed at one time on a single system (with the exception of linkix_sib and linkix_coproc, which are mutually exclusive). The additional link levels can be installed at any time by following the same instructions above.

The kernel must be linked in order to activate the link level package. This kernel linking step will take place automatically if an SNA level package is already installed; otherwise, the linking of the kernel is postponed until after the SNA level package has been installed.

9. Remove the Link level disk from the diskette drive and replace it with the “SNA level” disk set.

Software Installation

10. Using the same command as in step 3, install the SNA level files – repeating steps 4, 5 and 6. During this process the installation script asks that you verify or change the current SDLC Frame Size. *The value specified for the SDLC Frame Size must match the value of the MAXDATA parameter in the NCP Gen at the host (LINKix will not operate if the LINKix SDLC Frame Size and the MAXDATA value do not match).*

If you do not know what the MAXDATA value is, do not change the current SDLC Frame Size. Before using LINKix, however, you must confirm what the MAXDATA value is at the host and run the **LINKix.cfg** (p. 3-Error! Bookmark not defined.) utility to properly configure the SDLC Frame Size.

After the SNA level is installed, the kernel is relinked. This may take several minutes. (To activate the new kernel, you must reboot your system. You can wait until after all the LINKix software packages and supplements have been installed before rebooting, *just don't forget to do it.*)

11. Remove the SNA level disk from the diskette drive and replace it with any Feature level 1 package diskette. Continuing in the same manner as above, install all the feature level 1 packages that you have purchased (skipping steps 7 and 8 because they do not apply).
12. Install any feature level 2 packages (skipping steps 7 and 8 because they do not apply).
13. After you have installed all of the LINKix packages, you should install any supplements that have been included with your product- Follow whatever supplemental installation instructions may have been provided.
14. Reboot your system.
15. Perform the necessary setup steps described under **Additional Setup Requirements** (p. 3-13).

Removing LINKix

This section describes the procedure for removing the LINKix product.

1. Log in as root.
2. At the command prompt type:

```
pkginfo | grep -i linkix
```

If one or more LINKix packages is installed on the system, information similar to the following will be displayed:

system	linkx3270	linkix_3270, Feature Level 1 (3. 0 .1.0)
system	sib	linkix_sib, Link Level (3 .0. 1.0)
system	sna2541u	linkix_sna_2541u, SNA Level (3.0.1.0)

3. Using the information from step 2, at the command prompt type:

```
pkgrm shortname shortname ...
```

where *shortname* is one or more product names that appear in column two of the *pkginfo* display. For example, to remove the 3270 emulation package, the SNA level package and the link level package listed above:

```
pkgrm linkx3270 sna2541u sib
```

Because the LINKix software is installed as a number of separate packages, you must remove each package separately. With the exception of multiple link levels (described below), each package must be removed in the reverse order of installation: all feature level 2 packages first, followed by feature level 1, then SNA level and, finally, link level.

Multiple Link Levels: If more than one link level package is installed, you can remove all but one link level at any time. The last link level, however, can only be removed after the SNA level has been removed.

As the removal process is carried out, various messages will be displayed on the screen. Also, if you are removing one or more LINKix packages, you must confirm the removal of each package.

Software Installation

4. When removing the SNA level, you have the option to postpone linking the kernel. You can answer no to the “link now” prompt if you are removing the entire LINKix product, as the kernel will automatically be linked when the link level is removed.
5. If multiple link levels have been installed, the removal script allows you to postpone the kernel linking until you have removed as many of the link levels as you need or choose to. However, the kernel will always be linked automatically after the last link level is removed.
6. If you have removed the SNA or link level package, you must reboot your computer in order to activate the kernel changes.

SCO UNIX System V/386

The following information describes how to install and remove the software under SCO UN System V/386 Release 3.2v4.2.

Installing LINKix

Installation of the software package is performed with the custom utility. To install the software on your computer, follow these steps:

1. Log onto the UNIX system as root. See the note on page 3-1 about running in single-user (system maintenance) mode versus multi-user mode.
2. Type custom and press **ENTER**.

To install the product from a device other than the default (device 0), custom can be invoked with the -m switch, which specifies the device from which to install the product and overrides the default.

For example: `custom -m /dev/rdisk/flq18dt`

See your UNIX system documentation for complete details.

3. Custom's main menu is displayed, with a list of currently installed products.
 - ◆ If the package you're about to install appears in the list, see the next section, **Removing LINKix**, before you continue. A currently installed product must be removed (using *custom*) before you install a new copy.
 - ◆ If the package that you're about to install is not included in the list, select the Install option.
4. Custom's Install menu appears. Select the New Product option.
5. Custom displays a menu that allows you to install all or part of the product. Select the Entire Product option.
6. You are prompted to insert Volume 1. Insert Volume 1 of the Link Level package in the diskette drive and press **ENTER**.

Software Installation

7. After a short pause the LINKix rights legend is displayed, and you are asked if you wish to continue. If after reading the rights legend you wish to continue, answer yes; otherwise, answer no to terminate the installation process.

8. If you continue, you are prompted to insert the appropriate volume. Press **ENTER** to continue.

If you are installing a multi-volume package, you are prompted to insert the subsequent volumes as needed. Do so and press **ENTER**. This process continues until all files have been extracted from all Volumes.

9. After the files from the last volume have been installed, the installation script displays the message "Checking for streams..." - STREAMS is a standardized UNIX communication technology that's used by the software. If you've already installed a STREAMS-based product on your system (e.g., TCP/IP), the streams driver may already be enabled.

10. If you're installing a Link Level, you'll see a message saying that the installation process will add new drivers and build a new kernel. At this point, LINKix will ask for several items of hardware configuration information (all of the following may not be needed, depending on the hardware you have installed):

- ◆ Board type
- ◆ Number of boards installed
- ◆ IRQ (interrupt vector)
- ◆ I/O address
- ◆ Shared-memory address

If the installation script prompts for any of this information, read the instructions displayed on the screen and provide the appropriate answer. If you don't have the necessary information, you can terminate the install process.

When all hardware configuration information has been entered, the data is added to the UNIX kernel and the kernel is rebuilt. This may take several minutes. You should then see the message "UNIX kernel has been rebuilt."

11. Press **ENTER**. Custom completes the installation process, and you are returned to the main menu. Remove the Link Level disk.

Multiple Link Levels: More than one link level package can be installed at one time on a single system (with the exception of linkix_sib and linkix_coproc, which are mutually exclusive). The additional link levels can be installed at any time by following the same instructions above.

The kernel must be linked in order to activate the link level package. This kernel linking step will take place automatically if an SNA level package is already installed; otherwise, the linking of the kernel is postponed until after the SNA level package has been installed.

12. Insert volume 1 of the SNA Level package. Select the Install option and repeat the installation instructions at steps 4 through 11 (skipping step 10 because it does not apply).

During this process the installation script asks that you verify or change the current SDLC Frame Size. **The value specified for the SDLC Frame Size must match the value of the MAXDATA parameter in the NCP Gen at the host (LINKix will not operate if the LINKix SDLC Frame Size and the MAXDATA value do not match).**

If you do not know what the MAXDATA value is, do not change the current SDLC Frame Size. Before using LINKix, however, you must confirm what the MAXDATA value is at the host and run the *LINKix.cfg* (p. 3-15) utility to properly configure the SDLC Frame Size.

13. After the files from the last volume of the SNA Level package have been installed, the kernel is rebuilt. This will take several minutes
14. When you return to custom's main menu, remove the SNA level disk from the diskette drive and replace it with any Feature level 1 package diskette. Continuing in the same manner as above, install all the feature level 1 packages that you have purchased (skipping step 10 because it does not apply).
15. Install any feature level 2 packages (skipping step 10 because it does not apply).
16. After you have installed all of the LINKix packages, you should install any supplements that have been included with your product. Follow whatever supplemental installation instructions may have been provided.
17. Remove the last volume from the floppy drive and copy the sample configuration and security files, and the default style file, as explained under **Additional Setup Requirements** (p. 3-13).
18. Now perform an orderly UNIXshutdown (e.g., */etc/shutdown*) and reboot your system.

Removing LINKix

In order to remove the product, follow these steps:

Software Installation

- 1 . Log in to the system as *root*. (See the note on page 3-1 about running in single-user mode versus multi-user mode.)
- 2 . Type *custom* and press **ENTER**.
- 3 . At the main menu, select *Remove*.
- 4 . A menu of products appears. Select the applicable LINKix package.
- 5 . A second menu appears with the package displayed and highlighted. Press **ENTER** to select it.
- 6 . A message appears asking you to verify the removal. Answer *yes*.
- 7 . Several messages appear as the software package is removed.

If you're removing the SNA Level package you have the option to rebuild the UNIX kernel later (if you'll be removing the Link Level also, the kernel is rebuilt in that step).

When the kernel is rebuilt, the old kernel */unix* is copied to */unix.old* and the new kernel is copied to */uni.x*. A message is displayed indicating whether or not the kernel has been successfully linked. Press **ENTER**.
- 8 . After the final removal procedures are carried out, you return to *custom*'s main menu. Return to step 3 to continue removing packages or select *Quit*.
- 9 . Perform an orderly shutdown (e.g., */etc/shutdown*) and restart your system.

Additional Setup Requirements

Before you can begin using LINKix, you must perform all of the following steps that apply to your environment.

Copy a Configuration and Security File

You must copy one of the default configuration and security files that we've provided in *usr/lib/linkix/samples*. Depending on which features you've purchased and installed, copy one of the following pairs:

Configuration File	Security File	Description
comSNA.cfg	comSNA.sec	Defines an SDLC link and connection, no LUs.
comTKRN.cfg	comTKRN.sec	Defines a Token Ring link and connection, no LUs.
com3270.cfg	com3270.sec	Defines an SDLC link and connection, and two 3270 LUs (one display and one printer).
com3770.cfg	com3770.sec	Defines an SDLC link and connection, five RJE LUs and one RJE workstation (with a RJE printer and a RJE punch device).
comndm.cfg	comndm.sec	Defines an SDLC link and connection, a 3270 printer LU and an LUA LU for NetView DM access.
appc.cfg epic.cfg	appc.sec epic.sec	Contains definitions for local testing of the APPC/CPI-C samples provided with the APPC/CPI-C development kit.
appc 1 .cfg epic 1 .cfg	appc 1 .sec epic 1 .sec	Contains definitions for peer-to-peer testing with the APPC/CPI-C programs <i>asample</i> and <i>csample</i> .
appc2.cfg cpic2.cfg	appc2.sec cpic2.sec	Contains definitions for peer-to-peer testing with the APPC/CPI-C programs <i>asample</i> and <i>csample</i> .

Software Installation

As an example, if you're installing LINKix 3270 and want to use the sample 3270 configuration files, you enter the following at the command line:

```
cd /usr/lib/linkix
cp samples/com3270.cfg com.cfg
cp samples/com3270.sec com.sec
```

Copy the Style File

If you're installing LINKix 3270, you should also copy the default style file:

```
cp samples/com3270.stu com3270.stu
```

This style file contains parameters that control the appearance and behavior of your UNIX terminal during 3270 emulation. These parameters determine such session characteristics as display colors, key definitions, audible signals and terminal type.

Changing Hardware Configuration With LINKix.cfg

When the LINKix software package is installed, certain hardware configuration information is required for each communication adapter board that's installed in the computer (e.g., I/O address, shared memory address, IRQ). A hardware configuration file holds this information which is downloaded to the LINKix drivers when the UNIX system is initially booted.

The *LINKix.cfg* utility allows you to make changes to the hardware configuration without having to reinstall the software package. With this utility you can view or change the configuration information for all installed adapter boards, and you can add and remove board configurations in a multi-board installation.

Starting *LINKix.cfg*

You must be logged in as user *root* to use this utility. The *LINKix.cfg* utility is started by entering the following command at the UNIX command prompt:

```
LINKix. cfg
```

When first started, *LINKix.cfg* displays the Main Menu screen:

```
Main Menu
-----

LINKix Hardware Configurator
-----
1. ) FIFO/SIB
2. ) PC/SIB
3. ) PC/XL
F. ) SDLC Frame Size

Enter your selection or q to quit:
```

From this screen you specify what type of CLEO adapter board you wish to configure by entering the appropriate number from the list. There is also an option for changing the SDLC Frame Size; this is explained in detail under **Changing the SDLC Frame Size**. To quit without making any hardware configuration changes, enter "q."

Software Installation

After selecting a hardware option from the Main Menu screen, an appropriate menu of hardware configuration options is displayed:

```
FIFO
SIB Menu
-----

SIB Hardware Configuration Options
-----
1. ) Display current SIB LINKix HW configuration
2. ) Change LINKix HW configuration
M. ) To go back to Main Menu

Enter your selection:
```

The example above shows the hardware configuration options that are available for the FIFO/SIB board.

Displaying Configuration Information

If you select the option to display a current LINKix hardware configuration, the configuration information for the installed board(s) is displayed on the screen.

```
DISPLAY FIFO SIB CONFIGURATION

Configuration for board 1 is:
Install Flag = Y          Board Type = FIFO SIB
Interrupt Priority Level = 7 (1 lowest, 7 highest)
Interrupt Sharing = Sharable Interrupt Vector with same module only
IRQ Number = 5
Start I/O Address = 0x3E0 End I/O Address = 0X3EF
Memory Address range = Not Applicable
DMA Channel Number = Not Applicable

Configuration for board 2 is:
Install Flag = Y          Board Type = FIFO SIB
Interrupt Priority Level = 7 (1 lowest, 7 highest)
Interrupt Sharing = Sharable Interrupt Vector with same module only
IRQ Number = 5
Start I/O Address = 0x250 End I/O Address = 0X25F
Memory Address range = Not Applicable
DMA Channel Number = Not Applicable

Enter <CR> to continue
```

If more than one adapter board has been installed and configured in the computer, the screen will contain the configuration settings for the first two boards. Press **ENTER** (<CR>) to display the configuration settings for additional boards, until you return to the Hardware Configuration Options screen.

If there is not an existing configuration for this board type, a “No LINKix Hardware information found for board” message will be displayed. This message should only appear if the Link level software for this board type has not been installed.

Changing a Configuration

If you select the option to change the hardware configuration, the settings for the first installed board, of the type you have selected, are displayed, along with a list of fields that may be modified:

CHANGE FIFO SIB CONFIGURATION

Configuration for board 1 is:

```
Install Flag = Y           Board Type = FIFO SIB
IRQ Number = 5
Start I/O Address = 0X3E0   End I/O Address = 0X3EF
Memory Address range = Not Applicable
```

The following HWI fields may be modified:

1.) Install Flag
2.) IRQ Number
3.) Start I/O Address
4.) Add another SIB board
5.) Remove this SIB board

Please enter field to modify for board 1 or N to change next board:

Install Flag

This parameter specifies whether or not the board is considered installed and accessible. Changing this flag from Yes to No allows you to “deactivate” this configuration (after the kernel is relinked), while changing the flag from No to Yes will “reactivate” the previously installed board (after the kernel is relinked). This change can be useful in troubleshooting specific hardware problems.

IRQ Number

This parameter specifies the Interrupt Request Line that will be used by the board. This parameter must be the same for all LINKix hardware, and in a multiple board

Software Installation

environment you can only change the IRQ for board 1. The IRQ setting for all other boards is then automatically set to the same value.

Start I/O Address

The I/O address is the channel through which LINKix accesses the PC data paths. *This value must be different for each installed board. LINKix.cfg will not allow you to select an I/O address that is already in use by another device, including another FIFO/SIB board.*

Start Memory Address

This value is only required for CLEO coprocessor adapters. The shared memory address is the location of the adapter board memory that can be accessed by the PC's central processor., multiple AT/XL2s can have the same shared memory address).

Relinking the Kernel

LINKix.cfg will automatically detect any changes made to the existing configuration and display the following message:

LINKix.cfg: You must link the kernel for these changes to become effective.
Do you wish to link the kernel at this time (y/n) ?

Unless you will be immediately performing other changes to the UNIX system that require kernel relinking, we highly recommend that you relink the kernel now and then reboot the computer to activate the new kernel. If you answer no to the relink kernel message, *LINKix.cfg* displays the command that must be used later to relink the kernel.

Multiple Board Installation/Configuration Considerations

The following information applies only in those situations where multiple LINKix communication adapter boards will be installed in a single computer. If your installation contains a single board, you can ignore the following.

LINKix can currently support four FIFO/SIB adapter boards in a single UNIX computer. When multiple communication adapter boards are added to or removed from a single system, board numbers are automatically added, removed and changed.

Adding a Board

If you are configuring a board type for which LINKix supports multiple hardware adapters in a single computer (currently, FIFO/SIB only), an option is displayed that allows you to add one or more additional hardware configurations. If more than one FIFO/SIB board is already installed, you must move to the last board's configuration before the option to add another board can be used.

If you select this option, the script asks for the necessary hardware configuration information for the next available board in sequence (e.g., if boards 1 and 2 are already configured the new board configuration is for board 3). When all necessary information has been input, the hardware information is added to the configuration and the kernel must be relinked. **(Note:** The Install Flag for the newly added board(s) is automatically set to the value of the last board in the current configuration. If, for some reason, the Install flag for the last board is set to "No," the new board(s) will also be set to "No.")

The board numbers assigned in multiple-board installations are used when configuring the LINKix software's link service – for details on configuring the link, see the appropriate section of chapter 5, **Configuring Services**.

Removing a Board

If you are configuring a board type for which LINKix supports multiple hardware adapters in a single computer (currently, FIFO/SIB only), an option is displayed that allows you to remove the currently displayed hardware configuration. **(Note:** The configuration for board 1 can never be removed, it can only be changed.)

When a board is removed, configuration information for any boards that remain will be assigned to the next lower board number. For example, if boards 1, 2 and 3 are configured and you remove board 2, the configuration for board 3 is then assigned to board 2. Because the board numbers assigned in multiple-board installations are used when configuring the LINKix software link service (see the appropriate section of chapter 5, **Configuring Services**), this kind of change must also be reflected in your LINKix configuration.

If you need to "remove" the configuration for a board while at the same time keeping higher numbered board configurations unchanged, see the earlier information on the Install flag. This flag allows an existing configuration to be deactivated without changing the board numbers assigned to any configurations that remain.

Changing the SDLC Frame Size

SDLC Frame Size and MAXDATA Parameter at Host Must Match!

When changing the LINKix SDLC Frame Size, make sure that the value you specify matches the value of the MAXDATA parameter at the host. LINKix will not operate if these values do not match.

If you select the option to change the SDLC Frame Size, the following menu screen is displayed:

SDLC Frame Size Menu

SDLC Frame Size Options

- 1.) Display *current* LINKix SDLC Frame Size
- 2.) Ch&ge LINKix SDLC Frame Size
- M.) To go back to Main Menu

Enter your select ion:

Displaying the Current SDLC Frame Size

If you select the option that displays the current SDLC Frame Size, the information is displayed as follows:

* * * * *

Current SDLC Frame Size = 265 (256 + 9) bytes

* * * * *

Enter <CR> to continue:

This example shows that the current SDLC frame size is 265, consisting of a 256 byte data segment and nine bytes of header and trailer information.

Changing the SDLC Frame Size

If you select the option that allows you to change the SDLC Frame Size, the following screen is displayed:

Select new SDLC Frame Size:

1.)	2 6 5	(2 5 6 + 9)
2.)	5 2 1	(5 1 2 + 9)
3.)	7 7 7	(7 6 8 + 9)
4.)	1 0 3 3	(1 0 2 4 + 9)
5.)	2 0 5 7	(2 0 4 8 + 9)
6.)	4 1 0 5	(4 0 9 6 + 9)
q.)	quit	

CAUTION: This value must match the MAXDATA
parameter in the host NCP Gen.

Please enter SDLC Frame Size selection:

Select the option that corresponds to the SDLC Frame Size that you need to change to or press "q" to quit. Make sure that the value you specify matches the value of the MAXDATA parameter in the NCP Gen at the host. LINKix will not operate if these values do not match.

LINKix.cfg automatically detects if you make a change to the SDLC Frame Size and displays the following message if you change this value:

```
LINKix.cf q: You must link the kernel for these changes to become effective.  
Do you wish to link the kernel at this time (y/n)?
```

Unless you will be immediately performing other changes to the UNIX system that require kernel relinking, we highly recommend that you relink the kernel now and then reboot the computer to activate the new kernel. If you answer no to the relink kernel message, *LINKix.cfg* displays the command that must be used later to relink the kernel.

Limiting User Access to Functionality

The functions provided by LINKix can be divided into two categories: system administration (which encompass configuration and program management) and user functions (e.g., 3270 emulation and APIs). With the default setup, the user functions are accessible to all users of the UNIX system, while the system administration functions are accessible to those users who are a member of the bin group. Users who are not required to have system administrator privilege should not be members of the bin group.

If you need a more restricted setup, follow one of the recommended procedures below for limiting user access.

Restricting All Functions to a Specific Group of Users

To restrict all user functions to a specific group of users, you have to change the group id for the user files (e.g., *c3270*, *com3270*) and assign all LINKix users to that group. The permissions on all user files should then be changed to allow access by owner and group only, and not by others; for example, the 3270 emulation program should have permissions *r-xr-x--* and not *r-xr-xr-x*.

Alternatively, you can limit access to 3270 emulation features by removing the <DEFAULT> user entry. When a user starts the 3270 emulation program, LINKix checks the configuration file for a 3270 user with the login name used by the user. If this name is not found, LINKix then checks for the record named <DEFAULT> and uses this. Thus, if <DEFAULT> is not defined, the only users who can run *com3270* and access the host are those that have been explicitly configured.

Restricting System Administrator Functions to Single User

To restrict system administrator functions to a single user, a single login in the **bin** group should be set up as the system administrator login. All system administrator files should be owned by this login and should not be accessible by group or others.

These files include:

/usr/bin/cconfig and */usr/bin/comconfig*

/usr/bin/comerrlog

/usr/lib/linkix.comc fg.hlp

The error and audit log files

The trace files for kernel components

The configuration and security files should have read/write access to owner and read-only access to group.

The control daemon files (*comstart*, *snadaemon* and *comstop*), must be owned by *root* and have the access permissions *r-sr-s----*

4. Configuration Overview

After the software/hardware package has been successfully installed, you need to configure the software for the specific conditions under which it will be used. To carry out this configuration, you use the separate configuration program, *comconfig*, and the sample configuration file that we've provided, *com.cfg*. We suggest you make only those changes that are necessary to establish communications with the host system. When communications have been established, then you can consider making additional changes to this supplied configuration.

The instructions in this chapter assume that you have a working knowledge of the user interface that's employed (it complies with SAA guidelines for Common User Access). If this type of interface is not familiar to you or you're uncertain about how a feature operates, complete instructions for using the interface are contained in chapter 2, The User Interface.

The most common keystrokes you'll use are:

F1	Invokes the on-line help facility, which provides context sensitive information to help you perform the correct action or furnish the proper configuration data.
TAB or CTRL + I	Moves forward through the items in a dialog.
BACKTAB or CTRL + B	Moves backward through the items in a dialog.
LEFT and RIGHT Arrow Keys	Move the highlight within the menu bar and within edit boxes. They also select radio buttons.
UP and DOWN Arrow Keys	Move the highlight within menus and within list boxes. They also select radio buttons.
ENTER	Activates the menu bar when you're at the main configuration screen, and it selects the highlighted pushbutton at the bottom of dialogs.

Overview

Configuring the software involves the following steps:

1. Configuring Services

- ◆ First you must describe the link – the type of communications adapter (board) installed in the UNIX computer.
- ◆ Next you define the connection – the logical path between your UNIX computer and the remote computer(s) you are communicating with.
- ◆ Finally, you specify local node settings – a local node manages communications with the remote system.

2. Configuring for 3270

If you've installed LINKix 3270 and will be using 3270 emulation, you will configure the 3270 sessions and users. Here you specify the display and printer sessions that will be used for 3270 emulation, configure all users on the LINKix 3270 system and assign sessions.

3. Configuring for APPC

If you've installed the APPC product and will be using APPC transaction programs, you will configure LUs (local and remote), modes, TPs (transaction programs) and security information.

4. Configuring for LUA

If you've installed the LUA product and will be using LUA applications, you will configure the LUs that will be used by LUA application programs to communicate with other computers in the SNA network.

5. Configuring for 3770 RJE

If you've installed LINKix 3770 and will be using 3770 RJE (Remote Job Entry), you will configure LUs needed for RJE, define Forms Control Buffers (FCBs) and configure the RJE workstations.

6. Saving the configuration file

Configuration Files

Two files are used with this configuration:

- ◆ The configuration file
- ◆ The security file

The configuration file can be given any filename of up to 12 characters, which includes the extension `.cfg`; the security file has the same name but with the extension `sec`. If you later save a modified configuration to an existing file, backup copies of the two files are created automatically. These backups have the same base name with the extension `.bak` for the configuration file and `.bks` for the security file.

When the control daemon is started (with *comstart*), it looks for the default configuration file `/usr/lib/linkix/com.cfg` (and the associated security file `/usr/lib/linkix/com.sec`). You can create more than one configuration, however, by saving a configuration file to a different filename and/or directory, using the Save as option under the File menu. This allows you to run different configurations at different times. To use a different configuration, you start the daemon using the `-c` option. See chapter 11, Managing LINKix Components, for complete details.

If you haven't copied the sample configuration and security files that we've provided from `/usr/lib/linkix/samples/` to `/usr/lib/linkix/` as directed in the installation procedure, do so now (see Additional Setup Requirements in chapter 3 of this manual).

Configuration File

The configuration file contains information about:

- ◆ Links – the types of communications adapters you're using to communicate with remote computers.
- ◆ Connection information - the path from the local node (the UNIX computer) to each remote node over a link.
- ◆ Local nodes - the SNA PUS (physical units) on the LINKix LAN and the connections managed by each one.
- ◆ 3270 LUs – the 3270 LUs assigned to you by your host administrator.
- ◆ 3270 LU pools – how the configured 3270 LUs are grouped together to provide backup services or to support multiple users.
- ◆ 3270 users – the 3270 users on the LINKix system.

Configuration Overview

- ◆ 3270 display or printer sessions - whether an LU will be used for a display or printer session and parameters for each type of session.
- ◆ Modes – defines the characteristics of APPC LU sessions.
- ◆ Remote APPC LUs– the LUs on other computers in the SNA network with which APPC TPs will be communicating.
- ◆ Local APPC LUs – the LUs on the UNIX computer that APPC and CPI-C applications will be using to communicate with other applications.
- ◆ Invocable TPs – APPC TPs on the UNIX computer which can be accessed by other applications.
- ◆ TP security – conversation-level security parameters for invocable TPs on the local node.
- ◆ CPI-C symbolic destination names – the information required for CPI-C applications to invoke partner TPs.
- ◆ LUA LUs – the LUA LUs assigned to you by your host administrator.
- ◆ LUA LU pools – how the configured LUA LUs are grouped together to provide backup services or to support multiple users.
- ◆ RJE LUs – the RJE LUs assigned to LINKix 3770.
- ◆ RJE Workstations – the workstations used to submit jobs to hosts.
- ◆ RJE FCBs - the Forms Control Buffers, which specify the page layouts used for printing RJE output returned from the host.
- ◆ Diagnostics – where diagnostic information will be recorded, and whether or not the network management facilities can be used.

The configuration of the local system must match the settings at the remote system. You may find it necessary to update the configuration from time to time if there are changes in the remote configuration. For example, if LUs are added to the host configuration, you will have to add their definitions here before users can access them.

Security File

The security contains security information on conversation-level and session-level security keys for APPC. When you create a configuration file, the configuration program creates a security file with the same file name as the configuration file with the extension.sec. Do not remove this file, or the product will not run.

Starting the Configuration Program

Before you begin, gather all the necessary configuration information from the host or remote computer administrator(s) – use the appropriate Site Guides in appendix A and those that were provided with any “Getting Started” packet(s) provided with the product you purchased. Remember, too, that during the configuration process help is available by pressing **F1**.

A special configuration script (cconfig) has been provided for using *comconfig* from the console. (This script sets the TERM type and remaps the keyboard for correct response before the configurator is started.) If you are configuring the program from any device other than the console, you should not use this script.

To start the configuration process:

1. Logon to the UNIX system as root.

Since the configuration and security files you’re about to create must be located in */usr/lib/linkix/*, you may wish to make that the current working directory before you proceed with the configuration process.

Upgrading From an Earlier Release?

[f you have upgraded from a LINKix release earlier than 3.0.0.0 and wish to use configuration files from the earlier release, you’ll have to convert the files to the newer format first.

- 1.** Change to the directory that contains the configuration files.
- 2.** Because the conversion utility overwrites the existing files, we recommend you copy the existing files to another area before continuing.
- 3.** Convert the files with the following command:

```
comconvcfg <cfgfile> [<cfgfile2> <cfgfile3> . . .]
```

- 4.** After the conversion utility has converted the confirmation files (over-writing the existing files), you are ready to continue with configuration.
-

Configuration Overview

2. To start the configurator from the console and open the default configuration file, enter the following command:

```
cconfig [-M] /usr/lib/linkix/com.cfg
```

To start the configurator from any other device:

```
comconfig /usr/lib/linkix/com.cfg
```

If the console device has a monochrome video display, you must include the optional -M switch when starting the program with *cconfig*. This switch prepares the console for model 2 monochrome support (by default, all LINKix products provide complete color support).

Optionally, you can start the configurator from the console without loading a configuration file by entering:

```
cconfig
```

or from any other device:

```
comconfig
```

If the following warning message is displayed:

```
File Services 3270 APPC LUA RJE Diagnostics          Fl=Help
                LINKix Configurator 3.0

Warning
CFG4151  The running configuration file.
         could not be located.
         < More Info >  < Exit Message >

Configuration file : not loaded
Copyright (C) 1993 CLEO Communications

Press Return to select a menu
```

the configuration file, `/usr/lib/linkix/com.cfg`, could not be accessed because the control daemon is not running on the master configuration server. You may continue with the configuration by loading and modifying any available configuration file.

If you start the configurator while the control daemon is running, you can open the running configuration file, make changes to it and save the changes to the file. Please note, however, that those configuration changes will not take effect until the control daemon is stopped and restarted with the changed configuration file.

When the configuration program is started, the main screen appears with the File menu already selected.

To open a configuration file using the menu system:

- a. From the File menu select the Open option. The Open dialog appears.
- b. Select the radio button to Open a specific configuration file.
- c. TAB to the edit box that contains the Configuration file name and enter the name of the file (*com*).
- d. TAB to the edit box that contains the Configuration file path and enter the path (`/usr/lib/linkix/`).
- e. Choose <OK> (press **ENTER**).

If the following message is displayed when you open an existing configuration file, (by specifying the filename on the command line or by using the Open option):

userid is no longer a valid UNIX userid.

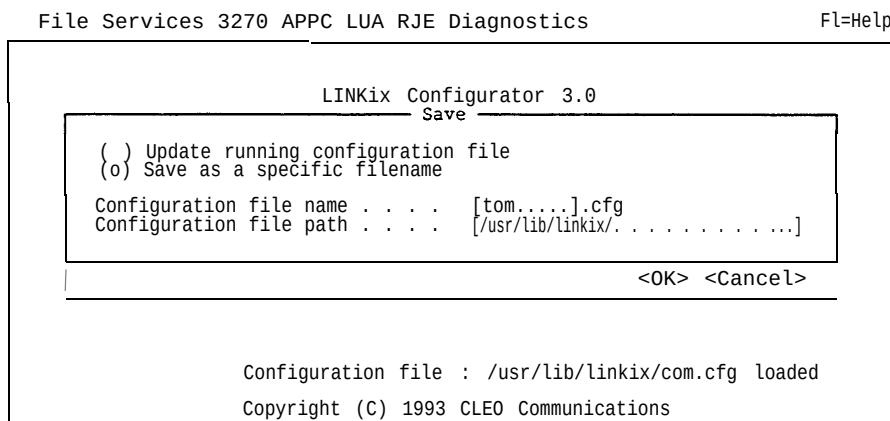
the configuration file you have loaded contains the *userid* shown, but this ID is not valid for the UNIX computer. You should either delete this user or rename to a valid UNIX userid; see **Configuring 3270 Users** later in this manual.

The configuration program has menus that are organized in the order in which the system must be configured. Using the menu bar at the top of the screen, you build your configuration by selecting menu options, working from left to right and from top to bottom. Help screens are also provided to guide you through the interfaces.

Saving a Configuration and Exiting

When you're ready to save a configuration file:

1. From the File menu select Save or Save as. If you choose Save, a conflation dialog appears that asks you to confirm the save. If you choose Save as, the Save dialog is displayed.



3. The edit boxes in the dialog will contain both the configuration file's name and its path. Change either the name or the path if you wish to save this configuration under a different name or in another area.
4. Select <OK> to save the file, or choose <Cancel> to cancel the save.

Standalone Configuration: If you change and save the running configuration file (the one being used by the control daemon), you must restart the daemon before the parameters you changed take effect.

Client/Server Configuration: In a client/server environment, more than one LINKix installation may use this configuration file. Any changes made to the running configuration file on a LINKix server may affect one or more LINKix services on this or other computers. If any LINKix services need to be restarted before the changes take effect, the services will be displayed on the screen as shown in the following illustration.

```
LINKIX Configurator 3.0
Service Type Name      Action Required
Local node      SDLCNODE must be restarted

CFG4261      The configuration was saved successfully.
< Exit Message >

Configuration file : /usr/lib/linkix/com.cfg  loaded
Copyright (C) 1993 CLEO Communications
```

For details on starting and stopping LINKix services refer to chapter 11 of this manual, **Managing LINKix Components**. Alternatively, if you have purchased the LJKix Management Utilities package, you can refer to the chapter on using the Manage program.

5. To exit from the configuration program, you:

- ◆ press **F3**, or
- ◆ activate the menu bar and from the File menu select Exit.

Change File Permissions

When a new configuration file (and its associated security file) is created, you may need to change the file permissions (via *chmod*) so that users of *com3270* have the proper read/write access to the file. Recommended access permissions are:

<i>com.cfg</i>	rw-r-----	Configuration file
<i>com.sec</i>	rw-r-----	Security file

The configuration program applies the current umask value to the file permissions when creating a new configuration file. If you've properly set the value of the umask, you will not need to manually change user permissions. See your UNIX documentation for complete details on the umask value.

Print Configuration File Contents

As part of the LINKix software package we provide a special utility, *comprintcfg*, that can be used to print a readable copy of a configuration file (e.g., *com.cfg*). The syntax for using this utility is:

```
comprintcfg cfgfile [ >output ]
```

The output from *comprintcfg* is directed to standard output unless you specifically redirect it to a file or a printer.

5. Configuring Services

This chapter describes the steps involved in configuring the link, connection and local node – services that are needed to communicate with a remote computer.

The instructions that follow assume that the configuration program has been started and the main configuration screen is displayed. If you have not yet started the configuration program, turn back to the previous chapter and read the information under *Using the Configurator*.

The default configuration file that we've provided should meet most of your startup requirements. We recommend that you initially use this file and make only those changes that are necessary to establish communication with the host system. After you've successfully established communication, then you can consider making additional changes to this supplied configuration.

The following configuration instructions assume that you'll be working with an existing configuration (the sample we've provided or another). If you need or wish to add, delete or rename a link or connection, instructions for doing so are located at the end of this chapter.

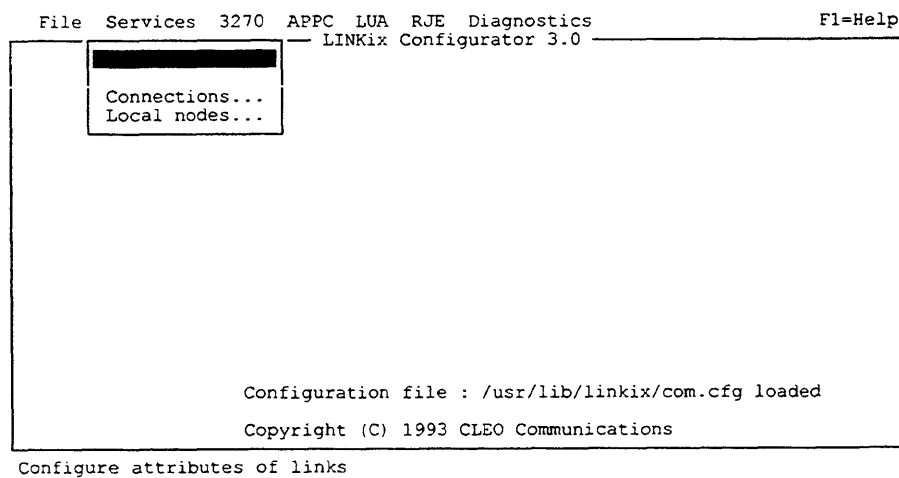
Configuring Links

Links define the physical interface between LINKix and the SNA network. When configuring the link, you provide the information needed by the communications adapter(s) in your UNIX computer to communicate with the SNA network:

- ◆ Link type
- ◆ Name of the link device file
- ◆ Which port to use for the link
- ◆ Communication parameters

If your UNIX computer contains more than one communications adapter, you must configure multiple links, one per adapter.

- 1 . Using the configuration program, open the default configuration file (tom.cjfg).
See **Starting the Configuration Program** (p. 4-5) for details.
- 2 . From the Services menu select Links.



File Services 3270 APPC LUA RJE Diagnostics

F1=Help

Links

Name	Type	Description
[NEW]	SDLC	
[NEW]	TR	#
[NEW]	x.25	#
[NEW]	VX25	#
		#
		#
		#
		#
		v

< Add > < Edit > < Delete > < Rename > < Exit >

ad

Copyright (C) 1993 CLEO Communications

Configuring Services

SDLC Link

If you are configuring an SDLC link, the dialog Configure SDLC Link dialog is displayed when you choose <Edit>.

```
File Services 3270 APPC LUA RJE Diagnostics Fl=Help
Links
Configure SDLC Link SDLC LINK
Description . . . . . [SDLC Link . . . . .]
Device name . . . . . [sna_SDLC]
Port number. . . . . [001]
Line type:
{0 Leased
  Switched auto-dial (Hayes AT)
  Switched manual dial
[ ] Constant carrier
< OK > <Cancel>
Copyright (C) 1993 CLEO Communications
```

Make whatever changes are necessary to the following options. (Those with a * beside them are the ones that are most likely to require modification.)

Description

The Description edit box contains a default description for this SDLC link. This description is optional and simply appears in the Links dialog, helping you to distinguish between various links.

Device name

The Device name field specifies the name of the device file for this link; **do not change this field**. This value is required and must match the name of the SDLC device file in the */dev* directory on the server on which you will run this link. The SDLC device file is *sna_SDLC*.

Port number

Port number indicates which communications adapter port this link uses. This value is required. Leave the value at 001, unless you have more than one adapter board installed or there is one adapter with more than a single port installed in the server and this link uses a board or port other than the first one. (Example: If three

FIFOSIB boards are installed, this field must be set to 002 to use board 2 and set to 003 to use board 3). If you're unsure which port number is assigned to a given board, use the *LINKix.cfg* utility to display the current configuration (as explained in chapter 3).

*** Line type**

The Line type radio group specifies the type of physical line being used: Leased line or Switched manual dial. Because LINKix does not currently support Switched auto-dial, this option is disabled (grayed). Your choice of Leased or Switched must match the host configuration. If you specify a Switched manual dial line, you specify a telephone number when you configure the SDLC connection.

*** Constant carrier**

The Constant carrier check box should be selected (checked) if this link is point-to-point and you are using a modem that supports constant carrier. Checking this option causes LINKix to raise RTS (Request To Send) when it is started, and LINKix keeps RTS high until it is stopped.

When the SDLC link settings contain the connect values, choose <OK> (press ENTER). You are returned to the Links dialog (p. 5-3).

Configuring Services

Token Ring Link

If you are configuring a Token Ring link, the dialog Configure TR Link dialog is displayed when you choose <Edit>.

File Services 3270 APPC LUA RJE Diagnostics Links Fl=Help

Configure TR Link TRLINK

Description [.]

Device name [sna_TR. .]

Adapter device name [.]

Port number [00]

< OK > <Cancel>

Copyright (C) 1993 CLEO Communications

Make whatever changes are necessary to the following options. (Those with a * beside them are the ones that are most likely to require modification.)

Description

The Description edit box contains a default description for this Token Ring link. This description is optional and simply appears in the Links dialog, helping you to distinguish between various links.

Device name

The Device name field specifies the name of the device file for this link; *do not change this field. This* value is required and must match the name of the Token Ring device file in the /dev directory on the server on which you will run this link. The Token Ring device file is sna_TR.

Adapter device name

The Adapter device name field identifies the particular adapter used by this link. This value must match the name of a Token Ring 802.5 or 802.3 device file in the /dev directory on the server on which you will run this link. Check with your Token Ring supplier for the name to specify here.

Port number

Port number indicates which communications adapter port this link uses, if you have more than one port available in the server. Check with your Token Ring supplier to determine whether or not this value is required. Leave the value at 0, unless you have more than one port *and* this link uses a port other than the first one.

After completing the fields in this dialog box, you are now ready to configure the Token Ring connection. Proceed to the section Configuring Connections for instructions.

Configuring Services

Vendor X.25 Link

If you are configuring a Vendor X.25 link, the dialogsConfigure Vendor X.25 Link dialog is displayed when you choose <Edit>.

```

File Services 3270 APPC LUA RJE Diagnostics
Fl=Help

Configure Vendor X.25 Link X25LINK

Description . . . . .]

Device name . . . . . sna_QLLC]

Interface card . . . . . [ . . . . .]

< OK > <Cancel>

```

Make whatever changes are necessary to the following options. (Those with a * beside them are the ones that are most likely to require modification.)

Description

The Description edit box contains a default description for this Vendor X.25 link. This description is optional and simply appears in the Links dialog, helping you to distinguish between various links.

Device name

The Device name field specifies the name of the device file for this link; **do not change this field**. This value is required and must match the name of the device file in the /dev directory on the server on which you will run this link. The device file we supply is sna_QLLC.

* Interface card

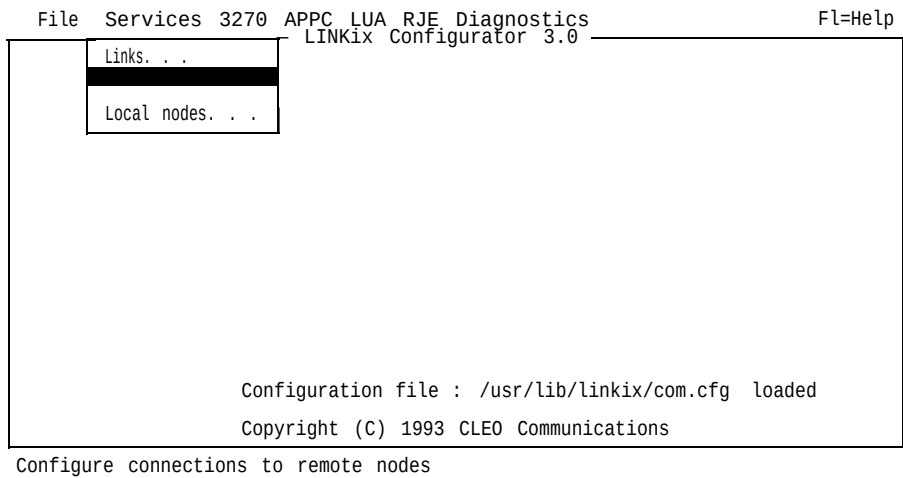
This field specifies the name of the X.25 interface card used by this link.

After completing the fields in this dialog box, you are now ready to configure the X.25 connection. Choose <OK> (press ENTER) and you return to the Links dialog (p. 5-3).

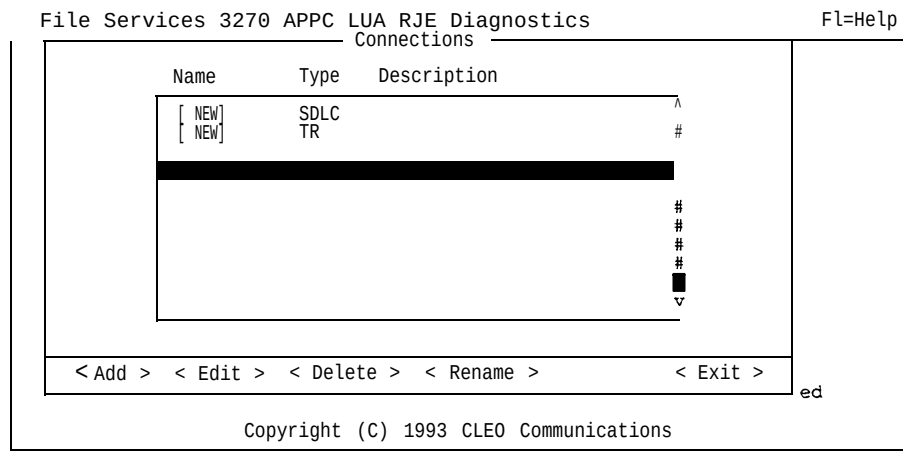
Configuring Connections

With the link configured, you’re ready to configure the connection.

- 1. From the Services menu select Connections.



The dialogConnections dialog is displayed.



Configuring Services

From this dialog you can add, edit, delete or rename one of the following connections: SDLC, TR (Token Ring), X.25. If the appropriate connection is not highlighted, move the selection bar with the cursor keys until it is highlighted.

3. Choose <Edit> (press `ENTER`). Depending on the connection type you selected, one of two different connection configuration dialogs will appear.
4. Make whatever changes are necessary to the options that are required by the connection type you selected. (Details for each type follow.)
5. When you return to the Connections dialog, `TAB` until the <Exit> pushbutton is highlighted, then press `ENTER`. You return to the main configuration screen.

SDLC Connection

If you are configuring an SDLC connection, the dialogsConfigure SDLC Connection dialog is displayed when you choose <Edit>.

File Services 3270 APPC LUA RJE Diagnostics

Fl=Help

Configure SDLC Connection SDLCCONN (not on any node)

Description [SDLC Connection]

Remote end is:

Link role:

Activation:

XID type:

☐ Host system

☐ Peer node

☐ Primary

☐ Secondary

☐ Negotiable

☐ Initially

☐ On demand

☐ By operator

☐ Format 0

☐ Format 3

Node ID to send [05D]. [FFFFF]

Node ID to receive [.. .1] - [.. .1]

Full Control point name . . [. 1. [.-1

Links usable

SDLCLINK

Other links

< Move -> >

< Parameters >

< OK > <Cancel>

The title of the dialog indicates which local node, if any, owns the connection. The connection will be owned by a local node if a local node has been configured and this connection is assigned to it, or if the connection was added using an existing connection that is already owned by a local node.

Make whatever changes are necessary to the following options. (Those with a * beside them are the ones that are most likely to require modification.)

Description

The Description edit box contains a default description for this SDLC connection. This description is optional and appears in the Connections dialog, allowing you to distinguish among your configured connections.

Remote end is

The Remote end radio group specifies whether the computer at the other end of this connection is a host computer (mainframe) or a peer computer.

Configuring Services

If you are going to use 3270, LUA or APPC to a host system on this connection, you must select Host for the remote end. If you are going to use APPC with a remote computer that is not a host on this connection, you must select Peer.

Host system	The relationship between the local and remote computers is hierarchical, with the remote computer always controlling communications. The UNIX computer is always secondary.
Peer	The two computers communicate on a peer-to-peer basis; either computer can control communications.

Link role

The Link role radio group defines the communications role of the UNIX computer with the remote computer. (This radio group is disabled (grayed) unless the remote end has been specified as a Peer Node in the previous radio group.)

Primary	The UNIX computer will always be primary and control communications with the remote computer, even if they are specified as peer systems (under Remote end is).
Secondary	The remote computer will always be primary and control communications with the UNIX computer, even if they are specified as peer systems (under Remote end is).
Negotiable	The UNIX computer and the remote computer negotiate, when the connection is activated, as to which computer is primary. This decision then remains fixed as long as the connection is active.

Contact the administrator of the remote system to determine the connect setting for this radio group.

Activation

The Activation radio group specifies when this connection should be activated.

Initially	The connection is activated as soon as the control daemon is started. It can then be deactivated or activated as required, using NetView commands or the Manage program. This is the recommended setting.
On demand	The connection is activated automatically when required by a session that uses this connection and is deactivated automatically when all sessions using it have ended.

By operator The connection is started manually when required, using NetView commands or the Manage program. The connection can then be deactivated or activated as required.

Note that “Initially” and “By operator” can both be started or stopped with NetView commands or the Manage program. The only difference is that “Initially” active connections are automatically activated when the control daemon is started.

If you activate or deactivate an “On demand” connection using NetView commands or the Manage program, this effectively changes the connection’s activation method to “By operator”; that is, the connection can only be activated or deactivated using NetView commands or the Manage program and is no longer automatically activated or deactivated as required by sessions. This change remains in effect until the control daemon is restarted.

XID type

This radio group specifies the type of XID messages being sent. MD messages identify the sending computer to the rest of the network.

XID type 3 formatted messages send up to 100 bytes of identifying information, including the Node ID and Control Point Name. XID type 0 formatted messages send only the Node ID. The default XID type is 3.

It’s important to verify with your host administrator that the host software can accept format 3 XIDs. Format 3 XIDs can cause problems for some versions of the host network control program (NCP). If the host software cannot accept them, or if you cannot determine the type used by the host, you must specify format 0 XIDs.

Note that connections configured for format 0 XIDs cannot be used by independent APPC LUs. (Independent LUs have an LU number of 0 in the host system VTAM.)

* Node ID to send

This edit box contains the ID you want this node to send to the remote computer – an eight-digit hexadecimal number that’s divided into: a three digit block number and a five digit node number. This value is required. Obtain this value from your remote administrator. The reserved values 0x000 and 0xFFFF cannot be used for the block number, and the reserved value 0x00000 cannot be used for the node number.

This edit box and the Node ID to receive edit box specify the IDs that local and remote computers may exchange when establishing communications. These IDs provide link-level security for both systems and establish which computer is primary.

Configuring Services

If this connection uses a leased line, the host will not initiate an exchange of IDs - the default values are acceptable in this case. If this connection uses a switched line, the values for block number and node number must match the values specified in the IDBLK and IDNUM parameters in the VTAM definition.

* Node ID to receive

This edit box contains the node ID expected from the remote computer before it establishes a session – an eight-digit hexadecimal number that's divided into: a three digit block number and a five digit node number. If you're strictly communicating via 3270 emulation, you do not have to enter a value for this field and any ID sent from the host will be accepted.

The reserved values 0x000 and 0xFF F cannot be used for the block number, and the reserved value 0x00000 cannot be used for the node number.

Usages of Node ID to Receive:

- ◆ Required if communicating to a host using a switched or leased line if the host is running VTAM Version 3 Release 2 or newer where XID=YES in the NCP PU definition - check with your host administrator for applicability.
- ◆ If using a multi-point configuration, this parameter maybe used by all, some or none of the remote stations – check with the remote administrator(s).

Full control point name

The Full control point name is the name by which the remote computer knows the local node that owns this connection. These optional fields may contain one to eight alphanumeric characters. When connecting to a host, the name you specify here should match the PU name in the host's PU macro – obtain this name from your host administrator.

If you leave the first part of the Full control point name blank, LINKix uses the Name of network field from the local node that owns this connection, if there is one. If there is no name configured for the node, or if you enter an asterisk (*) for the first part of the name, LINKix uses only the second part of the name. If you specify the first part of the name, you must also specify the second.

Links usable/Other links

The Links usable/Other links dual list boxes display the link that can currently be used by this connection and those that are available to be added. The Links usable list box requires at least one link and, by default, contains the configured SDLCLINK.

To remove the link from the Links usable list, select the link and choose <MoveÆ>. To add a link to the Links usable list, select the link you want to add from the Other Links list box and choose <ÆMove>.

When all the options contain the correct information, you're ready to set the parameters. Choose <Parameters> (press ENTER). The Configure SDLC Parameters dialog is displayed.

```

File  Services  3270  APPC  LUA  RJE  Diagnostics  Fl=Help
                  Configure SDLC Connection  SDLCCONN

  Configure SDLC Parameters

Encoding:          Poll address . . . . . [C1]
  (o) NRZ
  ( ) NRZI          No response timers/retries:
                    Data/Poll timeout . . . . . [...] 0.1s
  (o) Half-duplex   Data/Poll retry limit . . . . . [...]
  ( ) Full-duplex   Contact timeout . . . . . [...] 0.1s
                    Contact retry limit . . . . . [...]

Data rate select:  Poll rate . . . . . [...] per s
  ( ) Low
  (o) High          Switched connection
  [ ] Select standby establishment timeout . . . . . [...] s

Dial data:
  [.....]

                                < OK > <Cancel>

```

Make whatever changes are necessary to the following options. (Those with a * beside them are the ones that are most likely to require modification.)

* Encoding

This radio group specifies your modem's encoding scheme for transmitting and receiving data. Specify either NRZ (Non-return to zero) or NRZI (Non-return to zero inverted). This field applies only to outgoing calls. The encoding for incoming calls is specified as part of the link usage parameters, not the connection.

This option's value must match the value specified in the host's LINE/GROUP macro in VTAM – obtain this value from your host administrator. Your system must use the same encoding scheme as the remote computer.

Configuring Services

* Half-or Full-duplex

This radio group specifies whether you're using a half-or fill-duplex modem. If the link does not have Constant Carrier enabled (in the Configure SDLC Link dialog) this option is disabled (grayed).

Data rate select

This radio group is disabled (grayed) and not available for selection.

Select standby

This check box is disabled (grayed) and not available for selection.

* Dial data

This edit box contains a telephone number and, possibly, modem commands. If you're using a switched manual dial line, enter the phone number that needs to be manually dialed and that number will be displayed as part of a message that appears when the connection is started.

* Poll address

The Poll address edit box specifies the polling address used in communications. Since the connection is communicating with a host system, the local poll address must match the poll address specified in the ADDR= parameter in the PU macro of VTAM definition. Obtain this value from your host administrator.

No response timers/retries:

If you specified peer node and the link role as either negotiable or primary, you must enter values in the following edit boxes.

The information in these edit boxes determines how many times the local computer will try to contact or send data to the remote computer before determining that the connection has either been broken or cannot be made. It also specifies how long the local computer will wait for a connection to be made or between responses from the remote computer before deeming the connection is broken.

If the UNIX computer is configured to be secondary in communications, these edit boxes are disabled (grayed) and you cannot enter values in them.

Data/Poll Timeout

This edit box specifies the amount of time, in tenths of a second, that you want the local computer to wait for a response to a poll from the remote computer after the connection has been established, before retrying the poll.

The value may range from a minimum of 1 to a maximum of 100.

This parameter should be set according to the line speed of your modem connection. If this parameter is set at too low a value, you may have connection problems. A safe value is the time it takes to transmit three maximum length SDLC frames (267 bytes each).

The following table shows recommended settings for this parameter based on the line speed of your modem connection.

Recommended Settings	
Line speed (in baud)	Data/Poll Timeout setting
1,200	50
2,400	25
4,800	15
9,600	10
19,200	10

Data/Poll Retry Limit

This edit box specifies the number of times the system should poll or send data to the remote computer before considering the connection to be broken. The value may range from a minimum of 1 to a maximum of 255.

Contact Timeout

This edit box specifies the amount of time, in tenths of a second, the local system will wait while trying to connect to the remote computer before retrying.

This value may range from a minimum of 5 to a maximum of 100.

Contact Retry Limit

This edit box specifies the number of times the local computer will attempt to contact the remote computer before deciding that the connection cannot be made. This value may range from a minimum of 1 to a maximum of 20.

Poll rate

The Poll Rate edit box specifies the number of polls to be sent per second. This value may range from a minimum of 1 to a maximum of 50.

If the local computer is always secondary in communications, this option is disabled (grayed) and cannot be used.

Configuring Services

Switched connection establishment timeout

When the connection uses a switched SDLC line, this field holds a value that determines the amount of time in seconds (10-500) that the modem should need to connect with the remote computer. This value is also required if you're using a manual-dial line. This is the amount of time to wait for the number to be dialed and a connection to be made. If a connection is not made in the time specified, the system assumes that a connection cannot be made. This parameter is ignored by incoming calls.

When the parameters are set appropriately, choose <OK> (press `ENTER`). You return to the Configure SDLC Connection dialog. `TAB` to the <OK> pushbutton and press `ENTER`. You return to the Connections dialog (p. 5-9).

Token Ring Connection

If you are configuring a Token Ring (TR) connection, the Configure TR Connection dialog is displayed when you choose <Edit>.

File Services 3270 APPC LUA RJE Diagnostics F1=Help

Configure TR Connection TRCONN (not on any node)

Description [.....]

Remote end is:

Link role:

Activation:

☐ Host system

☐ Peer node

☐ Primary

☐ Secondary

☐ Negotiable

☐ Initially

☐ On demand

☐ By operator

Node ID to send

[05D].[FFFFF]

Node ID to receive

[...].[.....]

Full control point name

[.....].[.....]

Links usable

^

v

TRLINK

^

v

Other links

^

v

< Move -> >

< Parameters >

< OK >

<Cancel>

The title of the dialog indicates which local node, if any, owns the connection. The connection will be owned by a local node if a local node has been configured and this connections assigned to it or if the connection was added using an existing connection that is already owned by a local node.

Make whatever changes are necessary to the following options. (Those with a * beside them are the ones that are most likely to require modification.)

Description

The Description edit box contains a default description for this Token Ring connection. This description is optional and appears in the Connections dialog, allowing you to distinguish among your configured connections.

Remote end is

The Remote end radio group specifies whether the computer at the other end of this connection is a host computer (mainframe) or a peer computer.

Configuring Services

If you are going to use 3270, LUA or APPC to a host system on this connection, you must select Host for the remote end. If you are going to use APPC with a remote computer that is not a host on this connection, you must select Peer.

Host system	The relationship between the local and remote computers is hierarchical, with the remote computer always controlling communications. The UNIX computer is always secondary.
Peer	The two computers communicate on a peer-to-peer basis; either computer can control communications.

Link role

The Link role radio group defines the communications role of the UNIX computer with the remote computer. (This radio group is disabled (grayed) unless the remote end has been specified as a Peer Node in the previous radio group.)

Primary	The UNIX computer will always be primary and control communications with the remote computer, even if they are specified as peer systems (under Remote end is).
Secondary	The remote computer will always be primary and control communications with the UNIX computer, even if they are specified as peer systems (under Remote end is).
Negotiable	The UNIX computer and the remote computer negotiate, when the connection is activated, as to which computer is primary. This decision then remains fixed as long as the connection is active.

Contact the administrator of the remote system to determine the connect setting for this radio group.

Activation

The Activation radio group specifies when this connection should be activated.

Initially	The connection is activated as soon as the control daemon is started. It can then be deactivated or activated as required, using NetView commands or the Manage program. This is the recommended setting.
On demand	The connection is activated automatically when required by a session that uses this connection and is deactivated automatically when all sessions using it have ended.

By operator The connection is started manually when required, using NetView commands or the Manage program. The connection can then be deactivated or activated as required.

Note that “Initially” and “By operator” can both be started or stopped with NetView commands or the Manage program. The only difference is that “Initially” active connections are automatically activated when the control daemon is started.

If you activate or deactivate an “On demand” connection using NetView commands or the Manage program, this effectively changes the connection’s activation method to “By operator”; that is, the connection can only be activated or deactivated using NetView commands or the Manage program and is no longer automatically activated or deactivated as required by sessions. This change remains in effect until the control daemon is restarted.

*** Node ID to send**

This edit box contains the ID you want this node to send to the remote computer – an eight-digit hexadecimal number that’s divided into: a three digit block number and a five digit node number. This value is required. Obtain this value from your remote administrator. The reserved values 0x000 and 0xFFFF cannot be used for the block number, and the reserved value 0x00000 cannot be used for the node number.

This edit box and the Node ID to receive edit box specify the IDs that local and remote computers may exchange when establishing communications. These IDs provide link-level security for both systems and establish which computer is primary. If this connection uses a leased line, the host will not initiate an exchange of IDs – the default values are acceptable in this case. If this connection uses a switched line, the values for block number and node number must match the values specified in the IDBLK and IDNUM parameters in the VTAM definition.

*** Node ID to receive**

This edit box contains the node ID expected from the remote computer before it establishes a session – an eight-digit hexadecimal number that’s divided into: a three digit block number and a five digit node number. If you’re strictly communicating via 3270 emulation, you do not have to enter a value for this field and any ID sent from the host will be accepted.

The reserved values 0x000 and 0xFFFF cannot be used for the block number, and the reserved value 0x00000 cannot be used for the node number.

Configuring Services

Usages of Node ID to Receive:

- ◆ Required if communicating to a host using a switched or leased line if the host is running VTAM Version 3 Release 2 or newer where XID=YES in the NCP PU definition - check with your host administrator for applicability.
- ◆ If using a multi-point configuration, this parameter maybe used by all, some or none of the remote stations - check with the remote administrator(s).

Full control point name

The Full control point name edit box contains the name by which the remote computer knows the local node that owns this connection. This optional field may be 1-8 alphanumeric characters. When connecting to a host, the name you specify here should match the PU name in the host's PU macro – obtain this name from your host administrator.

If you leave the first part of the Full control point name blank, LINKix uses the Name of network field from the local node that owns this connection, if there is one. If there is no name configured for the node, or if you enter an asterisk (*) for the first part of the name, LINKix uses only the second part of the name. If you specify the first part of the name, you must also specify the second.

Links usable/Other links

The Links usable/Other links dual list boxes display the link that can currently be used by this connection and those that are available to be added. The Links usable list box requires at least one link and, by default, contains the configured SDLCLINK.

To remove the link from the Links usable list, select the link and choose <MoveÆ>. To add a link to the Links usable list, select the link you want to add from the Other Links list box and choose <ÆMove>.

When all the options contain the correct information, you're ready to set the parameters. Choose <Parameters> (press ENTER). The Configure TR Parameters dialog is displayed.

The screenshot shows a multi-level menu structure. The top menu bar includes 'File', 'Services', '3270', 'APPC', 'LUA', 'RJE', 'Diagnostics', and 'F1=Help'. The 'Connections' menu is open, showing 'Configure TR' and 'Configure TR Parameters'. The 'Configure TR Parameters' dialog box is active, displaying the following fields:

- Remote network address [000000000000]
- Remote SAP address [04]
- Retry limit [010]
- Receive ack threshold [002] frames
- Unacknowledged send limit [001] frames
- Maximum BTU length [1929]

At the bottom of the dialog box are the buttons '< OK >' and '<Cancel>'.

Make whatever changes are necessary to the following options. (Those with a * beside them are the ones that are most likely to require modification.)

* Remote network address

The Remote network address edit box specifies the address of the remote Token Ring node to which LINKix is connecting. This hexadecimal value may range from 0x000000000000 to 0xFFFFFFFFFFFF. Check this value with your network administrator.

* Remote SAP address

The Remote SAP address edit box specifies the address of the remote computer's Service Access Point (SAP). This value is required. It consists of a two digit hexadecimal number from 0x04 to 0xEC and must be a multiple of 4.

See your Token Ring network documentation for more information about SAP.

Retry Limit

The Retry limit edit box specifies the number of retries (0 to 255) the system should make if it does not receive a response when polling or sending data to the remote computer, before considering the connection to be broken.

Receive ack threshold

The Receive ack threshold edit box specifies the maximum number of frames (1 to 127) LINKix can receive from the remote computer before sending a response.

Configuring Services

Unacknowledged send limit

The Unacknowledged send limit edit box specifies the maximum number of frames (1 to 127) LJKix can send to the remote computer without receiving a response. After sending the maximum number of frames, LINKix does not send more frames until the remote computer responds.

Maximum BTU length

The Maximum BTU length edit box specifies the maximum length of a BTU (basic transmission unit). This is the number of bytes transmitted in a single data link control frame and may range from 265 to 4105.

When the parameters are set appropriately, choose <OK> (press `ENTER`). You return to the Configure TR Connection dialog. `TAB` to the <OK> pushbutton and press `ENTER`. You return to the Connections dialog (p. 5-9).

X.25 Connection

If you are configuring an X.25 connection, the Configure X.25 Connection dialog is displayed when you choose <Edit>.

File Services 3270 APPC LUA RJE Diagnostics

Fl=Help

Configure X.25 Connection X25CONN

Description [x.25 Connection]

Remote end is:

Link role:

Activation:

XID type:

☐ Host system

☐ Peer node

☐ Primary

☐ Secondary

☐ Negotiable

☐ Initially

☐ On demand

☐ By operator

☐ Format 0

☐ Format 3

Node ID to send [05D]. [FFFFF]

Node ID to receive [.. -1. [---1]

Full Control point name . . [. . . . -. .1. [. ---...1

VC type:

☐ Switched

☐ Permanent

Links usable

X25LINK

Other links

< Move -> >

< Parameters >

< OK > <Cancel>

The title of the dialog indicates which local node, if any, owns the connection. The connection will be owned by a local node if a local node has been configured and this connection is assigned to it, or if the connection was added using an existing connection that is already owned by a local node.

Make whatever changes are necessary to the following options. (Those with a * beside them are the ones that are most likely to require modification.)

Description

The Description edit box contains a default description for this X.25 connection. This description is optional and appears in the Connections dialog, allowing you to distinguish among your configured connections.

Remote end is

The Remote end radio group specifies whether the computer at the other end of this connection is a host computer (mainframe) or a peer computer.

If you are going to use 3270, LUA, APPC or CPI-C to a host system on this connection, you must select Host for the remote end. If you are going to use APPC

Configuring Services

or CPI-C with a remote computer that is not a host on this connection, you must select Peer.

Host system The relationship between the local computer and the remote computer is hierarchical, with the remote computer always controlling communications. The UNIX computer is always secondary.

Peer The two computers communicate on a peer-to-peer basis; either computer can control communications.

Link role

The Link role radio group defines the communications role with the remote computer. (This radio group is disabled (grayed) unless the remote end has been specified as a Peer Node in the previous radio group.)

Primary The UNIX computer will always be primary and control communications with the remote computer, even if they are specified as peer systems (under Remote end is).

Secondary The remote computer will always be primary and control communications with the UNIX computer, even if they are specified as peer systems (under Remote end is).

Negotiable The UNIX computer and the remote computer negotiate, when the connection is activated, as to which computer is primary. This decision then remains fixed as long as the connection is active.

Contact the administrator of the remote system to determine the correct setting for this radio group.

Activation

The Activation radio group specifies when this connection should be activated.

Initially The connection is activated as soon as the control daemon is started. It can then be deactivated or activated as required, using NetView commands or the Manage program. This is the recommended setting.

On demand The connection is activated automatically when required by a session that uses this connection and is deactivated automatically when all sessions using it have ended.

By operator The connection is started manually when required, using NetView commands or the Manage program. The connection can then be deactivated or activated as required.

Note that “Initially” and “By operator” can both be started or stopped with NetView commands or the Manage program. The only difference is that “Initially” active connections are automatically activated when the control daemon is started.

If you activate or deactivate an “On demand” connection using NetView commands or the Manage program, this effectively changes the connection’s activation method to “By operator”; that is, the connection can only be activated or deactivated using NetView commands or the Manage program and is no longer automatically activated or deactivated as required by sessions. This change remains in effect until the control daemon is restarted.

XID type

This radio group specifies the type of XID messages being sent. XID messages identify the sending computer across the X.25 network.

XID type 3 formatted messages send up to 100 bytes of identifying information, including the Node ID and Control Point Name. XID type 0 formatted messages send only the Node ID. The default XID type is 3.

It’s important to verify with your host administrator that the host software can accept format 3 XIDs. Format 3 XIDs can cause problems for some versions of the host network control program (NCP). If the host software cannot accept them, or if you cannot determine the type used by the host, you must specify format 0 XIDs.

Note that connections configured for format 0 XIDs cannot be used by independent APPC LUs. (Independent LUs have an LU number of 0 in the host system VTAM.)

* Node ID to send

This edit box contains the ID you want to send to the remote computer – an eight-digit hexadecimal number that’s divided into: a three digit block number and a five digit node number. This value is required. Obtain this value from your remote administrator. The reserved values 0x000 and 0xiFFF cannot be used for the block number, and the reserved value 0x00000 cannot be used for the node number.

This edit box and the Node ID to receive edit box specify the IDs that local and remote computers may exchange when establishing communications. These IDs provide link-level security for both systems and establish which computer is primary.

Configuring Services

If this connection uses a leased line, the host will not initiate an exchange of IDs – the default values are acceptable in this case. If this connection uses a switched line, the values for block number and node number must match the values specified in the IDBLK and IDNUM parameters in the VTAM definition.

* Node ID to receive

This edit box contains the node ID expected from the remote computer before it establishes a session – an eight-digit hexadecimal number that's divided into: a three digit block number and a five digit node number. If you're strictly communicating via 3270 emulation, you do not have to enter a value for this field and any ID sent from the host will be accepted.

The reserved values 0x000 and 0xFF F cannot be used for the block number, and the reserved value 0x00000 cannot be used for the node number.

Usages of Node ID to Receive:

- ◆ If communicating to a host using either a switched or leased line, this parameter is required if the host is running VTAM Version 3 Release 2 or newer where XID=YES in the NCP PU definition - check with your host administrator for applicability.
- ◆ If using a multi-point configuration, this parameter maybe used by all, some or none of the remote stations - check with the remote administrator(s).

* Full control point name

The Full control point name edit box contains the name by which the remote computer knows the local node that owns this connection. This optional field may be one to eight alphanumeric characters. When connecting to a host, the name you specify here should match the PU name in the host's PU macro – obtain this name from your host administrator.

If you leave the first part of the Full control point name blank, LINKix uses the Name of network field from the local node that owns this connection, if there is one. If there is no name configured for the node, or if you enter an asterisk (*) for the first part of the name, LINKix uses only the second part of the name. If you specify the first part of the name, you must also specify the second.

Virtual circuit type

The Virtual circuit type radio group specifies the type of virtual circuit used by this connection: Switched (SVC) or Permanent (PVC).

If you specified more than one link for this connection, or if you specified link(s) with only SVC channel ranges configured, this radio group setting must be SVC. If you specified a link with only PVC channel ranges configured, this radio group setting must be PVC.

If you have not yet specified a link for this connection, you can select either option. However, you will only be able to select links that have channels configured of the selected type.

You must select SVC or PVC before you choose the Parameters> pushbutton to configure X.25 connection parameters, because there are two different X.25 Connection Parameters dialogs depending on the virtual circuit type you select here.

Links usable/Other links

The Links usable/Other links dual list boxes display the link that can currently be used by this connection and those that are available to be added. The Links usable list box requires at least one link and, by default, contains the configured X.25 link.

To remove the link from the Links usable list, select the link and choose <Move>. To add a link to the Links usable list, select the link you want to add from the Other Links list box and choose <<Move>.

You can configure multiple links for use with a connection when using an SVC connection; you can configure only one when using a PVC connection. If you select more than one link for the connection, each selected link will be tried until a connection is successfully activated.

When all the options contain the correct information, you're ready to set the parameters. Choose <Parameters> (press ENTER). The Configure X.25 Parameters dialog is displayed. (There are two different X.25 Connection Parameters dialogs, depending on whether the connection is configured as an SVC or PVC connection.)

Configuring Services

SVC Connection

If you're configuring an SVC connection, the Configure X.25 Parameters dialog for an SVC connection is displayed.

File Services 3270 APPC LUA RJE Diagnostics
Configure X.25 Connection X25CONN F1=Help

Configure X.25 Parameters

Remote address [00000000000000..]

Facility data [.....]

User data [.....]

< OK > <Cancel>

Complete the fields in this dialog as necessary.

* Remote address

The Remote address edit box contains the address of the X.25 node to which you want to connect; from one to 15 digits, consisting of a 12-digit address (remaining digits are used to route incoming calls between stations using the same 12-digit address). Normally, only the first 12 digits are meaningful to the X.25 network.

This value is required. Check with your network administrator to determine the address of the remote computer.

* Facility data

The Facility Data edit box contains any optional facility data that's required by your network provider; This data is usually information required to make a connection between a local and remote computer. The contents should consist of an even number of hexadecimal digits from zero to 126.

Facility data is generally used to request non-default functionality from your X.25 network for a particular SVC connection. You will probably need to coordinate the

use of some particular facilities with your network supplier and, in some cases, with the administrator of the remote computer you are connecting to.

Facility data consists of a 2-character facility identifier followed by a sequence of parameter characters. The first character of the facility identifier indicates how many parameter characters will follow:

First character	Number of parameter characters
0 through 3	2
4 through 7	4
8 through B	6
C through F	Length given by characters 3 and 4

The second character of the facility identifier is usually specific to each facility.

A facility identifier plus its parameters constitutes a “facility request.” Many facility requests can be appended together to make a complete string of facility data. You can enter up to 63 hexadecimal bytes (126 hexadecimal characters) into the Facility Data field; identifiers should not be repeated.

Facility data requests can vary between sites; however, there are facility data requests common to most sites. Some common facility data requests are:

Facility	Identifier	Parameters
Reverse charging	01	00 Reverse charging NOT requested
		01 Reverse charging requested
Packet sizes	42	OrOs r=receive packet size s=send packet size r and s can have the following values: 6 for 64-byte packets 7 for 128-byte packets 8 for 512-byte packets

Configuring Services

Window sizes	43	OrOs r=receive window size s=send window size r and s can be between 1 and 7. They are usually set to the same value.

The send and receive in this context are from the point of view of the initiator of the virtual circuit (for example, the end that sends a CALL Request packet to start the connection). Note that, if you specify packet sizes or window sizes for a connection using facility data, the values will override those specified on the link used by this connection.

An Example

To initiate a connection using 256-byte packets, a send window of 7, a receive window of 4 and reversing the charges, the facility groups are 420808 for packet size, 430407 for window sizes, 0101 for reverse charging. The string of characters that you enter into the Facility data field is: 4208084304070101.

Some facility data is subject to negotiation by the computer receiving the initial CALL Request. The CALL Accepted packet returned from the network when the virtual circuit has been established contains facility data with the outcome of any negotiation that has occurred. For flow control facilities (such as packet and window sizes) the remote computer is only allowed to negotiate to, or closer to, the standard default values.

* User data

The User Data edit box contains any optional user data that's required by your network provider. User data is like facility data and is usually information required by the X.25 network when making a connection between a local and remote computer. The contents should consist of an even number of hexadecimal digits from zero to 32

PVC Connection

If you're configuring a PVC connection, the Configure X.25 Parameters dialog for a PVC connection is displayed.

```

File  Services  3270  APPC  LUA  RJE  Diagnostics
Configure X.25 Connection X25CONN

Configure X.25 Parameters

PVC alias . . . . . [01]

Default packet size . . . . . [128]

Default window size . . . . . [02]

< OK >  <Cancel>
  
```

Complete the fields in this dialog as necessary.

*** PVC alias**

The PVC Alias edit box contains the alias of the X.25 channel. This number identifies the appropriate PVC within the given range. This field may contain one to the maximum number of configured PVC channels. Enter 1 for the first PVC channel, 2 for the second channel, 3 for the third channel and so on.

*** Default packet size**

The Default packet size edit box contains the maximum bytes in a frame; this does not include header bytes, but only the number of data bytes sent in a frame. This value is negotiated with the X.25 network. Valid values are 64, 128, 256 and 512.

*** Default window size**

The Default window size field specifies the maximum number of frames that can be sent to the remote computer without having to wait for an acknowledgment. This value is negotiated with the X.25 network and may range from 1 to 7.

When all the options contain the correct information, choose <OK> (press **ENTER**). You return to the Configure X.25 Connection dialog. **TAB** to the <OK> pushbutton and press **ENTER**. You return to the Connections dialog (p. 5-9).

Configuring Local Nodes

With the connection configured, you're ready to configure the local node.

A local node is an addressable location on the SNA network. Each connection is assigned to a particular local node, which manages communications with the remote system using that connection.

As part of the configuration for a local node, you specify whether the local node can accept incoming calls from remote computers on particular links; that is, whether a remote computer is permitted to activate a connection to the local node.

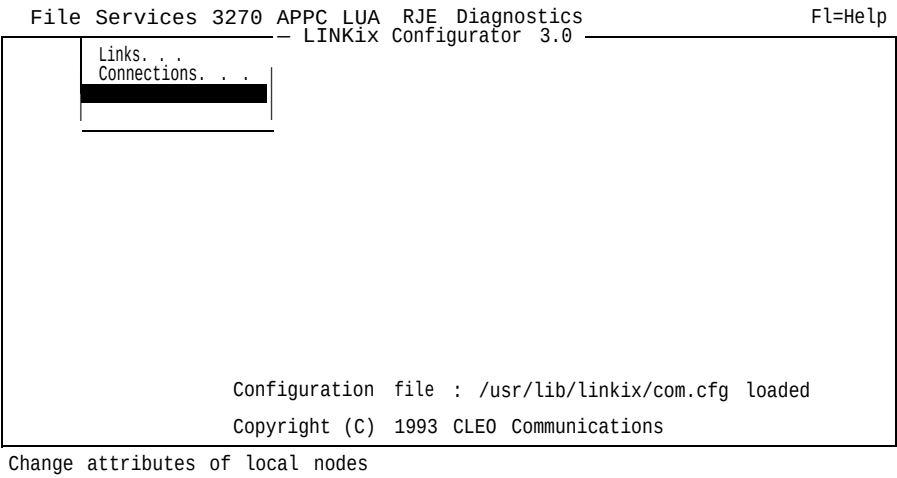
Client/Server: Multiple Local Nodes

If you have installed the optional client/server support package, you can configure up to 32 local nodes on your LINKix LAN (you need to obtain a license for each local node you use). The reasons for using more than one local node are as follows:

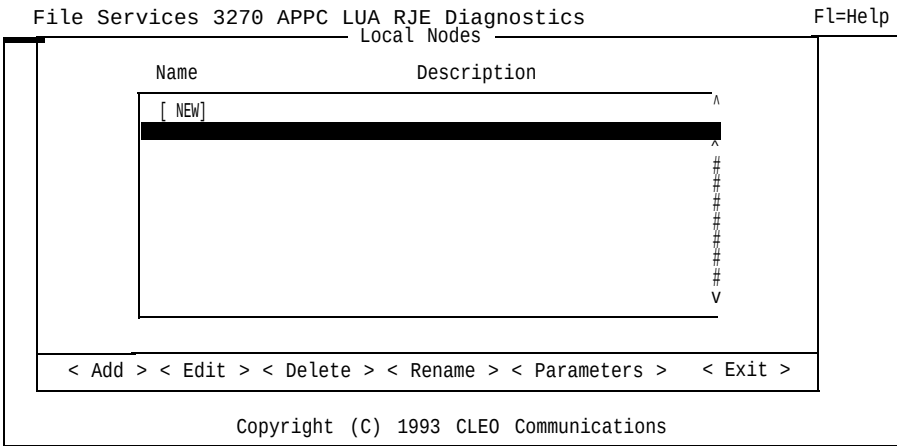
- ◆ **Load balancing:** If you have large numbers of 3270 users and/or LUA applications using large numbers of LUs, you may want to configure LU pools which contain LUs configured on more than one local node and run different local nodes on different servers. This can improve performance by sharing the processing load between the servers.
- ◆ **Hot backup:** Again, if you create LU pools across more than one local node, then users will be available to use the resources of one local node or connection even if another cannot be used because its server is temporarily unavailable.
- ◆ **Multiple links:** If you have different links running on different servers (according to the adapter type or types in each server), using a local node on the same server as the link helps to avoid unnecessary LAN traffic and can improve system performance.

To configure local nodes:

- 1. From the Services menu select Local nodes.



The Local Nodes dialog is displayed.



Configuring Services

From this dialog you can add, edit, delete, rename or specify configuration parameters for local nodes. If the appropriate node is not highlighted, move the selection bar with the cursor keys until it is highlighted.

2. Choose <Edit> (press **ENTER**). The Configure Local Node dialog is displayed.

File Services 3270 APPC LUA RJE Diagnostics Fl=Help

Configure Local Node NODE1

Description [.]

Name of network [.]

Connections used

Available connections

SDLC1
x250NE

< <- Move >

< OK > <Cancel>

3. Make whatever changes are necessary to the following options.

Description

The Description edit box contains a default description for this local node. This description is optional and appears in the Local Nodes dialog, allowing you to distinguish among your configured local nodes.

Name of network

Specify the name of the SNA network; this is used as a default for part of the Full Control Point Name for connections owned by this node, and for part of the fully-qualified LU name for local APPC LUs configured on this local node. In both cases, you can override this default by specifying an explicit name when configuring the connection or the LU.

This optional field may contain up to eight characters (A-Z, 0-9, #, \$ and @). Lowercase letters are converted to uppercase. You may wish to check with the remote system administrator for the correct information.

Connections used / Available connections

These two list boxes show connections owned by this local node and connections which are not currently owned by any local node. Use the < Move > pushbutton to move connections between the two list boxes.

A connection cannot be owned by more than one local node. You cannot use a connection unless it is owned by a local node; 3270 LUs, LUA LUs or RJE LUs configured on a particular local node must use a connection owned by that node.

If you move a connection into the Available Connections list box which is used by an APPC mode, a Confirm Updates dialog appears when you choose < OK >. You can either abandon the change so that the connection remains owned by the local node, or you can delete the modes which use the connection.

Similarly, if you move a connection which is configured as the Network Management connection into the Available Connections list, a Confirm Updates dialog appears when you choose < OK >. The connection cannot be used as the Network Management connection if it is not owned by a local node. You can either abandon the change so that the connection remains owned by the local node, or configure LINKix without a Network Management connection; in this case you must configure another connection as the Network Management connection in order to use the LINKix network management features. See the chapter 10, *Configuring Diagnostics*, for more information on configuring a Network Management connection.

4. After completing the information in this dialog, you save the changes by choosing the < OK > pushbutton or discard any changes by choosing < Cancel >. When you return to the Local Nodes dialog, you must specify the link usage parameters for this local node; these parameters determine how the local node uses different links.
5. To specify the link usage parameters, select the local node and choose the < Parameters > pushbutton.

Configuring Services

The Link Usage Parameters dialog for that node is displayed.

File Services 3270 APPC LUA RJE Diagnostics F1=Help

Local Nodes

Link Usage Parameters for NODE1

Name	Type	Incoming	Max	connections	Parameters

< Edit > < Exit >

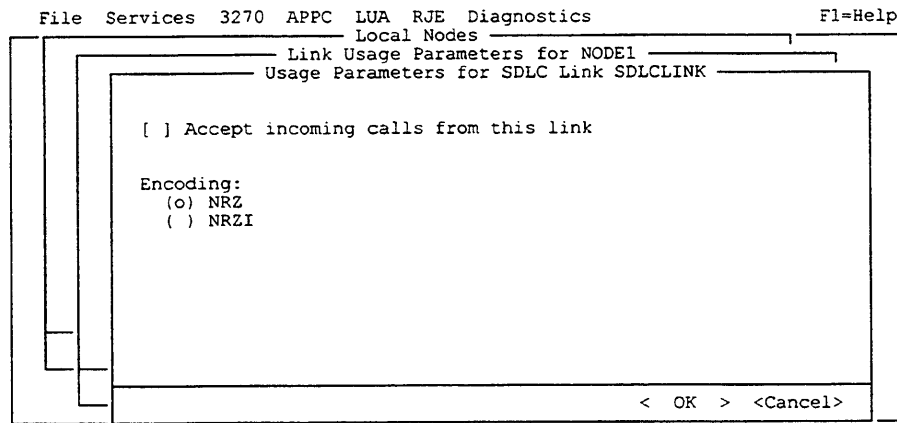
All configured links for this node are listed, with the following information:

Name	The name of the link.
Type	The type of link - SDLC, TR (TokenRing) or X.25. (The usage parameters for Vendor X.25 are the same as for standard X.25; both types appear as X.25 in this dialog.)
Incoming calls	Specifies whether the local node can receive incoming calls on this link.
Max connections	The maximum number of connections to the local node using this link which can be active at any time. For SDLC links this is always 1.
Parameters	Any additional usage parameters configured for the link.

- 6 . To change or specify parameters for a particular link, select the link and choose < Edit >. The appropriate Usage Parameters for Link dialog is displayed; there are three different Usage Parameters for Link dialog boxes, depending on the type of link (SDLC, X.25, or Token Ring).

SDLC Parameters

If the link type is SDLC, the Usage Parameters for SDLC Link dialog is displayed.



Accept incoming calls from this link

The Accept incoming calls from this link check box should be selected (checked) if remote computers can initiate connections with the local node using this link. This option is required when communicating with remote nodes that can initiate sessions. It's not applicable if you've configured the link to use a leased line.

Encoding

The Encoding radio group specifies the data encoding scheme your modem uses for incoming calls on this link: NRZ (Non-return to zero) or NRZI (Non-return to zero inverted). This field applies only to incoming calls; the encoding for outgoing calls is specified when configuring an SDLC connection that uses the link. The choice you make for this option must match the value specified in the LINE/GROUP macro in VTAM – obtain this value from your host administrator. Your modem must use the same encoding scheme as the modem at the remote computer.

Configuring Services

Token Ring Parameters

If the link type is Token Ring, the Usage Parameters for TR Link dialog is displayed.

File Services 3270 APPC LUA RJE Diagnostics F1=Help

Links
Link Usage Parameters for NODE1
Usage Parameters for TR Link TRLINK

[] Accept incoming calls from this link

Max active connections with link . . [..]

Local SAP address [04]

< OK > <Cancel>

Accept incoming calls from this link

The Accept incoming calls from this link check box should be selected (checked) if remote computers can initiate connections with the local node using this link. This option is required when communicating with remote nodes that can initiate sessions. It's not applicable if you've configured the link to use a leased line.

Max active connections with link

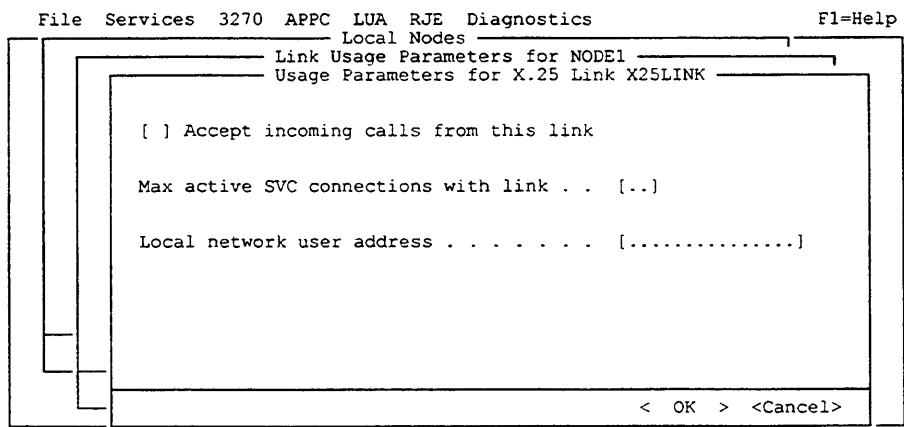
This field specifies the maximum number of active connections which this local node can have at any time using this link. This value may range from 1 to 32. Set this field to the number of connections configured to use this link. If a smaller number is specified, all of the connections cannot be active at the same time.

Local SAP address

This is the Service Access Point (SAP) address that remote nodes must specify to contact the LINKix local node. This value is required. It consists of a two digit hexadecimal number from 0x04 to 0xEC and must be a multiple of 4. The SAP address must not match the local SAP address of any other local node that can accept incoming calls from the same link. Check with the remote system administrator to determine the correct value to specify here.

X.25 Parameters

If the link type is X.25, the Usage Parameters for X.25 Link dialog is displayed.



Accept incoming calls from this link

The Accept incoming calls from this link check box should be selected (checked) if remote computers can initiate connections with the local node using this link. This option is required when communicating with remote nodes that can initiate sessions. For Vendor X.25, do not select this option unless the link has at least one SVC 2-way or SVC incoming channel; check with your X.25 administrator if necessary.

Max active SVC connections with link

This field specifies the maximum number of active connections which this local node can have at any time using this link. The value can range from zero to 15. Set this field to zero if the link is configured to use only PVC channels.

Local network user address

If you selected Accept Incoming Calls, this field specifies the address which remote nodes must specify in order to contact this local node. For Vendor X.25, check with your X.25 administrator for information on how to use this field. The value may range from zero to 15 hexadecimal digits. This field is required if you have selected the Accept Incoming Calls checkbox; otherwise it is not applicable.

Links: Adding, Deleting, Renaming

The default configuration file that we've provided defines a link that should meet most of your startup requirements. We recommend that you initially use this file and make only those changes that are necessary to establish communication with the host system. After you've successfully established communication, then you can consider making additional changes to this supplied configuration or add another link definition.

Adding a Link

When adding a link, you can use the system defaults as the basis for the new link or an existing link similar to the one you want to add (in which case the default settings will be the same as the existing link).

Note: After you add a link, that link is added to the list of available links that appear in the Links dialog.

To add a new link:

1. To add a new link using the system defaults, select the [NEW] entry for the type of link you are adding. For example, to add an SDLC link, choose [NEW] SDLC.

To add a new link which is similar to an existing link, select the existing link from the list box.

2. Choose < Add >. The Add a Link dialog (for the type of link you are configuring) is displayed. If you are using an existing link as a model, the name of this link is displayed in the dialog.

For example, if you were adding a new SDLC link based on an existing link named SDLCLINK the following dialog would appear.

File Services 3270 APPC LUA RJE Diagnostics Links Fl=Help

Name	Type	Description
[NEW]	SDLC	Add an SDLC Link

Link name [SDCLINK]

Defaults from SDCLINK

< OK > <Cancel>

ed

Copyright (C) 1993 CLEO Communications

3. Enter the name of the new link in the edit box. Link names must be one to eight characters long and can contain any alphanumeric character and the special characters \$, # and @. (Lowercase letters are converted to uppercase.) The link name cannot be the same as any other link name or invocable APPC TP name in your configuration, or the reserved names NODE or NETMA.N.
4. Choose < OK > (press ENTER).

The configuration dialog for that type of link is displayed. Configure the link as explained previously.

Deleting a Link

Note: You cannot delete a link if any connection is using exclusively this one link.

To delete an existing link:

1. Select the link you want to delete from the list box in the Configure Links dialog.
2. Choose < Delete >. A message box is displayed that prompts you to confirm the deletion of this link.
3. Choose < OK > to continue, or choose < Cancel > to cancel the operation.

Configuring Services

Renaming a Link

To rename an existing link:

1. Select the link you want to rename from the list box in the Configure Links dialog.
2. Choose < Rename >. A dialog is displayed that prompts you for the new name for the link.
3. Enter the new name for the link. Link names must be one to eight characters long and can contain any alphanumeric character and the special characters \$, # and @. (Lowercase letters are converted to uppercase.) This link name cannot be the same as any other link name, local node name or invocable APPC TP name in your configuration, or the reserved names BSDSLIM, LMUSLIM, NETMAN or UNIX.
4. Choose < OK > to continue, or choose< Cancel> to cancel the operation.

Connections: Adding, Deleting, Renaming

The default configuration file that we've provided defines a connection that should meet most of your startup requirements. We recommend that you initially use this file and make only those changes that are necessary to establish communication with the host system. After you've successfully established communication, then you can consider making additional changes to this supplied configuration or add another connection definition.

Adding a Connection

As with links, you add connections based on the type of communications adapter installed in your machine.

You can add connections using the system defaults; or you can use an existing connection similar to the one you are adding (this means the defaults for the prompts will be the same as the ones configured for the existing connection).

To add a new connection:

1. To add a new connection using the system defaults, select the [NEW] entry.
To add a new connection using an existing one as a model, select the existing connection.
2. Choose < Add >. The Add a Connection dialog is displayed. If you are using an existing connection as a model, its name is displayed in the dialog.

For example, if you were adding a [NEW] connection the following dialog would appear.

Configuring Services

Name	Type	Description
[NEW]	SDLC	

Connection name [.....]

< Add > < Edit > < Delete > < Rename > < Exit >

Copyright (C) 1993 CLEO Communications

3. Enter the name of the new connection in the edit box. Connection names must be one to eight characters long and can contain any alphanumeric character and the special characters \$, # and@. (Lowercase letters are converted to uppercase.) The connection name cannot be the same as any other connection in your configuration.
4. Choose < OK > (press ENTER).

The configuration dialog for that type of connection is displayed. Configure the connection as explained previously.

Deleting a Connection

Note: You cannot delete a connection if it has an LU (3270, LUA or RJE) or an APPC mode associated with it. An error message is displayed to inform you which LU or APPC mode uses the connection; before you can delete the connection, you must reconfigure the LU or APPC mode so that it does not use the connection.

To delete an existing connection.

1. Select the connection you want to delete from the list box in the Connections dialog.

2. Choose < Delete >. A message box prompts you to confirm the deletion.
3. Choose < OK > to continue, or choose < Cancel > to cancel the operation.

Renaming a Connection

To rename an existing connection:

1. Select the connection you want to rename from the list box in the Connections dialog.
2. Choose < Rename >. A dialog prompts you to enter the new name for the connection.
3. Enter the new name for the connection. Connection names must be one to eight characters long and can contain any alphanumeric character and the special characters \$, # and @. (Lowercase letters are converted to uppercase.) The connection cannot be the same as any other connection in this configuration.
4. Choose < OK > to continue, or choose < Cancel > to cancel the operation.

Local Nodes: Adding, Deleting, Renaming

The default configuration file that we've provided defines a local node that should meet most of your startup requirements. We recommend that you initially use this file and make only those changes that are necessary to establish communication with the host system. After you've successfully established communication, then you can consider making additional changes to this supplied configuration or add another local node definition.

Adding a Local Node

When adding a local node, you can use the system defaults as the basis for your local nodes; or you can use an existing local node similar to the one you want to configure (in this case the defaults will be the same as the existing local node).

To add a local node:

1. To add a new local node using the system defaults, select the [NEW] entry.

To add a new local node which is similar to an existing local node, select the existing local node from the list box.

2. Choose < Add >. The Add a Local Node dialog is displayed.

File Services 3270 APPC LUA RJE Diagnostics F1=Help

Name	Description
[NEW]	

Add a Local Node

Node name [SDLCNODE]

Defaults from SDLCNODE

< OK > <Cancel>

< Add > < Edit > < Delete > < Rename > < Parameters > < Exit >

Copyright (C) 1993 CLEO Communications

If you are using an existing local node as a model, the name of this local node is displayed in the dialog box; otherwise the “Defaults from” line does not appear.

3. Enter the name of the new local node in the edit box. Local node names must be one to eight characters long and can contain any alphanumeric character and the special characters \$, #, and @. (Lowercase letters are converted to uppercase.) The local node name cannot be the same as any other local node name, link name or invocable APPC TP name in your configuration, or the reserved names BSDSLIM, LMUSLIM, NETMAN or UNIX.
4. Choose < OK > or press **ENTER**.

The Configure Local Node dialog is displayed. See **Configuring Local Nodes**, earlier in this chapter, for further information.

Deleting a Local Node

To delete a local node:

1. Select the local node you want to delete from the list box in the Configure Local nodes dialog.
2. Choose < Delete >.

A conflation dialog is displayed asking if you want to delete the local node.

3. Choose < OK > to continue or < Cancel > to cancel the operation.

Note: If the local node being deleted has any LUs (3270, APPC, LUA or RJE) or LU pools associated with it, the configuration program displays a warning message that the LUs and/or pools will be deleted.

If the local node owns any connections, a confirmation dialog is displayed.

Configuring Services

File Services 3270 APPC LUA RJE Diagnostics Fl=Help
Local Nodes

Name	Description
Confirm Deletion	
The node being deleted owns some connections. Select <OK> to delete the node, the connections and the modes using them. Select <Cancel> to retain the node.	
< OK > <Cancel>	
< Add > < Edit > < Delete > < Rename > < Parameters > < Exit >	

Copyright (C) 1993 CLEO Communications

You can either delete all the connections on the node and any APPC modes which use them, or use the < Cancel > pushbutton to abandon deleting the local node. If you want to retain the connections while still deleting the node, return to the Configure Local Node dialog and remove all the connections from the node before deleting the local node.

If the local node owns the connection which is designated as the Network Management connection (see chapter 10, **Configuring Diagnostics**, for more information on configuring a network management connection), the following dialog is displayed.

File Services 3270 APPC LUA RJE Diagnostics Fl=Help
Local Nodes

Name	Description
Confirm Deletion	
The node being deleted owns the Network Management connection CONN1. Select <OK> to delete the node and the Network Management connection. Select <Cancel> to retain the node.	
< OK > <Cancel>	
< Add > < Edit > < Delete > < Rename > < Parameters > < Exit >	

Copyright (C) 1993 CLEO Communications

You have the following options:

- ◆ Select < OK > to delete the connection. In this case, you must configure another Network Management connection if you need to use LINKix's network management features.
- ◆ Select < Cancel > to abandon deletion of the local node.

Renaming a Local Node

To rename a local node:

1. Select the local node you want to rename from the list box in the Configure Local Nodes dialog.
2. Choose < Rename >.

A dialog is displayed prompting you to enter the new name for the local node.

3. Enter the new name for the local node. Local node names must be one to eight characters long and can contain any alphanumeric character and the special characters \$, #, and @. (Lowercase letters are converted to uppercase.) The local node name cannot be the same as any other local node name, link name or invocable TP name in your configuration, or the reserved names BSDSLIM, LMUSLIM, NETMAN or UNIX.
4. Choose < OK > to continue or < Cancel > to cancel the operation.

Where to Next?

When you've completed the process of configuring the link and connection, you're ready to configure for 3270 emulation, APPC, LUA or 3770 RJE.

- ◆ To configure for 3270 emulation, turn to chapter 6, **Configuring for 3270**.
- ◆ To configure for APPC, turn to chapter 7, **Configuring for APPC**.
- ◆ To configure for LUA turn to chapter 8, **Configuring for LUA**.
- ◆ To configure for 3770 RJE, turn to chapter 9, **Configuring for 3770 RJE**.

6. Configuring for 3270

This chapter contains instructions for configuring LINKix 3270 for 3270 emulation. The following topics are covered in this chapter:

- ◆ Overview of 3270 configuration
- ◆ How to configure 3270 LUs
- ◆ How to configure 3270 LU pools
- ◆ How to configure 3270 Users
- ◆ How to assign LUs to Users
- ◆ How to configure Response Time Monitor (RTM)
- ◆ How to configure NetView user alerts
- ◆ Modifying the 3270 style file

Overview of 3270 Configuration

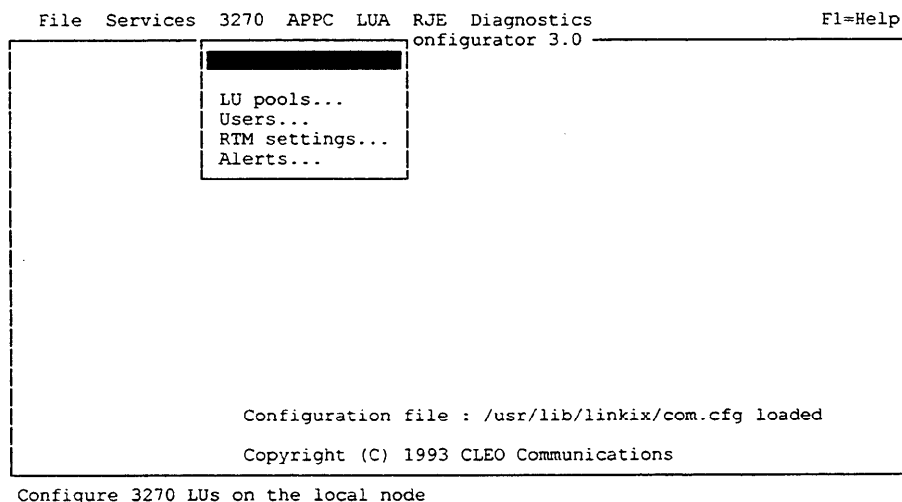
Configuring for 3270 emulation involves the following steps:

1. Before you configure for 3270 emulation you must have configured:
 - ◆ One or more local nodes
 - ◆ One or more links
 - ◆ One or more host connections that are owned by the local nodes and use the links
2. You configure the 3270 LUs (display and printer) that will be used during 3270 emulation.
3. Optionally, you configure 3270 LU pools for your system - LU pools are groups of 3270 display LUs.
4. You configure the 3270 Users – you can also setup a default configuration which can be used by any LINKix 3270 user. During this configuration process you assign previously configured 3270 LUs to specific users.
5. If you're using the LINKix 3270 remote SNA product, you may optionally configure the Response Time Monitor (RTM).
6. If you're using the LINKix 3270 remote SNA product, you may optionally configure NetView user alerts.
7. Finally, you may wish to optionally modify the 3270 style file which contains parameters that control the appearance and behavior of the UNIX terminal during 3270 emulation.

Configuring 3270 LUs

Before LINKix 3270's 3270 emulation program can be used, you must configure the system for operation in the 3270 environment. The first option you must configure is the 3270 LUs (display and printer) that will be used during 3270 emulation.

You configure 3270 LUs for your system using the LUs option of the 3270 menu:



When you select the LUs option, the 3270 LUs dialog is displayed.

Configuring for 3270

File Services 3270 APPC LUA RJE Diagnostics Fl=Help

3270 LUs

Local nodes	Configured LUs
	LU name Description
SDLCNODE ^	[NEW] ^
#	CRTL Display LU #
#	PRTL Printer LU #
#	#
#	#
#	#
#	#
#	#
v	v

< Add > < Add range > < Edit > < Delete > < Rename > < Exit >

Copyright (C) 1993 CLEO Communications

This dialog allows you to add one or more 3270 LUs, as well as view, edit, delete and rename existing LUs. It contains two list boxes. The left-hand list box shows all the configured local nodes. When you select a local node in this list box, the right-hand list box shows all the 3270 LUs currently configured for this local node.

Adding 3270 LUs

LINKix 3270 provides the ability to add one or more LUs to the configuration. These LUs maybe added individually, or several maybe added as a range.

Note: Before an LU can be used, it must be assigned to a 3270 user session. See the section on Configuring 3270 Users for complete details.

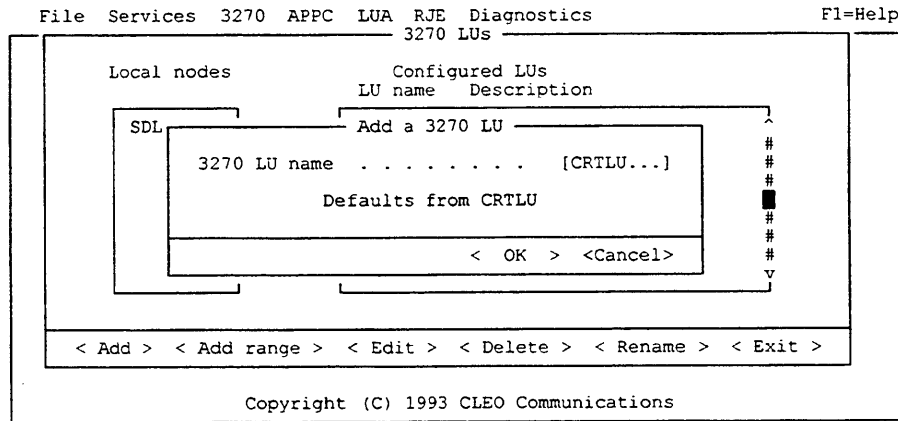
Adding a Single LU

To add a single 3270 LU:

1. From the left-hand list box, select the local node on which you want to configure the LU.
2. To add a new 3270 LU using the system defaults, select the [NEW] entry from the right-hand list box.

To add a new 3270 LU using the existing LU as a model, select the existing LU from the right-hand list box.

3. Choose < Add >. The Add a 3270 LU dialog is displayed.



If you used an existing LU as a model, the dialog contains the name of the existing LU.

4. Type the name you want to give the new LU; this can be up to eight characters (alphanumeric characters or the special characters #, @ and \$). You must enter a unique name for each LU – the name cannot be the same as any existing name for a 3270 LU, 3270 pool, APPC LU, LUA LU, LUA pool or RJE LU in this configuration.
5. Choose < OK >. The Configure 3270 LU dialog is displayed.
6. Complete the fields with the needed information. See Configuring a 3270 LU (p. 6-11) for details on the fields.

Base LU name

Specify a name of one to six characters as the base name for the LUs. All LUs will have this name, with a numeric or alphanumeric extension of one to three characters (one or two characters if the base name is six characters long). For example, you could specify NEWLUS as the base name and create LUs with names NEWLUS1, NEWLUS2 and NEWLUS3.

Extensions specified in

Specify whether you want the numeric extensions for the LU names to be in hexadecimal (0-9 and A-F), decimal (0-9) or alphanumeric (0-9 and A-Z) format. The default is hexadecimal.

First extension

Specify the extension you want for the first LU of the range. This can be up to three digits if the base name is one to five characters long, or up to two digits if it is six characters long (because the LU names must not be more than eight characters long in total). Subsequent LUs in the range will be given consecutive numeric extensions; for example, if you specify a base name of NEWLUS, hexadecimal extensions and first extension 9, the LUs will be named NEWLUS9, NEWLUSA, NEWLUSB and so on.

First LU number

Specify the LU number you want for the first LU of the range. Subsequent LIJs in the range will be given consecutive LU numbers. You should check that no existing LUs that use the same connection have LU numbers in this range; LU numbers must be unique across all 3270 LUs, APPC LUs, LUA LUs and 3770 RJE LUs using the same connection.

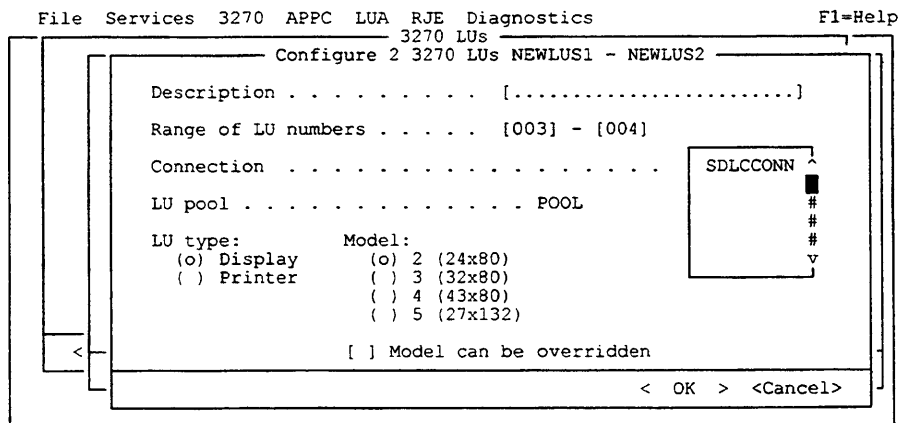
Check with your host administrator to determine the value you should enter here. Each LU number in the range must match the value set for the LOCADDR= parameter of an LU macro in the NCP Gen.

Number of LUs

Specify how many LUs you want to add. The default is 2.

4. After completing the fields, choose < OK > to configure the LUs. The Configure *n* 3270 LUs dialog is displayed.

Configuring for 3270



Make whatever changes are necessary to the following options. (Note that the options in this dialog apply to all the LUs in the range. However, you can edit individual LU configurations later if you need to. **See Editing a 3270 LU Configuration** for details.))

Description

The Description edit box contains a default description for the LUs. This description is optional and simply appears in the Configure 3270 LUs dialog.

Range of LU numbers

The numbers of the LUs, as specified in the previous dialog, are shown as a range and cannot be changed here.

Connection

The Connection list box shows all available configured host connections owned by the local node. Select the connection name from the list box.

LU pool

If you added a range of LUs that use an existing LU from an LU pool as a model, the new LUs are automatically in the same pool and the pool name is shown here.

LU type

The LU type radio group specifies the type of LUs you are configuring: display or printer. Check with your host administrator to determine the LU type. This radio group is grayed if the LUs are in a pool – the LU must first be removed from the pool before you can change its type.

Model

This radio group contains the model types for a display LU. The difference between the models is the screen display size (in rows and columns). You cannot change the screen model of the LU without first removing it from the pool. This radio group is disabled (grayed) if the LU type is printer.

Model can be overridden

If this check box is selected (checked) the user has permission to override the model type from the 3270 Control Interface. This field is disabled (grayed) if the LU type is printer. This radio group is also disabled (grayed) if the LUs are in a pool – you cannot change the override permission of an LU without first removing it from the pool.

5. After completing the fields in this dialog, choose < OK > to complete the configuration of the LUs, or choose < Cancel > to return to the Add a Range of LUs dialog. If you choose < Cancel >, you can either change the options and choose < OK > to add a new range of LUs, or you can choose < Cancel > to abandon adding the new LUs.

Editing a 3270 LU Configuration

To modify the configuration for an existing 3270 LU:

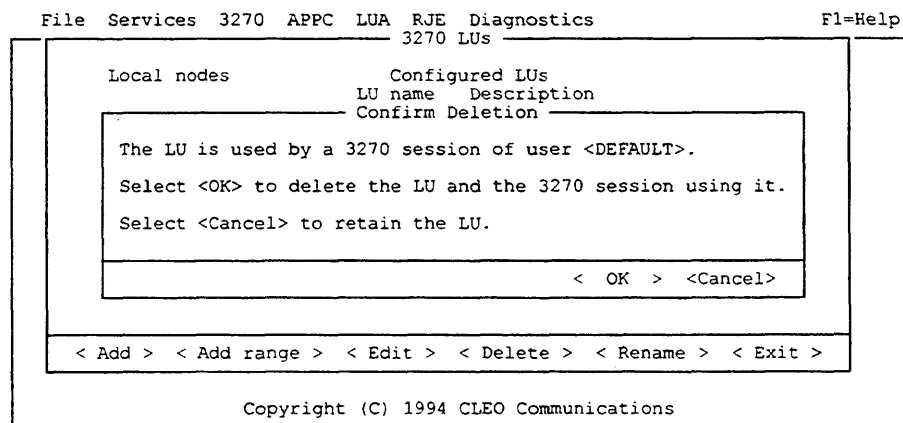
1. From the left-hand list box, select the local node on which the LU is configured.
2. Select the LU to modify from the right-hand list box in the 3270 LUs dialog.
3. Choose < Edit >. The Configure 3270 LU dialog is displayed, showing the existing configuration for the LU.
4. Make whatever changes are necessary. See **Configuring a 3270 LU** (p. 6-11) for details on the fields.

Deleting an LU

To delete a 3270 LU:

1. From the left-hand list box, select the local node on which the LU is configured.
2. Select the LU you wish to delete from the right-hand list box.
3. Choose < Delete >. Provided a 3270 session is not currently assigned to this LU, a message box appears that prompts you to confirm the deletion. Choose < OK > to delete the LU, or choose < Cancel > to abandon the deletion.

If one or more sessions are assigned to this LU, the following dialog appears:



4. Choose < OK > to delete the LU or choose < Cancel > to abandon the deletion.

Renaming an LU

To rename a 3270 LU:

1. From the left-hand list box, select the local node on which the LU is configured.
2. Select the LU you wish to rename from the right-hand list box.

3. Choose < Rename >. The Rename a 3270 LU dialog box is displayed, showing the existing name of the LU.
4. Type the new name you want to give the LU; this can be up to eight characters (alphanumeric characters or the special characters #, @ and \$). You must enter a unique name for the LU – the name cannot match the name of any other 3270 LU, 3270 pool, APPC LU, LUA LU, LUA pool or RJE LU specified in this configuration.
5. Choose < OK >.

Configuring a 3270 LU

When you add or modify a 3270 LU, the Configure 3270 LU dialog is displayed.

File Services 3270 APPC LUA RJE Diagnostics F1=Help

Configure 3270 LU CRTLU

Description [Display LU.....]

LU number [002]

Connection SDLCCONN

LU type: Model:

(o) Display (o) 2 (24x80)

() Printer () 3 (32x80)

() 4 (43x80)

() 5 (27x132)

[X] Model can be overridden

< OK > <Cancel>

Copyright (C) 1993 CLEO Communications

1. Make whatever changes are necessary to the following options. (Those with a * beside them are the ones that are most likely to require modification.)

Description

The Description edit box contains a default description for this LU. This description is optional and also appears in the previous dialog.

Configuring for 3270

* LU number

Each 3270 session (on a connection) has a unique number - this number identifies the LU in the SNA network and the device in BSC communications. This value is required and you must enter a unique value for each LU.

The number of LUs available on your connection depends on the host configuration. Check with your host administrator to determine the value you should enter here – the LU number must match the value set for the LOCADDR= parameter of the LU macro in the NCP Gen.

Connection

This list box shows all configured host connections owned by the local node. If the connection you want this LU to use isn't already selected, select the appropriate connection.

LU type

This radio group specifies the type of LU you are configuring: display or printer.

Model

This radio group contains the model types for a display LU. The difference between the models is the screen display size (in rows and columns). This radio group is disabled (grayed) if the LU type is printer.

Model can be overridden

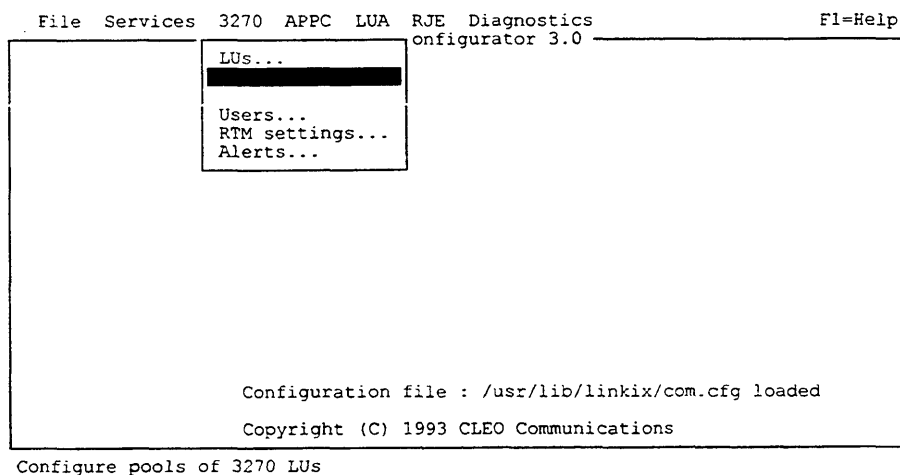
If this check box is selected (checked) the user has permission to override the model type from the 3270 Customize menu. This field is disabled (grayed) if you selected printer as the LU type.

2. When you've entered all necessary configuration information, choose < OK > to save the configuration and leave this dialog, or you can choose < Cancel > to abandon any changes before leaving the dialog.
3. You return to the 3270 LUs dialog where you can configure one or more additional LUs.
4. To exit from the 3270 LUs dialog, TAB to the < Exit > pushbutton and press ENTER. You are returned to the main configuration screen.

Configuring 3270 LU Pools

LU pools are groups of 3270 display LUs. 3270 sessions can be assigned to an LU pool instead of a specific LU; when a session is enabled, the first available LU from the pool will be used. You can configure several sessions to use the same LU pool; this means that you can provide a group of LUs to be used on a first come, first served basis. If there are more users than LUs available, user sessions will be assigned to LUs from the pool as required until there are no more free LUs. You can also create pools containing LUs on more than one connection and/or more than one local node; this means that sessions can still be established on an alternative host connection or local node if one connection fails. Note that an LU cannot be in more than one pool.

To begin configuring 3270 LU pools, select LU pools from the 3270 menu.



The 3270 LU pools dialog is displayed.

Configuring for 3270

File Services 3270 APPC LUA RJE Diagnostics F1=Help

3270 LU Pools

Name	Description
[NEW]	
COLPOOL	Pool of Color Terminals

< Add > < Edit > < Delete > < Rename > < Exit >

Copyright (C) 1993 CLEO Communications

This dialog contains a list box showing all the currently configured 3270 LU pools.

Adding a 3270 LU Pool

To add a new 3270 LU pool:

1. To add a new 3270 LU pool using the system defaults, select the [NEW] entry from the list box.

To add a new 3270 LU pool using an existing pool as a model, select the existing pool from the list box.
2. Choose < Add >. The Add a 3270 LU Pool dialog is displayed, showing the name of the existing LU pool.
3. Type the name you want to give the new LU pool; this can be up to eight characters (alphanumeric characters or the special characters #, @, and \$). You must enter a unique name for each LU pool – the name cannot match that of any 3270 LU, 3270 pool, LUA LU, LUA pool or RJE LU in this configuration.
4. Choose < OK >. The Configure 3270 LU Pool dialog is displayed. If you used an existing pool as a model, the defaults for fields in this dialog are taken from the existing pool, but the LUs in the existing pool are not shown because they cannot be in more than one pool.
5. Complete the screen fields as explained in **Configuring an LU Pool** (p. 6-16).

Editing an LU Pool Configuration

To modify the configuration for an existing LU pool:

1. Select the LU pool you want to modify from the list box in the 3270 LU Pools dialog.
2. Choose < Edit >. The Configure 3270 LU Pool dialog is displayed, showing the existing configuration for the LU pool.
5. Complete the screen fields as explained in the section **Configuring an LU Pool** (p. 6-16).

Deleting an LU Pool

To delete a 3270 LU pool:

1. Select the LU pool you wish to delete from the Configure 3270 LU Pools dialog.
2. Choose < Delete >. If no 3270 session is assigned to this LU pool, a message box prompts you to confirm the deletion. Choose < OK > to delete the LU pool, or choose < Cancel > to abandon the deletion.

If a 3270 session is assigned to this LU pool, the following confirmation dialog appears:

The screenshot shows a graphical user interface for configuring 3270 LU pools. At the top is a menu bar with options: File, Services, 3270, APPC, LUA, RJE, Diagnostics, and F1=Help. Below the menu bar is a window titled '3270 LU Pools'. Inside this window, there is a table with two columns: 'Name' and 'Description'. A modal dialog box titled 'Confirm Deletion' is open in the center. The dialog contains the following text: 'The pool is still used by 2 3270 sessions. Select <OK> to delete the pool and the 3270 sessions using it. Select <Cancel> to retain the pool.' At the bottom of the dialog are two buttons: '< OK >' and '<Cancel>'. Below the dialog, within the '3270 LU Pools' window, is a row of buttons: '< Add >', '< Edit >', '< Delete >', '< Rename >', and '< Exit >'. The bottom of the window displays the copyright notice: 'Copyright (C) 1993 CLEO Communications'.

Configuring for 3270

Choose the appropriate pushbutton. Choosing < Cancel> abandons the deletion of the LU, while choosing < OK > confirms the deletion (in which case the sessions will be left unused – not assigned to any LU or pool).

Renaming an LU Pool

To rename an LU pool:

1. Select the pool you wish to rename from the Configure 3270 LU Pools dialog.
2. Choose < Rename>. The Rename a 3270 LU Pool dialog is displayed, showing the existing name of the LU pool.
3. Type the new name for the LU pool; this can be up to eight characters (alpha-numeric characters or the special characters #, @ and \$). You must enter a unique name for each LU pool – the name cannot match the name of any other 3270 LU, 3270 pool, LUA LU, LUA pool or RJE LU in this confirmation.
4. Choose < OK >.

Configuring an LU Pool

When you add or modify an LU pool, the Configure 3270 LU Pool dialog is displayed:

File Services 3270 APPC LUA RJE Diagnostics Fl=Help

3270 LU Pools

Configure 3270 LU Pool NEWPOOL

Description [.....]
Pool size {0000}
Use count of pool {0000}

Model: [] Model can be overridden
(o) 2 (24x80)
() 3 (32x80)
() 4 (43x80)
() 5 (27x132)

LUs in this pool		Available LUs			
LU name	LU#	LU name	LU#	Mod	Ovr
		NEWLUS01	101	2	N
		NEWLUS02	102	2	N

< <- Move >

< OK > <Cancel>

1. Make whatever changes are necessary to the following options.

Description

The Description edit box contains a description for the LU pool. This description is optional and simply appears in the Configure 3270 LU Pools dialog.

Pool size

The Pool size field contains the number of LUs in this pool. This value is maintained automatically and cannot be changed.

Use count

This field shows the number of sessions assigned to this pool. This value is maintained automatically and cannot be changed.

Model

This radio group specifies the model type for the LUs in this pool. The difference between the models is the screen display size (in rows and columns). All LUs in the pool must have the same model type; you can move an LU into the pool, however, even if it does not have the same model type and its model will be changed to match the model specified for the pool.

Model can be overridden

If this check box is selected (checked), users have permission to override the model type from the Customize menu.

LUs in this pool/Available LUs

This dual list box shows all LUs in this pool and all other display LUs not in any pool. Move to the list box which contains the item you want to move. Select the item you want to move. The < Move > pushbutton is automatically highlighted and the arrow next to the word Move points in the direction in which the item will be moved. Press **ENTER**, or the accelerator key of the < Move > pushbutton, to move the item. It will disappear from one list box and appear in the other.

2. When you've entered all necessary configuration information, choose < OK > to save the configuration and leave this dialog, or you can choose < Cancel > to abandon any changes before leaving the dialog.

Configuring 3270 Users

LINKix 3270 allows you to configure each user of the UNIX system as a user of the 3270 emulation program, providing different sessions and user permissions for each user. This provides an additional level of security required at some installations.

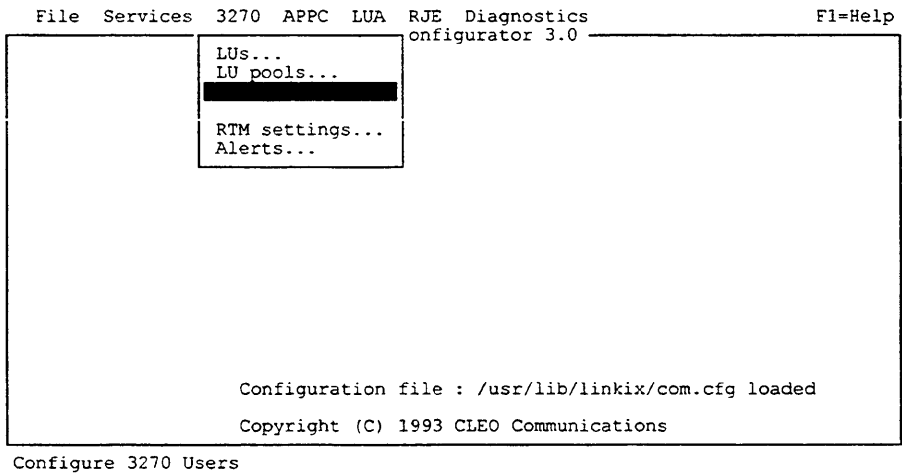
Users are identified by their UNIX login names, which normally have a maximum length of eight characters (the maximum length is defined in */usr/include/utmp.h*). If your UNIX system allows longer names than this, you should be aware of the following restrictions imposed by the LINKix 3270 configuration program:

- ◆ Names must be unique in the first eight characters (you should not have two user names of which the first eight characters are the same)
- ◆ Names longer than 20 characters will be truncated to 20 characters when displayed in the configuration program.

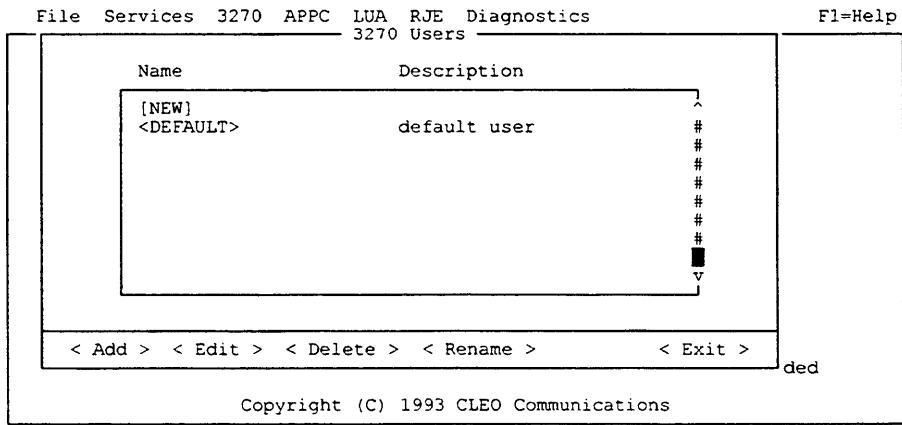
You can also set up a default configuration which can be used by any LINKix 3270 user; this default has the name `<DEFAULT>`. When a user starts the 3270 emulation program, LINKix 3270 checks the configuration file for a 3270 user with the login name used by the user. If this name is not found, LINKix 3270 then checks for the record named `<DEFAULT>` and uses this. Thus, if `<DEFAULT>` is defined, any user who can run *com3270* may access the host via the sessions assigned to the `<DEFAULT>` user.

When you create a new configuration file, LINKix 3270 includes a default 3270 user record with the name `<DEFAULT>`. This provides the default 3270 user record. If you do not want to provide a default, you can delete `<DEFAULT>` from the configuration file; in this case, you must explicitly configure each user who will be using the 3270 emulation program.

You configure the 3270 users by selecting the Users option of the 3270 menu.



The 3270 Users dialog is displayed.



Configuring for 3270

This dialog has a list box that contains all of the configured LINKix 3270 users and the default record <DEFAULT>. The <DEFAULT> user record is the one that will be used for all users who do not have a specific user configuration. This dialog allows you to add one or more users, as well as view, edit, delete and rename existing user configurations.

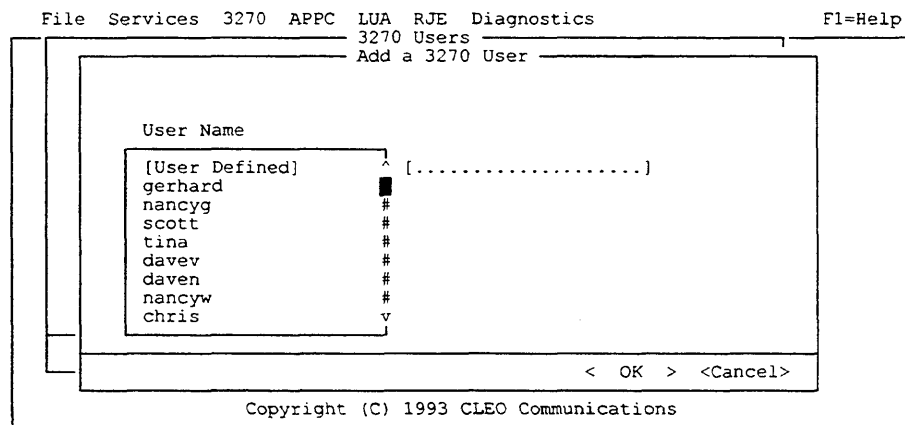
Adding a User

To add a new 3270 user:

1. To add a new 3270 user using the system defaults, select the [NEW] entry from the list box.

To add a new 3270 user using an existing user as a model, select the existing user from the list box.

2. Choose < Add >. The Add a 3270 User dialog is displayed. This dialog contains a list box with all valid login names on the LINKix computer on which you are running the configuration program. If you're using an existing user as a model, the name of this user is shown.



3. Use the cursor keys to highlight the new user in the list box and press **ENTER**. Alternatively, select [User Defined] from the list box and type the user name into the edit box (use this technique if the user name is a valid UNIX login ID on a different computer in a client/server environment).
4. Choose < OK >. The Configure 3270 User dialog is displayed.
5. Complete the screen fields as explained in Configuring a 3270 User (p. 6-22).

Editing a 3270 User Configuration

To modify the configuration for an existing user:

1. Select the user you want to modify from the list box in the 3270 Users dialog.
2. Choose < Edit > (press **ENTER**). The Configure 3270 User dialog is displayed.
3. Make whatever changes are necessary to the user configuration. See **Configuring a 3270 User** (p. 6-22) for details.

Deleting a User

To delete an existing 3270 user:

1. In the 3270 Users dialog list box, highlight the user you wish to delete.
2. Choose < Delete >. A message box appears prompting you to confirm the deletion.
3. Choose < OK > to delete the user, or choose < Cancel > to abandon the deletion.

Renaming a User

To rename a 3270 user

1. In the 3270 Users dialog list box, highlight the user you wish to rename.
2. Choose < Rename >. The Rename a 3270 User dialog is displayed. This dialog shows the current user name and contains a list box of valid login names on the computer on which the configuration program is running.

Configuring for 3270

3. Choose the new name of the user from the list box. Alternatively, select [User Defined] from the list box and type the required name into the edit box (use this technique if the user name is a valid UNIX login ID on a different computer in a client/server environment).
4. Choose < OK >.

Configuring a 3270 User

When you add or modify a 3270 user, the Configure 3270 User dialog is displayed.

File Services 3270 APPC LUA RJE Diagnostics Fl=Help
Configure 3270 User <DEFAULT>

Description [default user.....]
Max enabled sessions . . . (o) No limit () Limit of [...]

Session number	LU/pool name	Type
1	CRTL	D
2	PRTL	P
3	[UNUSED]	#
4	[UNUSED]	#
5	[UNUSED]	#
6	[UNUSED]	#
7	[UNUSED]	#
8	[UNUSED]	#
9	[UNUSED]	#
10	[UNUSED]	v

Permissions for user:
☐ Modify LU/session mapping
☐ Send user-input alerts
☐ View RTM data
☒ Modify initial style

Initial style
[NONE] [com3270.]
[User Defined] #
com3270 v

< Zoom > < Remap > < OK > <Cancel>

1. Make whatever changes are necessary to the following options.

Description

The Description edit box contains a default description for this user. This description is optional and provided solely to help you identify who this user is.

Max enabled sessions

This radio group allows you (the system administrator) to define the maximum number of simultaneously active (enabled) sessions this user may have. Thus, while a user may have ten sessions assigned, it is possible to limit the number of enabled sessions to less than ten.

Sessions

This list box shows the session numbers. Any sessions already assigned to an LU or LU pool are shown with the name of the LU or pool; sessions not assigned are shown as [UNUSED].

To assign a 3270 user session to an LU or LU pool, or to make a session unused, follow these steps:

- a. **TAB** to the Sessions list box and highlight the session you wish to change.
- b. Choose < Zoom > (press **ENTER**). The Configure Session dialog is displayed.

LU/pool	Type	LU#	Connection	Use count	Pool size
[UNUSED]					
CRTL	Display	002	SDLCCONN	0	-
PRTL	Printer	003	SDLCCONN	1	-

The list box shows all printer LUs, all LU pools and all display LUs which are not in a pool (and therefore available for use).

The Use count/Pool size column shows the number of sessions (for this user or any other user) already assigned to each LU or pool, and (for an LU pool only) the number of LUs in the pool. Note that the use counts do not include the session which you are currently configuring, but include all other sessions assigned to the LU or pool.

Configuring for 3270

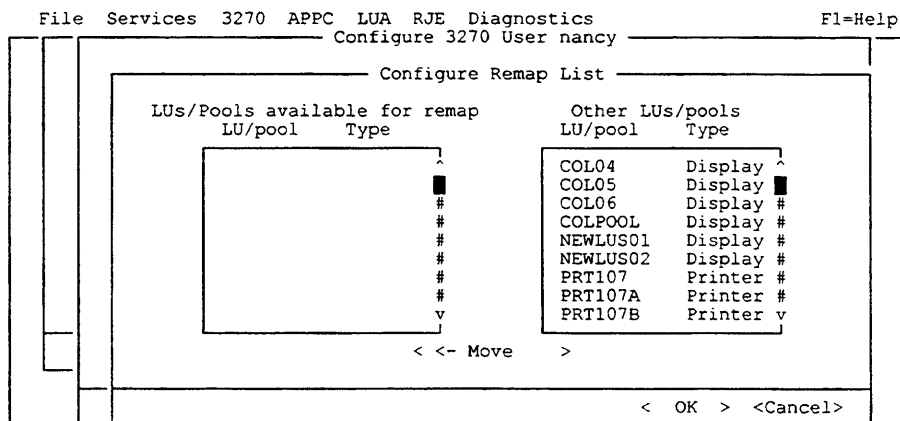
- c. Highlight the LU or pool you want this session to use, or highlight the [UNUSED] option if you want to leave the session unused.
- d. Choose < OK > to confirm the change, or choose < Cancel > to abandon.

You return to the Configure 3270 User dialog. Repeat steps a through d for each session to be added to the list box.

Modify LU/session mapping

This check box allows you to specify whether the user can change 3270 sessions to use a different LU or LU pool. If this option is enabled and selected, you must also specify which LUs and LU pools will be available for remapping. To do so, choose the < Remap > pushbutton – this pushbutton is disabled (grayed) unless you’ve given the user permission to remap sessions.

The Configure Remap List dialog is displayed:



The list box on the left shows all LUs and LU pools available for remapping by this user. The list box on the right shows all LUs and pools which are not in the user’s remap list and are not used by any of the user’s sessions. LUs or pools which are currently used by this user’s sessions are not shown, but are implicitly included in the remap list.

Move to the list box which contains the item you want to move. Select the item you want to move. The < Move > pushbutton is automatically highlighted and the arrow next to the word Move points in the direction in which the item will be moved. Press **ENTER**, or the accelerator key of the < Move > pushbutton, to move the item. It will disappear from one list box and appear in the other. Move LUs and pools which are to be available for remapping into the left-hand list box, and LUs and pools which are not to be available for remapping into the right-hand list box. Choose < OK > to confirm any changes, or < Cancel > to abandon the changes.

You return to the Configure 3270 User dialog.

Send user-input alerts

This check box specifies whether or not the user is allowed to send NetView user alerts to host NetView personnel. If you select this option, you must use the NetView Alerts option in the 3270 menu to specify the alerts. See **Configuring NetView User Alerts** for full details.

View RTM data

This check box specifies whether or not the user is allowed to view a display of host response times for 3270 sessions and whether or not the last transaction time indicator (LTTI) is displayed on the 3270 emulation status line. If you select this option, you need to configure the RTM settings, too (see **Configuring RTM settings** for full details).

Modify initial style

Specifies whether users are allowed to load new style files or to modify their 3270 customization. If this option is not selected, the user will only be able to use the default style file specified below in “Initial style” and will not be able to customize 3270 features such as session IDs and color mappings. This option is selected by default, allowing the user to modify the initial style.

Initial style

This edit box displays the default style file that will be applied if the user does not specify one when starting the 3270 emulation program. If the style file that we’ve provided (*com3270.stu*) has already been selected from the list box, its name appears in the edit box to the right. If you select the [User Defined] option, type the name of the style file (without the *.stu* extension) into the edit box. LINKix 3270 searches for this file first in the user’s home directory, then in */usr/lib/linkix/*. If you do not specify an initial style file and the user does not specify one at startup, the emulation program uses the built-in default settings.

Configuring for 3270

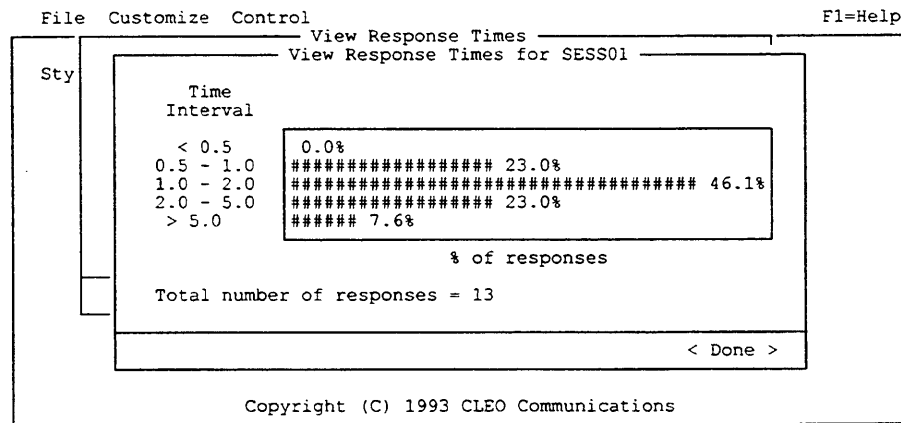
For details on creating style files, see the LINKix 3270 *3270 User's Guide*.

2. When you have finished configuring the 3270 user, choose < OK > to accept the changes, or choose < Cancel > to abandon the changes. You are returned to the 3270 Users dialog.
3. To exit from the 3270 Users dialog, choose < Exit >.

Configuring RTM Settings

This feature is available in LINKix 3270's remote SNA product only. If you're using any other package, RTM is not configurable.

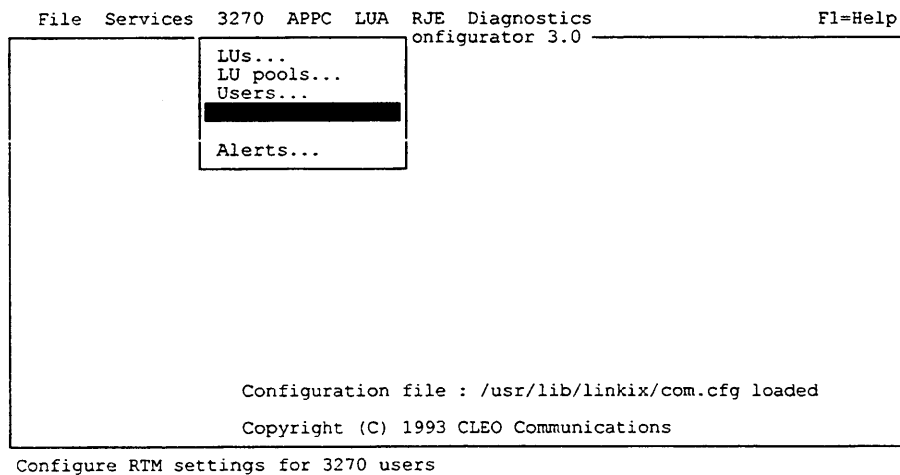
The Response Time Monitor (RTM) saves statistics on the amount of time it takes the host computer to respond to the local computer during 3270 display sessions. The 3270 user can view this information in the form of a bar chart (if permitted by the user configuration and the host).



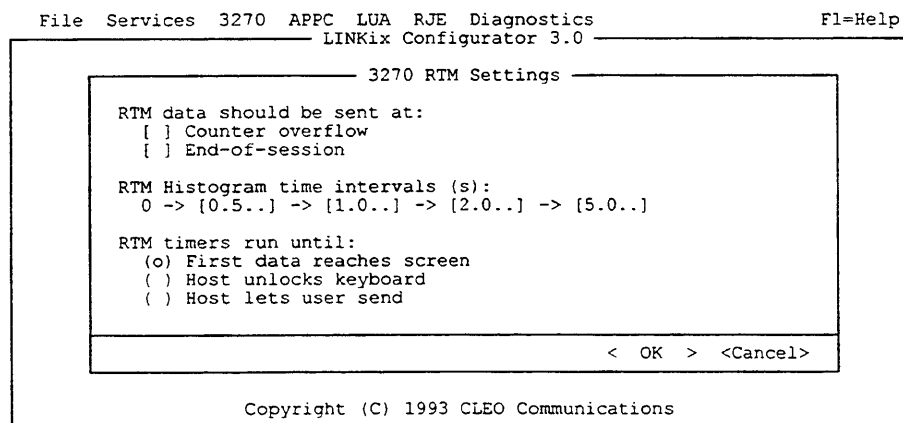
The host may choose to disable the display of RTM data, or to specify different measurement intervals or a different measurement definition from what you configure here. If this occurs, the host's settings will always override the values you configure.

Configuring for 3270

To configure the RTM settings, select RTM Settings from the 3270 menu.



The 3270 RTM Settings dialog is displayed.



1. Make the necessary changes to the following settings:

RTM data should be sent at:

RTM data can be requested by the host at any time. You can also specify that RTM data be sent at Counter Overflow and/or End-of-session. If you do not select an option, RTM data maybe lost.

Counter overflow If checked, RTM data is sent to the host when a response counter reaches its maximum value. Each time interval has a response counter that's incremented every time a response is received from the host.

End-of-session If checked, RTM data is sent to the host at the end of each communication session.

RTM Histogram time intervals

The RTM Histogram time intervals field sets the intervals under which the response times will be recorded. Intervals are measured in units of seconds. In the example dialog, response times will be recorded under one of five intervals: less than 1.0 seconds, from 1.0 to 2.0 seconds, from 2.0 to 3.0 seconds, from 3.0 to 4.0 seconds and greater than 4.0 seconds.

RTM timers run until:

This radio group defines when the host has responded.

First data reaches screen Specifies that response times are measured from the time data is sent from the local computer until the first of the returned data from the host reaches the screen.

Host unlocks keyboard Specifies that response times are measured from the time data is sent from the local computer until the host unlocks the keyboard.

Host lets user send Specifies that response times are measured from the time data is sent from the local computer until the host lets the local computer send more data.

2. When all necessary changes have been made, choose < OK > to save the changes or < Cancel > to abandon the changes.

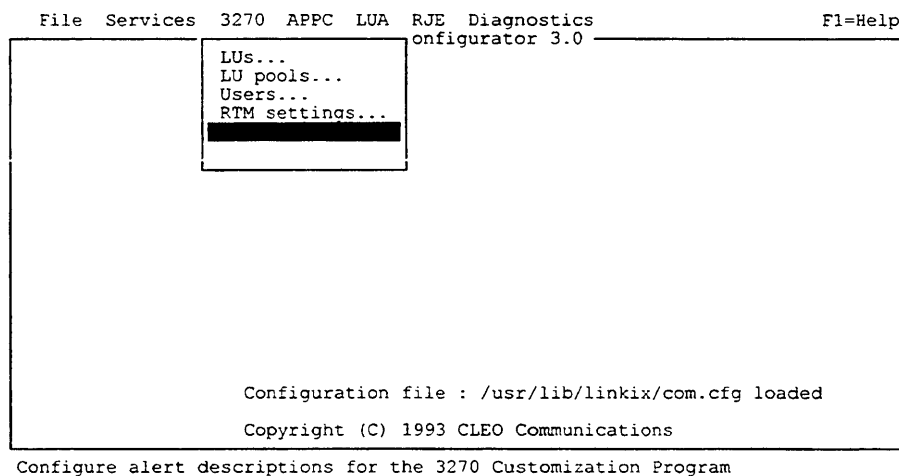
Configuring NetView User Alerts

This feature is available in LINKix 3270's remote SNA product only. If you're using any other package, NetView user alerts are not configurable.

You can configure up to 20 NetView user alerts for LINKix 3270. Alert descriptions and parameter prompts configured here appear in the 3270 user interface and tell the 3270 user what information should be provided for a particular alert.

When configuring alerts, you must work with your host administrator to decide upon the meaning of each alert number. NetView operators recognize alerts by number, since NetView alerts are sent in 3174 format which allows only numbers in the range of 1 to 20.

To configure NetView alerts, select Alerts from the 3270 menu.



The Configure 3270 Alert Descriptions dialog is displayed.

File Services 3270 APPC LUA RJE Diagnostics F1=Help
 Configure 3270 Alert Descriptions

No.	Alert description
01	Tape mount request for host 1
02	[UNUSED]
03	[UNUSED]
04	[UNUSED]
05	[UNUSED]
06	[UNUSED]
07	printer forms request for host 2
08	[UNUSED]
09	[UNUSED]
10	[UNUSED]

< Zoom > < Unuse > < OK > <Cancel>

Copyright (C) 1993 CLEO Communications

This dialog lists user alert numbers from 1 to 20, along with the alert description currently assigned to each one.

1. Select the appropriate alert number and choose < Zoom >. The Configure Descriptions for Alert dialog is displayed.

File Services 3270 APPC LUA RJE Diagnostics F1=Help
 Configure 3270 Alert Descriptions
 Configure Descriptions for Alert #06

Alert description [.....]

Parameter 1 description [.....]

Parameter 2 description [.....]

Parameter 3 description [.....]

< OK > <Cancel>

2. Make whatever changes are necessary to the following fields:

Configuring for 3270

Alert description

Each configured alert must have a description – a maximum 34 characters. This description should make sense to the user. NetView operators will not see this description, since they identify alerts by number, not descriptions.

This description should match the alert type specified by the host administrator.

Since NetView alerts are specific to each host, you may wish to include information in the alert description about which host the alert is intended for. If an alert is sent to an incorrect host, the user receives a popup message stating the host did not accept the alert.

Parameter (1,2,3) description

Enter any prompts you want displayed to the user in the three parameter description fields – maximum of 34 characters per parameter.

For example, if you're using an alert to ask the NetView operator to mount a special tape, you can specify a prompt for the tape name in one parameter description and one for the tape number in another parameter description. You could also define a general alert which the user can use to request special forms be loaded on the system printer, with a parameter that prompts for the form type.

You do not have to specify all of the prompts. Check with your NetView personnel the information required for each alert and specify a prompt for each required parameter; parameters for which you do not supply prompts will not be used.

Style File

In addition to the settings in the configuration file, the LINKix 3270 style file contains parameters that control the appearance and behavior of the UNIX terminal during 3270 emulation. During the configuration process, a default style file can be assigned to each user (we've provided one sample style file, *com3270.stu*, in the directory */usr/lib/linkix/samples/*).

Users can make changes to the style file during 3270 emulation, unless you've specified otherwise in the configuration. In that case, users will only be able to use the default style file that you've specified, and they will not be able to customize 3270 features. Unless you've changed the default setting, users are allowed to change the default style file.

For details on modifying the style file, see chapter 5 of the *3270 User's Guide*, **Customizing 3270 Emulation**.

7. Configuring for APPC and CPI-C

This chapter explains how to configure for Advanced program-to-program Communication (APPC) and CPI-C. It covers the following topics:

- ◆ An overview of APPC configuration
- ◆ An overview of CPI-C configuration
- ◆ Configuration requirements
- ◆ How to configure APPC modes
- ◆ How to configure remote APPC LUs
- ◆ How to configure local APPC LUs
- ◆ How to configure invocable TPs
- ◆ How to configure TP security
- ◆ How to configure CPI-C symbolic destination names

If you're unfamiliar with the concepts of APPC or CPI-C programming, we suggest that you read chapters 1 and 2 of the *APPC Programmer's Guide* or the *CPI-C Programmer's Guide*.

Overview of APPC Configuration

The following elements are involved in APPC communications:

- ◆ Modes
- ◆ LUs (logical units) — local and remote
- ◆ TPs (transaction programs)
- ◆ Security information (the TP specifies whether to use conversation-level security and, if so, provides a user ID/password combination)

Depending on the circumstances, not all of these elements maybe required for a particular APPC configuration. See **Configuration Requirements**, later in this chapter, for more information.

Mode Configuration

A mode specifies the networking characteristics used by a local LU to communicate with its partner LU. This includes information on the way data is transmitted between the two LUs and whether they can establish parallel sessions (more than one session at a time between the same two LUs). You can configure more than one mode for use by the same pair of LUs.

A mode can either be used for communications with a remote partner LU (in which case it also specifies which connection is used for communications with the partner LU), or for communications between two local LUs (in which case it is configured as a “local mode” and has no connection specified).

LU Configuration

APPC LUs are either local LUs or remote LUs. Local LUs are LUs on the local node; remote LUs are the LUs on remote nodes with which the local node communicates.

Each local LU can be configured to access one or more partner LUs. Sessions are established between a local LU and its partner.

A partner LU can either be a remote LU (on another computer) or another local LU (on the UNIX computer). If you want to use a remote LU as a partner, you must

configure information on the remote LU; if you want to use another local LU as a partner, you must configure both local LUs as being locally usable.

TP Configuration

TPs (transaction programs) are the application programs which use APPC to communicate with each other. An APPC conversation is established between two TP's when one TP requests APPC to start a conversation with its partner TP. The TP which starts the conversation is referred to as the invoking TP, and its partner is referred to as the invoked TP.

When an invoking TP starts, it identifies itself to APPC and specifies which local LU to use; it then specifies which remote LU and remote TP it wants to access and which mode to use. The mode specifies which connection to use and hence which computer in the SNA network will run the invoked TP. Invoking TP's do not need to be configured, but you need to configure the partner LUs and modes they will access; a TP may require multiple partner LUs and/or modes.

Invoked TP's on the local node (which can be invoked either by remote TP's or by other local TP's) must be configured with the information needed to start the TP. This includes information such as the file to execute, whether the TP is started automatically or manually by an operator, whether security information must be supplied when it is started and how long it waits to establish communications with the invoking TP.

If the invoked TP specifies that security information must be supplied when the TP is started, then you must also configure the user IDs and passwords which TP's on the UNIX computer can accept from invoking TP's. When the invoking TP invokes this TP, it must supply a user ID/password pair which matches one you have configured previously.

Implicit Modes

CLEO's implementation of APPC supports the use of **implicit modes**. This means that a session can be established between two LUs without specifying explicit mode names in the configuration.

If you configure an implicit mode for a pair of LUs, the invoking TP on the remote computer can specify any mode name using the [MC_]ALLOCATE verb; if it specifies a mode name which has not been configured for the local LU, the LU

Configuring for APPC

accepts the unknown mode name and uses the networking characteristics of the implicit mode.

To configure a mode as an implicit mode, enter an asterisk (*) as the first character of the mode name. A pair of LUs can have only one implicit mode configured, but can also have multiple named modes configured.

You can change a normal mode to an implicit mode or the other way around by changing the mode name. You cannot change a normal mode to an implicit mode if the change would cause a pair of LUs to have more than one implicit mode. You cannot change an implicit mode to a normal mode if a local LU is using the implicit mode with an implicit partner LU. (See the next section, **Implicit Partner LUs**, for further information on configuring implicit partner LUs.)

For further information about implicit modes, see **Configuring Implicit Modes** (p. 7-37) under **Configuring Local APPC LUs** later in this chapter. In addition, see the *APPC Programmer's Guide* for information on programming TPs to use explicit mode names.

Implicit Partner LUs

CLEO's APPC also supports the use of **implicit partner LUs**. This means that a session can be established between two LUs when the partner LU alias is not configured for the LU pair.

Implicit partner LU support provides added flexibility in configuring APPC operation. For example, by pairing an implicit remote LU with a local LU, you can allow any remote LU to establish a session with the local LU. You can only configure one mode for use with an implicit partner LU — this mode must be an implicit mode.

For further information about implicit partner LUs, see **Configuring Implicit Remote LUs** in the section **Configuring Local APPC LUs** later in this chapter. In addition, see the *APPC Programmer's Guide* for information on programming TPs to use LU names.

Host System Restrictions

Using APPC, you can communicate with either host computers or peer computers. When communicating with host computers running older versions of VTAM, you need to be aware of the following:

When communicating with a host running a VTAM system older than version 3 release 2, the use of dependent LUs only is supported. Dependent LUs are configured with a value of 1 or greater in the LOCADDR= parameter in the NCP Gen. Also, these older versions of VTAM do not allow parallel sessions; each LU can only be in one session at a time.

When communicating with a host running VTAM version 3 release 2 or newer with either NCP version 5 release 2 or newer (3720 and 3745 systems) or NCP version 4 release 3 or newer (3725 systems), the use of independent LUs is supported. Independent LUs are configured with a value of 0 in the LOCADDR= parameter in the NCP Gen. These newer versions of VTAM can also support multiple and parallel sessions per configured LU. When using independent LUs, a maximum of 254 sessions per LU are allowed.

To determine the VTAM version of the host system to which you are communicating, contact the host system administrator.

Overview of CPI-C Configuration

Although the programming interfaces for APPC and CPI-C are different, the actual data transmitted between programs is the same. Because CPI-C uses the same local LUs, remote LUs and modes as APPC for its communications, the configuration requirements are similar. To configure for communications between two CPI-C applications, you configure local and remote APPC LUs and modes in exactly the same way as you would for APPC communications. You configure an invocable CPI-C application and security information, if necessary, in the same way as an invocable APPC TP.

CPI-C configuration also includes a feature not used by APPC — symbolic destination names. See **Configuring CPI-C Symbolic Destination Names**, later in this chapter, for more information. If you are configuring only for APPC and not for CPI-C, you can ignore the Sym Dest Name menu option.

Configuration Requirements

Configuration requirements vary depending upon whether both TPs are on the same LAN or one TP is on the UNIX computer and its partner is on a remote node.

Two TPs on the Local Computer

To enable a pair of TPs on the same local computer so that they will communicate with each other, you must configure:

- ◆ A local mode; that is, one for which the connection is specified as [LOCAL].
- ◆ Two locally usable local APPC LUs. As a part of the local LU configuration process, you associate one local LU with another local LU and then associate the LU pair with a mode. This configuration process must be done for each of the two local LUs.
- ◆ The invocable TP.
- ◆ A user ID/password combination if conversation-level security is to be used.

In addition, you must start the control daemon before starting a TP — this procedure is explained in chapter 11, **Managing LINKix Components**.

In order for the control daemon to start, both a link and a connection *must* be configured, too. The TPs do not have to be associated with the connection, nor does the connection need to be active; the link and connection merely have to be configured, otherwise the control daemon will not start and an error is returned.

TPs on Different Computers

If you want a TP residing on the local node (the local UNIX computer) to communicate with a TP on another node, both computers must be configured correctly.

For the local node, configure:

- ◆ A link over which communication takes place — described in chapter 5, **Configuring Services**.
- ◆ A connection that is owned by a LINKix local node and uses the link — described in chapter 5, **Configuring Services**.

Configuring for APPC

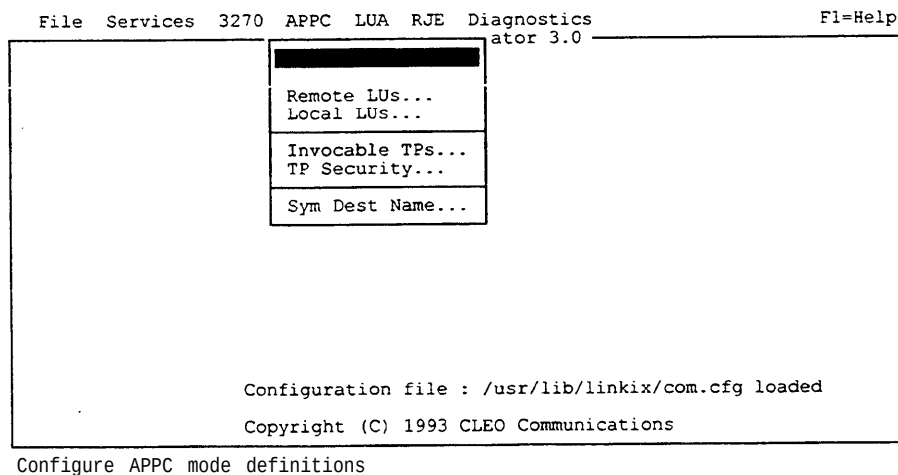
- ◆ A mode on the connection specifying the class of service and session parameters required for communication to take place.
- ◆ A remote APPC LU of the partner TP.
- ◆ A local APPC LU. As part of the local LU configuration process, you associate the local LU with the remote LU and associate a mode with the LU pair.
- ◆ The invocable TP, if it is on the local node.
- ◆ A user ID/password combination, if the invocable TP is on the local node and requires conversation-level security.

Configuring APPC Modes

A mode is a set of networking characteristics used by two APPC LUs to communicate. LU pairs use the parameters specified in the mode definition to determine if they can communicate with each other and, if so, how to pass data.

When you configure a mode, you establish a set of networking characteristics under a mode name. Later in the configuration process, you associate the modes you have configured with pairs of LUs.

You configure APPC modes by selecting the Modes option from the APPC menu.



The APPC Modes dialog is displayed.

Configuring for APPC

File Services 3270 APPC LUA RJE Diagnostics F1=Help
APPC Modes

Mode ID	Mode name	Description
[NEW]		
000	LOCMODE	For APPC Test Programs
		#
		#
		#
		#
		#
		#
		^
		v

< Add > < Edit > < Delete > < Exit >

ded

Copyright (C) 1993 CLEO Communications

Adding a Mode

You can add APPC modes using the system defaults or an existing mode as a model.

To add a mode, follow these steps:

1. To use the system defaults as a model for the new mode, select [NEW] from the list box.

To use an existing mode as the model for the new mode, select the existing mode from the list box.

2. Choose < Add >. The Configure APPC Mode with ID dialog is displayed. The mode ID, displayed as part of the dialog heading, is assigned by the configuration program and cannot be changed.

File Services 3270 APPC LUA RJE Diagnostics F1=Help
 APPC Modes
 Configure APPC Mode with ID 000

Description	[.....]
Mode name	[.....]
Connection	[LOCAL] ^
	SDLCCONN █
	#
	#
	v
[X] High priority mode	
Session limit	[001]
Minimum contention winner limit	[000]
Partner minimum contention winner limit	[000]
Automatic activation limit	[000]
Min send RU length . . . [0256]	Min receive RU length . . [0256]
Max send RU length . . . [1024]	Max receive RU length . . [1024]
Send pacing count . . . [04]	Receive pacing count . . [04]

< OK > <Cancel>

3. Configure mode parameters as explained in **Configuring a Mode** (p. 7-13).

Editing an APPC Mode Configuration

To modify an existing APPC mode configuration:

1. Select the mode to modify from the list box in the APPC Modes dialog.
2. Choose < Edit >. The Configure APPC Mode with ID dialog is displayed showing the existing configuration for the mode.
3. Make whatever changes are necessary to the mode parameters as explained in **Configuring a Mode** (p.7-13).

Deleting a Mode

To delete an APPC mode:

1. Select the mode to delete from the list box in the APPC Modes dialog.
2. Choose < Delete >. If the mode is not associated with any LU pairs, the Delete APPC Mode dialog is displayed; choose < OK > to confirm deleting the mode or < Cancel > to abandon the deletion.

Configuring for APPC

If the mode is associated with an LU pair, one of the following two Confirm Updates dialogs appears:

File Services 3270 APPC LUA RJE Diagnostics F1=Help
APPC Modes

Mode ID	Mode name	Description
[NE		Confirm Deletion
000		
001		LU pair L1A-R1A still uses the mode.
002		Select <OK> to remove all references to mode.
		Select <Cancel> to retain the mode.
		< OK > <Cancel>

< Add > < Edit > < Delete > < Exit >

ded

Copyright (C) 1993 CLEO Communications

This dialog indicates that the mode is associated with a pair of LUs, but is not the only mode configured for them. You can either abandon deletion of the mode or remove all references to this mode. If you remove references to the mode, the pair of LUs will still have at least one mode to use.

File Services 3270 APPC LUA RJE Diagnostics F1=Help
APPC Modes

Mode ID	Mode name	Description
[		Confirm Deletion
0		
0		LU L1A will have no modes to use with R1A.
0		Select <OK> to leave the LUs with no modes to use.
		Select <Cancel> to retain the mode.
		< OK > <Cancel>

< Add > < Edit > < Delete > < Exit >

ded

Copyright (C) 1993 CLEO Communications

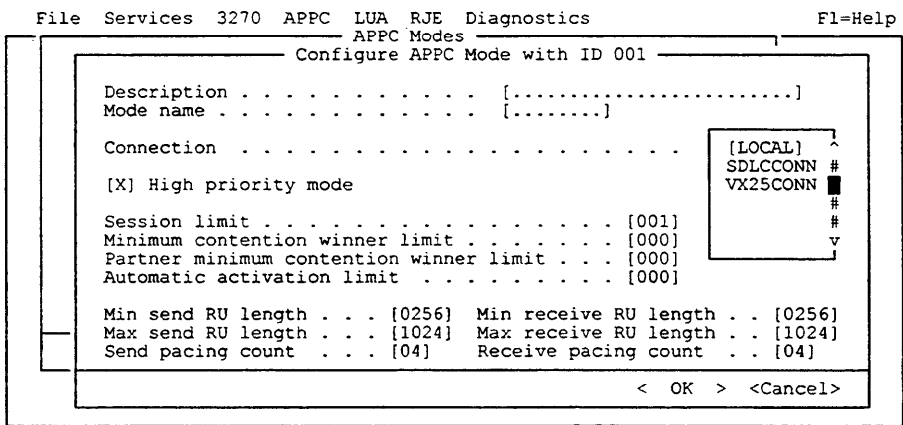
This dialog indicates that the mode is the only mode associated with a pair of LUs. You can either abandon deletion of the mode, or confirm deletion of the mode and leave the LUs with no modes to use. If you confirm deletion of the mode, you need to configure another mode for the LUs before they can be used.

Renaming a Mode

To rename an APPC mode, follow the instructions below for editing an APPC mode configuration and change the mode name.

Configuring a Mode

When you add or modify an APPC mode configuration, the Configure APPC Mode dialog is displayed.



Make whatever changes are necessary to the following fields.

Description

The Description edit box contains a default description for the APPC mode. This description is optional and may be up to 25 characters long. This description appears on the APPC Modes display screen and allows you to distinguish among your configured modes.

Configuring for APPC

Mode name

The Mode Name edit box contains the mode name. The mode is distinguished by the ID assigned by the configuration program, so the mode name does not have to be unique and is optional. However, all mode names associated with a specific pair of LUs must be unique. This name may be up to eight characters from the ranges A-Z and 0-9, and the special characters \$, # and @. (Lowercase characters are converted to uppercase.) You can enter a mode name consisting of all blanks; this appears as [blank] in the APPC Modes dialog. To indicate an implicit mode, begin the mode name with an asterisk (*).

Connection

The Connection list box shows all configured connections which are owned by local nodes (it is possible to configure a connection which is not owned by a local node, but in this case you cannot assign an APPC mode to it). Select the connection to be used by this mode from the list box. If the mode is to be used by two LUs that are communicating locally (both LUs reside on the local node), select [LOCAL]. This field is ignored if you're configuring an implicit mode.

High priority mode

Modes are either high- or low-priority. During communication between TPs, the local node always services high-priority modes before checking low-priority modes. If you're using a high-speed link for TPs, configuring the mode as high-priority will give you better service.

Session limit

The session limit field specifies the maximum number of sessions that a pair of LUs can have using this mode. An APPC mode is said to be **single session** when the session limit is 1. The mode is said to be **parallel session** when the session limit is greater than 1. Parallel sessions are multiple concurrent sessions between the same pair of LUs. This value may range from 1 to 254, but it must be less than or equal to the parallel session limit of any associated local LUs (p. 7-33); the default is 1.

Single session APPC modes can be used by parallel session LUs (local LUs with a session limit greater than 1 or remote LUs with Supports Parallel Sessions specified). Parallel session modes cannot be used by an LU pair if one of the LUs is single session. If a single session mode is already used by a single session LU, you cannot change the mode to parallel session by increasing the session limit.

Minimum contention winner limit

This field contains the minimum number of sessions for which the local LU must be the contention winner. The value may range from 0 to 254. The default value is 0. This value is required. The sum of this number and the value entered for the Partner Minimum Contention Winner Limit cannot exceed the number you specified for the maximum number of parallel sessions for this mode.

Partner minimum contention winner limit

This field contains the minimum number of sessions for which the partner LU must be the contention winner. This value may range from 0 to 254. The default is 0. This value is required. The sum of this number and the value entered for the Minimum Contention Winner Limit cannot exceed the number you specified for the maximum number of parallel sessions for this mode.

Automatic activation limit

This field specifies how many contention winner sessions will be activated for the local LU when the connection for this mode is started. This value may range from 0 to the minimum contention winner limit. The default is 0. This value is required and should not exceed the minimum contention-winner limit a session cannot be auto-activated if it is not the contention winner.

Min send RU length

This field specifies the minimum value in bytes that will be accepted for the maximum Request Unit (RU) length when sending to the partner LU. For further information about RU lengths, see **Defining RU Lengths**. This value may range from 256 to 4096; the default is 256. This value is required.

Max send RU length

This field specifies the maximum value in bytes that will be accepted for the maximum RU length when sending to the partner LU. this value may range between the Min Send RU Length value and 4096; the default is 1024. This value is required.

Send pacing count

This field specifies the maximum number of frames the local LU can send before waiting for a response from the partner LU. This value may range from 0 to 63; the default is 4. This value is required. A value of 0 indicates no send pacing count. This means that an unlimited number of frames can be sent without receiving a response from the partner LU, which may result in buffer shortage problems. Note that the partner LU can negotiate this value and set it to a non-zero value.

Configuring for APPC

Min receive RU length

This field specifies the minimum value in bytes that will be accepted for the maximum RU length when receiving from the partner LU. This value may range from 256 to 4096; the default is 256. This value is required.

Max receive RU length

This field specifies the maximum value in bytes that will be accepted for the maximum RU length when receiving from the partner LU. This value may range from Between the Min Receive RU Length value and 4096; the default is 1024. This value is required.

Receive pacing count

This field specifies the maximum number of frames that can be received from the partner LU before the local LU must send a response. This value may range from 0 to 63; the default is 4. This value is required. A value of zero indicates no receive pacing count. This means that an unlimited number of frames can be received without responding to the partner LU, which may result in buffer shortage problems.

Session Limits and Contention Winner Sessions

When two LUs communicate using parallel sessions, each LU is generally designated as the contention winner for a certain number of the sessions. If a local LU is the contention winner for a session, it has priority over its partner LU when attempting to allocate a conversation on that session, and will therefore win the contention if both LUs attempt to allocate a conversation on the session at the same time. If the partner LU is the contention winner for a session, the local LU must request permission of its partner before allocating a conversation on the session, and will therefore lose the contention if both LUs attempt to allocate a conversation on the session at the same time.

The APPC mode configuration contains values for the maximum number of parallel sessions which can be used by a pair of LUs using this mode, and the minimum number of sessions for which each LU is the contention winner. Since the remote LU's mode configuration may have different values, these values are negotiated between the two LUs before any conversations are allocated; the values can only be altered downward, not upward.

The automatic activation limit must be set to a value less than or equal to the local LU's minimum contention winner limit, because a session cannot be automatically activated if it is not the contention winner.

Sessions which are not automatically activated at connection startup are activated when they are required for use by a conversation (they can also be activated manually using the Manage program).

Supporting Parallel Sessions

For a pair of APPC LUs to support parallel sessions, a mode named SNASVCMG is required. This mode is used to negotiate session limits and should not be used for communications between application programs. It is required even if parallel-session LUs use a single-session mode.

For parallel session LUs, the extra mode is automatically configured with the default values shown below. This default mode is not shown in the Other Modes list box on the Select Modes for Use with Partner LU dialog.

Parameter	Setting
Priority	high
Maximum active sessions	2
Minimum contention winner	1
Partner minimum contention winner	1
Auto-activation limit	0
Minimum RU Lengths	256
Maximum RU Lengths	256
Send Pacing Count	4

The default mode parameters will be suitable for most configurations. If you need to use different values (for example, if the remote system's configuration cannot support the default values), you can override the defaults by configuring a mode with the name SNASVCMG and specifying values other than those shown. When you define the LU pair, you must associate the SNASVCMG mode as well as a standard mode with the LU pair.

Defining RU Lengths

When a session is established, partner APPC LUs negotiate the maximum size of request units transferred. Four of the fields in the Configure APPC Mode dialog define the parameters for this negotiation. The four fields are as follows:

- ◆ Minimum Send RU Length
- ◆ Maximum Send RU Length

Configuring for APPC

- ◆ Minimum Receive RU Length
- ◆ Maximum Receive RU Length

For the two LUs to communicate, the minimum-to-maximum range of send RU lengths for one LU must overlap with the minimum-to-maximum range of receive RU lengths for the other LU. The actual maximum RU length is negotiated when the session is set up. For further information, see IBM's *System Network Architecture Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2*.

The following table shows the overlapping of minimum-to-maximum ranges. In this example, the maximum RU length will be negotiated to 1024 bytes.

Parameter	Minimum	Maximum
Local LU send RU length	256	1024
Partner LU receive RU length	512	2048

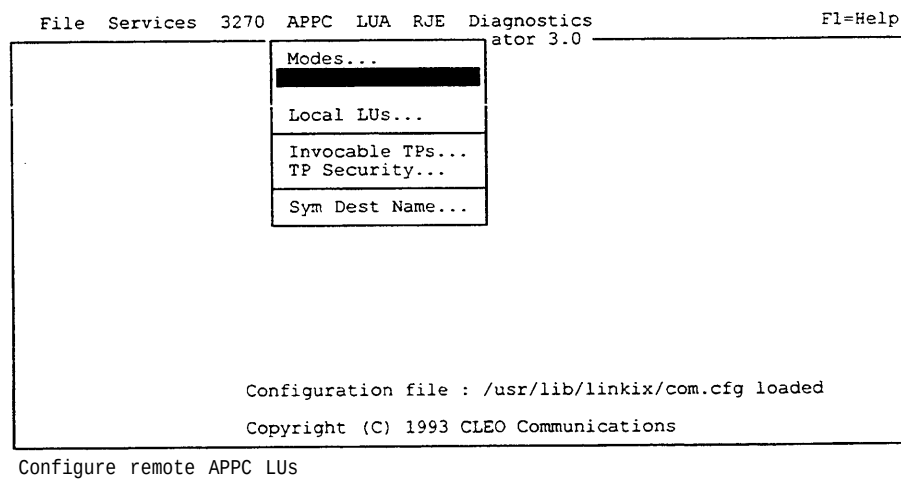
The local LU's receive RU length must overlap with the partner LU's send RU length in the same manner to provide a usable range of RU lengths the local LU can receive from the partner LU.

Configuring Remote APPC LUs

A remote LU resides on another SNA node. The remote LU can serve both invoking or invoked TPs.

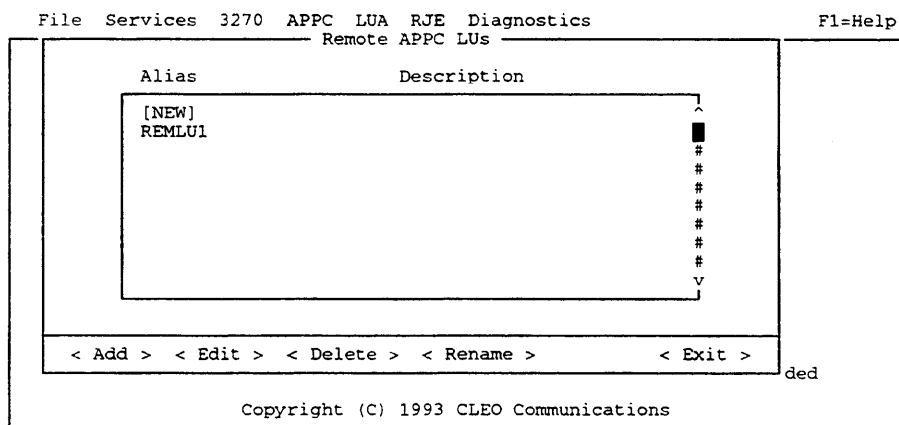
When you configure a remote LU, you provide information the local node needs to communicate with the remote LU. Later in the configuration process, you associate the remote LU with a local LU to create an LU pair.

To configure remote APPC LUs, select Remote LUs from the APPC menu.



The Remote APPC LUs dialog is displayed.

Configuring for APPC



This dialog lists all the remote LUs currently configured in your system. You can add, rename, delete or modify LUs from this dialog. These procedures are explained in the following sections.

Adding a Remote LU

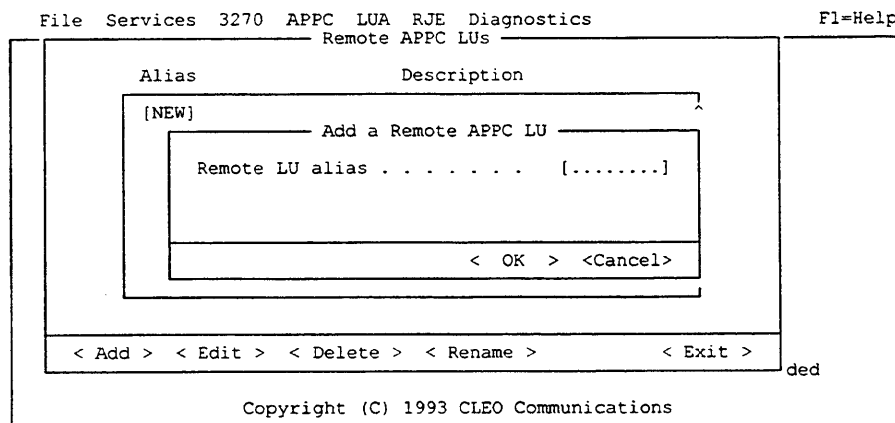
When you are adding a new remote LU, you can either use the system defaults as the basis for the LU or you can use an existing LU as a model.

To add a remote LU:

1. To use the system defaults as the model for the new remote LU, select [NEW] from the list box.

To use an existing LU as the model for the new remote LU, select the remote LU from the list box.

2. Choose < Add >. The Add a Remote APPC LU dialog is displayed.



3. Enter the alias of the remote LU. The alias can be one to eight uppercase characters (lowercase characters are converted to uppercase), the numerals 0 through 9, the special characters %, \$, # and @, and spaces between characters.

The remote LU alias must not match the alias of any other remote LU or any locally usable local LU. To define an implicit remote LU, begin the remote LU name with an asterisk (*).

4. Choose < OK >. The Configure Remote APPC LU dialog is displayed.
5. Configure the LU as explained in **Configuring a Remote LU** (p. 7-23).

Editing a Remote LU Configuration

To modify the configuration for an existing remote LU:

1. Select the remote LU you want to modify from the list box in the Remote APPC LUs dialog.
2. Choose < Edit >. The Configure Remote LU dialog is displayed showing the existing configuration for the remote LU.
3. Modify the configuration as explained in **Configuring a Remote LU** (p. 7-23).

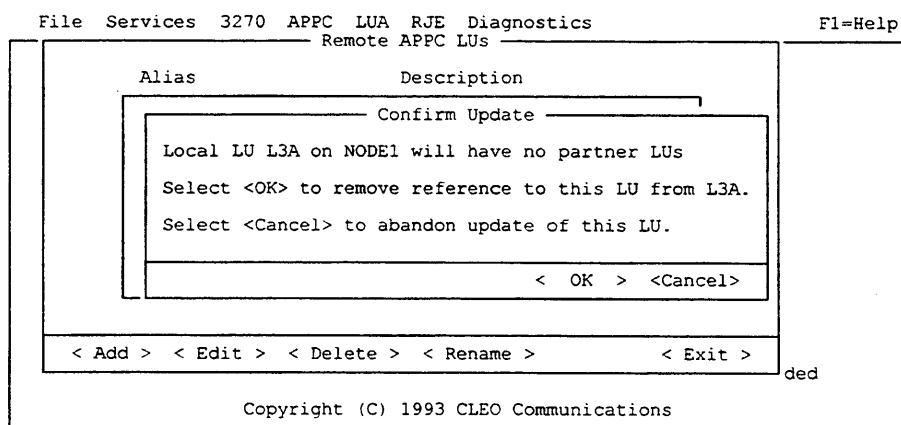
Deleting a Remote LU

To delete an existing remote LU:

1. Select the remote LU to delete in the Remote LUs list box.
2. Choose < Delete >.

If the remote LU is not the only partner LU associated with any local LU, the Delete Remote LU dialog is displayed; choose < OK > to confirm deleting the mode or < Cancel > to abandon the deletion.

If the remote LU is the only partner LU associated with a local LU, the Confirm Updates dialog appears:



You can either abandon deletion of the remote LU or confirm deletion of the remote LU and leave the local LU with no partner LUs to use. If you confirm deletion of the remote LU, you will need to configure another partner LU for the local LU before it can be used.

Renaming a Remote LU

To change the LU alias of a remote LU which has already been configured:

1. Select the remote LU from the Remote APPC LUs dialog.
2. Choose < Rename >. The Rename a Remote APPC LU dialog is displayed, showing the current LU alias.
3. Enter the new alias of the remote LU. The alias can be one to eight uppercase characters (lowercase characters are converted to uppercase), the numerals 0 through 9, the special characters %, \$, # and @, and spaces between characters.

The remote LU alias must not match the alias of any other remote LU or any locally usable local LU. To define an implicit remote LU, begin the remote LU name with an asterisk (*).

4. Choose < OK >.

Configuring a Remote LU

After naming the remote LU, you must specify the information needed by the system to configure the LU. You do this using the Configure Remote APPC LU dialog.

File Services 3270 APPC LUA RJE Diagnostics F1=Help

Remote APPC LUs

Configure Remote APPC LU with Alias REMLU2

Description [.....]

Full network name [.....].[REMLU2..]

Uninterpreted LU name [REMLU2..]

[] Supports parallel sessions

[] Use conversation-level security

[] Can pre-validate conversation security

Session-level security:

(o) No session-level security

() Security key in hex [.....]

() Security key in characters [.....]

< OK > <Cancel>

Copyright (C) 1993 CLEO Communications

Provide the necessary information in the following fields.

Configuring for APPC

Description

The Description edit box contains a description of the remote LU you're configuring. This description is optional and maybe up to 25 characters long. This description appears on the Remote APPC LUs display screen and allows you to distinguish between your configured remote LUs.

Full network name

This field contains the SNA network name of the remote LU you're configuring. This name is made up of two fields: the name of the network and the name of the remote LU. Each field in the network name has a maximum of eight characters from the ranges A-Z and 0-9, and the special characters \$, # and @. Lowercase letters are converted to uppercase.

This name is required. See your host administrator for the network name and remote LU name. This name must not match the full network name of any local LU that can access the remote LU.

If the LU alias contains a character that is not allowed in the full network name (for example, a blank space between characters), the portion of the LU alias that precedes the illegal character is used as the default LU name. For example, the LU alias "MY LU" would produce "MY" as the default LU name since the space is allowed only in the LU alias string, not in the full network name string.

Uninterpreted LU name

This field contains the alias of the remote LU as defined to the remote System Services Control Point (SSCP). This name may contain a maximum of eight alphanumeric characters and the special characters \$, #, @ and period (.); the default is the remote LU alias. (If the LU alias contains a character that is not valid in this field, the default is the portion of the LU alias up to the invalid character.)

This field is applicable only if you're using dependent local LUs to communicate with this remote LU; it's not applicable if the local LU is independent or if this is an implicit remote LU. A dependent LU communicates with VTAM; it needs a VTAM SSCP to activate it with an ACTLU command. The dependent LU must send the remote SSCP the name of the remote LU with which it wants to establish a session.

Supports parallel sessions

This check box specifies whether or not the remote LU supports parallel sessions (multiple concurrent sessions between two LUs). The default is unchecked (no support for parallel sessions).

If you change the remote LU from supporting single sessions to supporting parallel sessions, the system requires an SNASVCMG mode between the remote LU and any local LUs that use this LU. The system uses a default SNASVCMG mode unless you explicitly configure one. For information about configuring the SNASVCMG mode, see **Supporting Parallel Sessions** in the section **Configuring APPC Modes**, earlier in this chapter.

Use conversation-level security

This check box specifies whether or not TPs at this remote LU must use conversation-level security. The default is unchecked (no conversation-level security).

When conversation security is used, the invoking TP must supply a user ID/password combination that matches a user ID/password combination configured for the invoked TP.

Can pm-validate conversation security

This check box specifies whether or not the remote TP indicates that security has already been verified. For example, assume that TP A invokes TP B with a valid user ID and password and TP B in turn invokes TP C. TP B can send the LU for TP C the user ID from TP A and an already-verified indicator. This indicator tells TP C not to validate the password. The default is unchecked (cannot pre-validate conversation security). This option is valid only if you selected Use Conversation-Level Security.

Session-level security

Session-level security is transparent to the application programmer Writing a TP. It works by ensuring that the session is activated only if both ends have the same security key.

Select one of the following radio buttons as appropriate for your needs.

No Session-Level Security	Remote LU does not support session-level security.
Security Key in Hex	Remote LU supports session-level security; security key is in hexadecimal. Enter the key in the edit box. The security key must be exactly 16 hexadecimal characters long (lowercase characters are converted to uppercase).
Security Key in	Remote LU supports session-level security; security key is

Configuring for APPC

Characters in characters. Enter the key in the edit box. The security key must be exactly eight characters long.

Configuring Implicit Remote LUs

If you configure an implicit remote LU and pair it with a local LU, any remote LU can establish a session with the local LU. If the remote LU supplies an LU name not recognized by the local LU, the local LU accepts the unknown LU, applying to it the characteristics given in the BIND message. An implicit mode must be associated with the LU pair (local LU and implicit remote LU).

To setup an implicit remote LU, use an asterisk (*) as the first character of the LU alias. (The asterisk is valid in the first position only.) If the remote LU is implicit, only the Supports Parallel Sessions field is applicable.

You can change a normal remote LU to an implicit remote LU or the other way around by changing the LU alias. You cannot change a normal remote LU to an implicit LU if this would cause a local LU to have more than one implicit LU or if a local LU is using a normal mode with this remote LU.

If you change an implicit LU to a normal LU, the full network name and uninterpreted name for the remote LU are left blank; you may need to return to the Configure Remote APPC LU with Alias dialog to configure them.

For further information about implicit remote LUs, see **Configuring for Implicit Remote LUs** (p. 7-35) in the following section, **Configuring Local APPC LUs**.

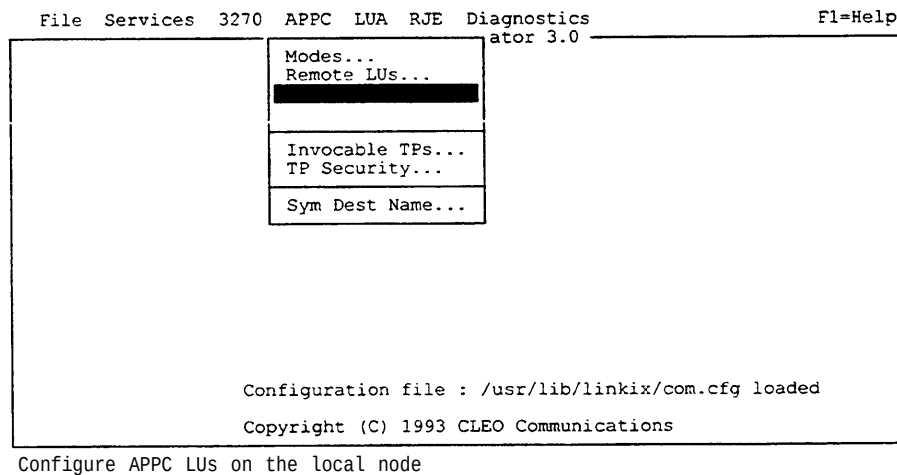
Configuring Local APPC LUs

A local LU resides on a LINKix local node. A local LU can serve either an invoking or invoked TP. Both applications in a conversation can reside on the local node. In this case you must configure a local LU for each of the two applications.

Configuring a local LU is a three-part process:

1. Define the characteristics of the local LU.
2. Associate the local LU with a partner LU to make an LU pair — you can associate the local LU with multiple partner LUs.
3. Associate a mode with each LU pair — you can associate multiple modes with an LU pair.

To configure local LUs, select Local LUs from the APPC menu.



The Local APPC LUs dialog is displayed.

Configuring for APPC

File Services 3270 APPC LUA RJE Diagnostics Fl=Help
Local APPC LUs

Local nodes	Configured LUs LU alias Description
[NEW]	[NEW]
	TPLU2 LU for Test Program 2

< Add > < Edit > < Delete > < Rename > < Exit >

Copyright (C) 1993 CLEO Communications

This dialog contains two list boxes. The left-hand list box shows all the configured local nodes. When you select a local node in this list box, the right-hand list box shows all the APPC local LUs currently configured for this local node and a [NEW] entry. From this dialog, you can add local APPC LUs, modify, rename or delete existing APPC LUs, and configure APPC LU parameters.

Adding a Local APPC LU

To add a new local APPC LU:

1. Select from the left-hand list box the local node on which you want to configure the LU.
2. To use the system defaults as a model for the new APPC local LU, select [NEW] from the right-hand list box.

To use an existing LU as the model for the new APPC local LU, select an existing local LU from the right-hand list box.

3. Choose < Add >. The Add a Local APPC LU dialog is displayed.

File Services 3270 APPC LUA RJE Diagnostics F1=Help
Local APPC LUs

LU alias	Description
[NEW]	

Add a Local APPC LU

APPC LU alias [.....]

< OK > <Cancel>

< Add > < Edit > < Delete > < Rename > < Exit >

Copyright (C) 1993 CLEO Communications

4. Enter the alias of the local LU. The alias can be one to eight uppercase characters (lowercase characters are converted to uppercase), the numerals 0 through 9, the special characters %, \$, # and @, and spaces between characters.

The alias cannot match any other local APPC LU alias. Furthermore, if this LU is to be locally usable, it cannot have the same alias as a remote LU.

5. Choose < OK > (press ENTER). The Configure Local APPC LU with Alias dialog is displayed.
6. Configure the LU as explained in **Configuring a Local LU** (p. 7-31).

Editing a Local LU Configuration

To modify the configuration for an existing local LU:

1. Select from the left-hand list box the local node on which the LU is configured.
2. Select the LU you want to modify from the right-hand list box.
3. Choose < Edit >. The Configure Local APPC LUs dialog is displayed showing the existing configuration for the local LU.
4. Modify the configuration as explained in **Configuring a Local LU** (p. 7-31).

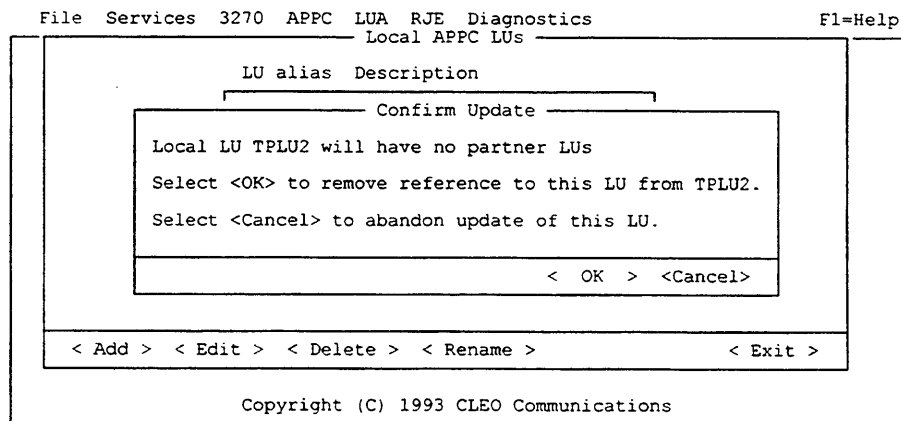
Configuring for APPC

Deleting a Local LU

To delete an existing local LU:

1. Select from the left-hand list box the local node on which the LU is configured.
2. Select the LU to delete from the right-hand list box.
3. Choose < Delete >. If the local LU is not the only partner LU associated with any locally usable local LU, the Delete Local LU dialog is displayed; choose < OK > to confirm deleting the LU or < Cancel > to abandon the deletion.

If the local LU is the only partner LU associated with a locally usable local LU, the Confirm Updates dialog appears:



You can either abandon deletion of this local LU or confirm deletion of the LU and leave the other local LU with no partner LUs to use. If you confirm deletion of this local LU, you will need to configure another partner LU for the other local LU before it can be used.

Renaming a Local LU

To change the LU alias of a local LU which has already been configured:

1. Select from the left-hand list box the local node on which the LU is configured.
2. Select the LU from the right-hand list box.
3. Choose < Rename >. The Rename a Local APPC LU dialog is displayed, showing the current LU alias.
4. Enter the new alias of the local LU. The alias can be one to eight uppercase characters (lowercase characters are converted to uppercase), the numerals 0 through 9, the special characters %, \$, # and @, and spaces between characters.

The alias cannot match any other local APPC LU alias. Furthermore, if this LU is to be locally usable, it cannot have the same alias as a remote LU.

5. Choose < OK >.

Configuring a Local LU

When you add or modify a Local LU, the Configure Local APPC LU with Alias dialog is displayed.

File Services 3270 APPC LUA RJE Diagnostics F1=Help

Local APPC LUs

Configure Local APPC LU with Alias NEWLLU

Description [.....]
 Full network name [.....].[NEWLLU..]
 LU number [...]
 Session limit [001]
☐ LU in the pool of default LUs ☐ Locally usable
☐ Conversation security
☐ Can pre-validate security

Other local LUs	Partner LUs	Other remote LUs
	Alias Type	
	TPLU2 L	

<-- Move > < Move >

< Modes > < OK > <Cancel>

Make whatever changes are necessary to the information in the following fields.

Configuring for APPC

Description

The Description edit box contains a description of the local LU you're configuring. This description is optional and maybe up to 25 characters long. This description appears in the Configure Local APPC LUs display screen and allows you to distinguish among your configured local LUs.

Full network name

This field contains the SNA network name of the local LU you're configuring. This name is made up of two fields: the name of the network and the LU name of the local LU. Each field in the network name has a maximum of eight characters from the ranges A-Z and 0-9, and the special characters \$, # and @. By default, this field contains the "Name of network" field contents (from the configuration of the local node that owns this LU) for the network name and the LU alias for the LU name.

If the LU alias contains a character that is not allowed in the full network name (for example, a blank space between characters), the portion of the LU alias that precedes the illegal character is used as the default LU name. For example, the LU alias "MY LU" would produce "MY" as the default LU name since the space is only allowed in the LU alias string, not in the full network name string.

See your host administrator for the network name and LU name. This name must not match the full network name of any partner LU for this local LU.

If the node communicates over separate connections to separate hosts (subareas), you may have to override the default network name for at least one of the LUs. The network name component of the fully qualified LU name must be the name of the subarea with which the LU is communicating. This not true for peer-to-peer communications, and the default values will normally be connect.

LU number

This field specifies the local network address of this local APPC LU. This address depends upon whether the local APPC LU is dependent or independent (see **Host System Restrictions**, earlier in this chapter). This value may range from 0 to 254.

This value is required. For an independent LU, specify 0 for the LU number. When using an independent LU, the LOCADDR parameter of the VTAM/NCP LU macro must be set to 0. For a dependent LU, specify 1 to 254 for the LU number. This value must match the LOCADDR parameter of the VTAM/NCP LU macro. (See your host administrator for the LU number.) The LU number must be unique for all APPC, 3270, LUA and RJE LUs that can access any of the connections used by this local APPC LU.

An independent LU cannot use a mode for a partner LU over a connection that exchanges format 0 XIDs — independent LU pairs require that format 3 XIDs be exchanged between the communicating nodes.

Session Limit

This field specifies the maximum number, of parallel sessions this local APPC LU can have in peer-to-peer communications. LUs supporting parallel sessions can use single session modes while LUs that support only single sessions cannot use parallel session modes. If either LU in an LU pair is single session, then parallel session modes are not allowed. This value may range from 1 to 254, but it must be greater than or equal to the session limit of any associated mode (p. 7- 14); the default is 1.

This value is required. If the local LU is a dependent LU (see **Host System Restrictions**, earlier in this chapter), this value must be 1. If the limit for a local LU is changed from a single session to parallel sessions, the system requires an SNASVCMG mode. You can let the system configure this mode by default or define your own SNASVCMG mode. For further information, see **Supporting Parallel Sessions** in the section **Configuring APPC Modes** earlier in this chapter.

If the LU number is specified, the LU must be dependent and the session limit is 1.

LU in the pool of default LUs

This check box whether or not the APPC LU you are configuring is a member of the default APPC local LU pool. When a TP is started, it can request that APPC assign a suitable local LU from the default LU pool by filling the LU alias field with hexadecimal zeros, instead of specifying an individual LU. The request is processed as follows:

- ◆ If an LU is available in the pool when the TP requests it, it is assigned to the TP.
- ◆ If a suitable LU is not available in the pool, the Allocate verb from the TP is rejected with an ALLOCATION_FAILURE return code.

Requirements for a suitable LU depend on the “return control” parameter specified by the TP:

Configuring for APPC

IMMEDIATE	The LU must have a free active contention-winner session of the specified mode with the specified partner LU.
WHEN SESSION AVAILABLE/FREE	The LU must support sessions of the specified mode with the specified partner LU.

For further information, see the *APPC Programmer's Guide*.

Locally Usable

This check box specifies whether or not other local LUs can access this LU and this LU can access other local LUs. If the local LU has the same alias as a remote LU, the local LU cannot be locally usable.

If you've made the LU locally usable and configured another local LU as a partner, this check box is checked and disabled (grayed). You must move the other local LU out of the Partner LUs list box before you can de-select this check box.

Conversation Security

This check box specifies whether or not TPs using this local LU as a partner are allowed to use conversation-level security.

This option only applies if the local LU can be used as a partner LU — that is, if it is configured as locally usable. It is disabled (grayed) if the local LU you're configuring is not locally usable.

Can Pre-validate Security

This check box specifies whether or not an intermediate TP (an invoked TP that in turn invokes another TP) can indicate that security has already been verified. For example, assume that TP A invokes TP B with a valid user ID and password and TP B in turn invokes TP C. TP B can send the LU for TP C the user ID from TP A and an already-verified indicator. This indicator tells TP C not to validate the password.

This option is enabled only if you selected Conversation Security.

Other local LUs, Partner LUs, Other remote LUs

From the Other Remote LUs and Other Local LUs list boxes, choose LUs that you want this local LU to be able to access and use the < Move > pushbuttons to move them into the Partner LUs list box. You can choose multiple partner LUs. Each LU

name in the Partner LUs list box is followed by the letter R or L to signify whether the LU is a remote or local LU.

If you select a partner LU from the Other Remote LUs list box, the resulting LU pair can be used for communication with a remote node.

The Other Local LUs list box is empty unless you configured this LU as locally usable, and it contains only the other local LUs which are configured as locally usable; that is, if you want to configure a pair of local LUs as partners, they must both be configured as locally usable.

If you select a partner LU from the Other Local LUs list box, you must also return to the configuration of the partner LU and configure this LU as a partner for it. For example, for the two locally usable LUs A and B to communicate, you must configure LU A and make LU B the partner LU; you must configure LU B and make LU A the partner LU. You must also configure modes for both the A-B LU pair, and the B-A LU pair.; the default is (none)

When you choose a local LU from one of the boxes, the left-hand < Move > pushbutton is enabled and can be used to move the LU between the Other Local LUs and Partner LUs list boxes. When you choose a remote LU from one of the boxes, the right-hand < Move> pushbutton is enabled and can be used to move it between the Other Remote LUs and Partner LUs list boxes. The arrow on the enabled pushbutton always points away from the selected LU toward the box into which the LU can be moved.

Configuring for Implicit Remote LUs

A local LU can have access to only one implicit remote LU. If the partner LU is an implicit remote LU, you can select only an implicit mode for the LU pair.

Associating a Mode with an LU Pair

For a pair of LUs to communicate, you must specify at least one mode. A mode is a set of networking characteristics that determine how data flows between two LUs.

To associate a mode with a pair of LUs:

1. If there's more than one partner for the local LU, select the partner LU in the Partner LUs list box. (If there's only one partner LU, you don't need to specify any selection, the local LU and the partner LU form the LU pair.)

- ◆ You cannot select a parallel-session mode if either of the LUs in the pair is a single-session LU.

Configuring Implicit Modes

All implicit modes are shown in the Other Modes box, regardless of whether the partner LU is local or remote.

You can configure only one implicit mode for an LU pair. If the partner LU is an implicit remote LU, the Other Modes list box contains only implicit modes, and only one can be selected.

Configuring Invocable TPs

Invocable TPs are APPC programs that can be invoked by other programs (local or remote). An invocable TP issues the RECEIVE_ALLOCATE verb.

You must configure invocable TPs that can be invoked on a LINKix server or client computer, providing the information needed to start the TP when it is invoked. If the invocable TP uses conversation-level security, you must also configure at least one user ID/password combination. (See **Configuring TP Security** at the end of this chapter for more information.)

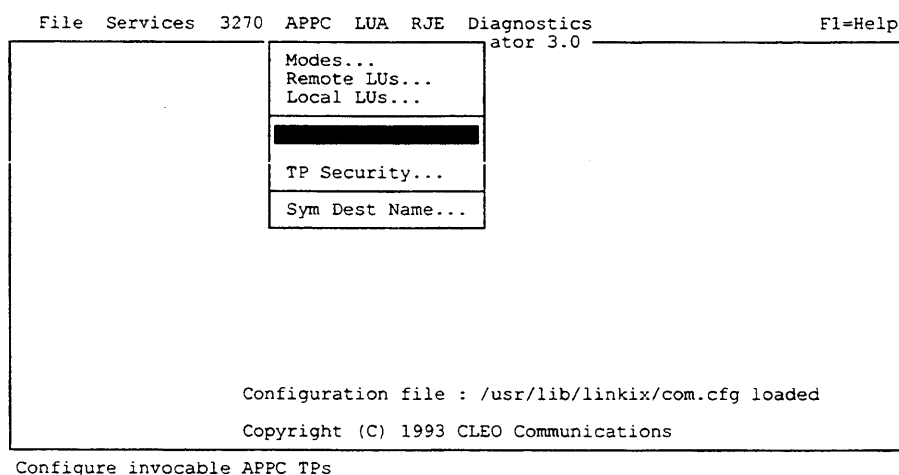
File Ownership of Invocable TPs

Be aware of the following points when configuring invocable TPs:

Invocable TPs are started from the home directory of the owner of the executable file for that TP. This means that the default path for trace files and other files accessed by the TP is the owner's home directory. Make sure the executable file has the correct ownership for LINKix to access the required files.

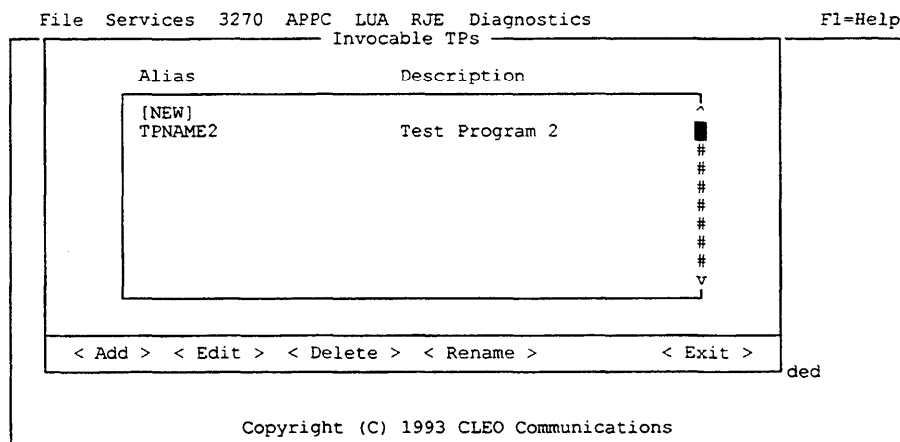
If the owner of the executable file is **root**, the file must have **setuid** permission to be started automatically by LINKix.

To configure invocable TPs, select Invocable TPs from the APPC menu.



Configure invocable APPC TPs

The Invocable TPs dialog is displayed.



This dialog lists all invocable TPs already configured. The alias and the description of the invocable TP are shown in the list box.

From this dialog, you can add, modify, delete or rename invocable TPs.

Adding an Invocable TP

To add an invocable TP:

1. Tousetie system default as the model for the new invocable TP, select [NEW] from the list box.

To use an existing invocable TP as the model for the new TP, select an existing invocable TP from the list box.

2. Choose < Add >. The Add an Invocable TP dialog is displayed.

Configuring for APPC

File Services 3270 APPC LUA RJE Diagnostics F1=Help

Invocable TPs

Alias	Description
[NEW] TP	

Add an Invocable TP

TP alias [.....]

< OK > <Cancel>

< Add > < Edit > < Delete > < Rename > < Exit >

ded

Copyright (C) 1993 CLEO Communications

3. Enter the alias of the invocable TP you are configuring. The alias can be one to eight uppercase characters (lowercase characters are converted to uppercase), the numerals 0 through 9, the special characters %, \$, # and @, and spaces between characters.

The alias cannot match the alias of any other invocable TP in the configuration. It also cannot match the name of any link or local node, or the reserved names BSDSLIM, LMUSLIM, NETMAN or UNIX.

4. Choose < OK >. The Configure Invocable TP dialog is displayed.
5. Configure the TP as explained in **Configuring an Invocable TP** (p. 7-41).

Modifying an Existing Invocable TP

To modify the configuration for existing invocable TPs:

1. Select the invocable TP you want to modify from the list box in the Invocable TPs dialog box.
2. Choose < Edit >. The Configure Invocable TP dialog is displayed showing the existing configuration for the invocable TP.
3. Modify the configuration as explained in **Configuring an Invocable TP** (p. 7-41).

Renaming an Existing Invocable TP

To rename invocable TP that have already been configured:

1. Select the invocable TP you want to rename from the list box in the Invocable TPs dialog.
2. Choose < Rename >. The Rename Invocable TP dialog is displayed, showing the existing alias of the invocable TP.
3. Enter the new alias of the invocable TP. The alias can be one to eight uppercase characters (lowercase characters are converted to uppercase), the numerals 0 through 9, the special characters %, \$, # and @, and spaces between characters.

The alias cannot match the alias of any other invocable TP in the configuration. It also cannot match the name of any link or local node, or the reserved names BSDSLIM, LMUSLIM, NETMAN or UNIX.

4. Choose < OK >. You return to the Invocable TPs dialog.

Deleting an Invocable TP

To delete an invocable TP:

1. Select the invocable TP you want to delete from the list box in the Invocable TPs dialog box.
2. Choose < Delete >. The Delete Invocable TP dialog is displayed, showing the alias of the invocable TP.
3. Choose < OK > to confirm the deletion, or choose < Cancel > to abandon the deletion. You return to the Invocable TPs dialog.

Configuring an Invocable TP

When you add or modify an Invocable TP, the Configure Invocable TP dialog is displayed.

Configuring for APPC

```
File  Services  3270  APPC  LUA  RJE  Diagnostics  F1=Help
                        Configure Invocable TP NEWTP

Description . . . . . [.....]

( ) Queued - operator started      [ ] Conversation security
(o) Queued - autostarted           [ ] Accept already-verified
( ) Non-queued - autostarted

TP type . . . . . (o) APPC      ( ) CPI-C
TP name in . . . . . ( ) Hex    (o) Characters
Full TP name . . . . . [.....]

Executable file . . . . . [.....]
Parameters . . . . . [.....]
Environment . . . . . [.....]
Target machine . . . . . [.....]

Timeout for loading TP . . . . . [0060] seconds
Timeout for servicing TP ( ) Infinite (o) Finite [0060] seconds

                        < OK > <Cancel>
```

Make whatever changes are necessary to the information in the following fields.

Description

The Description edit box contains a description of the invocable TP you are configuring. This description is optional and may be up to 25 characters long. This description appears in the Invocable TPs display screen and allows you to distinguish among configured TPs.

TP's loading method

Specify how the invocable TP will be loaded when invoked by another TP:

Queued operator started

The operator must load the program when prompted by the invoking TP in order for the conversation between the TPs to take place.

The operator might also start the invocable TP before the invoking TP, in which case the invocable TP waits for an incoming [MC_]ALLOCATE to arrive at the local LU.

Queued autostarted

The invocable TP is automatically loaded the first time an invoking TP tries to begin a conversation. Every subsequent attempt to invoke the TP is queued until the TP issues another RECEIVE_ALLOCATE, or until the TP ends and can be reloaded. This method ensures you

will have only one instance of the TP running at one time.

**Non-queued
autostarted**

The TP is automatically loaded every time an invoking TP requests that it be loaded. Using this method, you can have more than one instance of an invocable TP running at the same time.

Conversation security

This check box specifies whether or not the invoking TP must supply a user ID and password when it issues the [MC_] ALLOCATE verb.

See the **Configuring TP Security** section at the end of this chapter for details on configuring user IDs and passwords.

Accept already-verified

This check box specifies whether or not the invoked TP can accept an indication from the invoking TP that security has already been verified.

This option is not available unless conversation security is specified for the invocable TP using the Conversation Security check box in this dialog.

TP type

Specify whether the invocable TP uses APPC or CPI-C.

TP name in

APPC TP names can be specified in either hexadecimal or characters. Use the Hex and Characters radio buttons to specify whether the name of the invocable TP is in hexadecimal or character format.

Full TP name

The full name of the invocable TP, up to 64 characters. Valid characters are a-z, A-Z, 0-9, and. (period). For an APPC invocable TP, the following characters are also valid: # @ \$. (For a CPI-C invocable TP, the following characters are also valid< (+&*) ; - / , % _ > ? : ' = " .)

Executable file

The file name of the executable file for the invocable TP. This must include the full directory path of the file, up to a maximum of 64 ASCII characters.

Configuring for APPC

This field is required. For operator-started TPs, this information appears in the operator console message which prompts the operator to load the TP; for auto-started TPs, it indicates the executable file which will run.

Parameters

This field specifies any start-up parameters for the TP (these would be used on the command line if the TP were started from the command line). This optional field holds a maximum of 64 ASCII characters. Use spaces to separate parameters.

Environment

This field specifies any optional environment variables to be set before the invocable TP is loaded. Use the following format for the environment variable(s) string:

```
envvar1=value1 | envvar2=value2 . . .
```

The entire string can be up to 64 ASCII characters. Using this parameter sets the required environment variables before starting the program.

Target machine

This field specifies the machine name of the server or client on which the invocable TP is to be loaded. This field may contain up to 15 characters. If you leave this field blank, the TP will be started on the server on which the local node receiving the incoming Allocate is running.

Timeout for loading TP

This field specifies how long the invoking TP will wait, after issuing the Allocate verb, for the invocable TP to be loaded (for the invocable TP to be started and successfully issue the RECEIVE_ALLOCATE or Accept Conversation). After the specified time the invoking TP times out. This value may range from 1 second to 3600 seconds; the default is 60 seconds.

If the invocable TP is operator-started, this value should be longer than 60 seconds to give the operator time to execute the TP.

Timeout for servicing TP

This field specifies how long the invocable TP will, after issuing the RECEIVE_ALLOCATE or Accept Conversation, wait for an incoming Allocate for this TP name to arrive at the local LU. If the incoming Allocate does not arrive before the time specified by this field, the invocable TP times out. You can specify Infinite if you want the TP to wait indefinitely for an incoming Allocate, or Finite

(with a timeout value) if you want to set a timeout. This value may range from 0 second to 3600 seconds; the default is 60 seconds. A timeout value of zero means that the invocable TP will always time out unless an incoming Allocate has already been received by the time it issues RECEIVE.ALLOCATE or Accept Conversation. A non-zero value means that the TP will wait for the specified number of seconds after issuing RECEIVE_ALLOCATE or Accept Conversation before it times out.

Configuring TP Security

If you configure an invocable TP to use conversation-level security, the invoking TP must supply a user ID and password. This user ID/password combination must match a user ID/password combination you have configured using the Configure TP Security Information dialog.

If you configure an invocable TP to use pre-validated security, the user ID passed by the invoking TP must match a user ID you have configured using the Configure TP Security Information dialog.

Note that these user IDs and passwords are used solely by APPC (and CPI-C invocable) TPs; they have no connection with the login IDs and passwords on your UNIX computer, or with 3270 user IDs.

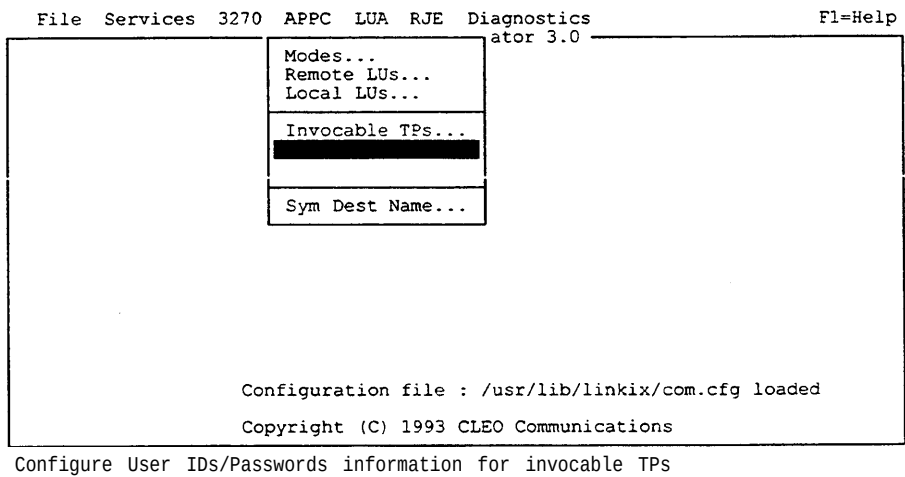
Note also that user IDs and passwords are not tied to specific invocable TPs. If a TP attempts to invoke another TP that requires conversation-level security, the invoking TP can provide any user ID and password that has been configured to be used with TP security. For example, assume the following have been configured:

Local TPs	Remote TPs	User IDs/Passwords
LOCALTP1	RMOTETP1	USER1/FRANK
LOCALTP2	RMOTETP2	USER2/BOB
LOCALTP3		USER3/UTIL
LOCALTP4		
LOCALTP5		

RMOTETP1 and RMOTETP2 use a remote LU and the five LOCALTPS use the local LU. Three user ID/password combinations are also defined on the local node. The LOCALTPS are configured to require conversation-level security. In this scenario, RMOTETP1 can specify any of the three user ID/password combinations to invoke any of the LOCALTPS.

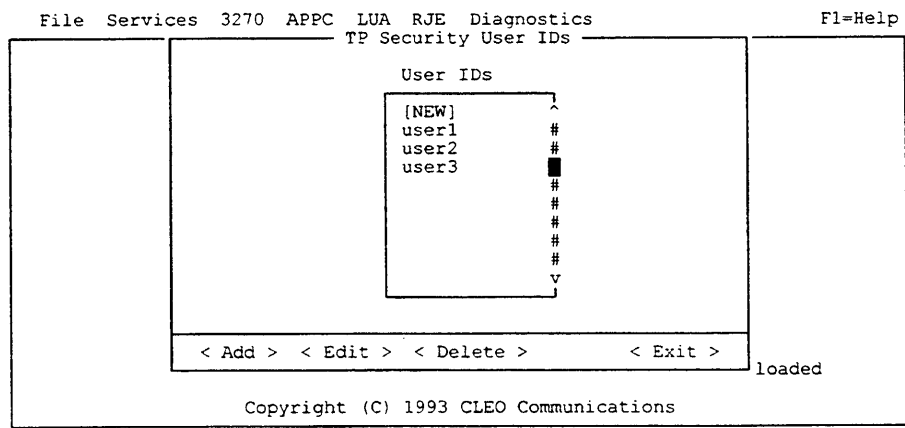
If RMOTETP1 invokes LOCALTP1, and LOCALTP1 later needs to invoke RMOTETP2 (which is configured to accept “already-verified security”), LOCALTP1 can successfully invoke RMOTETP2 by sending the user ID it received from RMOTETP1 with the “already-verified security ”indicator.

To configure security for TPs, select the TP Security option from the APPC menu.



Configure User IDs/Passwords information for invocable TPs

The Configure TP Security Information dialog is displayed.



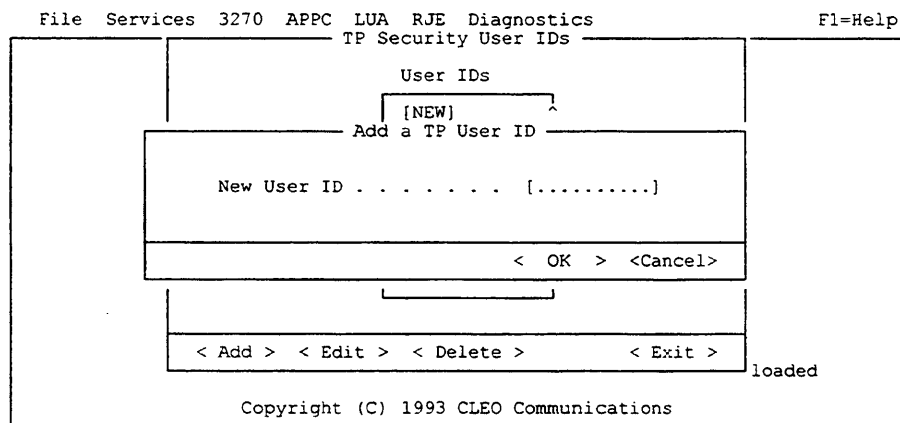
The User IDs list box displays all the configured user IDs. From this dialog, you can add anew user ID, get to a user’s password information or delete a user ID.

Adding a New User ID

To add a new user ID:

1. Select the [NEW] item in the User IDs list box and choose < Add >.

The Add a TP User ID dialog is displayed.

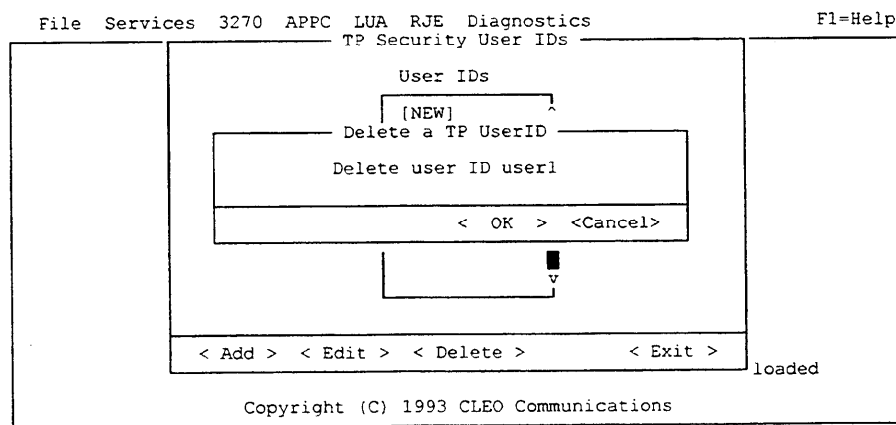


2. Enter the new user ID in the edit box and choose < OK >. The user ID cannot match any other user ID in the list box. It can be one to 10 characters long, and it can contain any alphanumeric character as well as the special characters \$, #, @ and period (.).

Deleting a User ID

To delete a user ID:

1. Select the user ID to be deleted from the User IDs list box and choose < Delete>. The Delete a TP User ID dialog is displayed.



2. Choose < OK > to delete the user ID. You can also choose < Cancel > to cancel the delete process.

Specifying a Password

To specify a new password for a user ID, or to change an existing password, perform the following steps:

1. Select the appropriate user ID from the User IDs list box, and choose < Edit >. The Configure TP Security Information for User dialog is displayed.

Configuring for APPC

```
File  Services  3270  APPC  LUA  RJE  Diagnostics  F1=Help
      TP Security User IDs
            User IDs
      Configure TP Security Information for User user1
      Password . . . . . [.....]
      Password check . . . . [.....]
                                < OK > < Cancel >
                                < Add > < Edit > < Delete > < Exit >
                                loaded
      Copyright (C) 1993 CLEO Communications
```

2. Type the password in the Password edit box.

Passwords can be one to 10 characters long, and can contain any alphanumeric character, as well as the special characters \$, #, @ and period (.). The password is not displayed on the screen as you type it but the cursor moves as you type each character.

3. As a check that you have entered the password correctly, type the same password in the Password Check edit box.

If the entries in the two edit boxes are not the same, an error message is displayed. Check to make sure the password is correct and enter it again in both edit boxes.

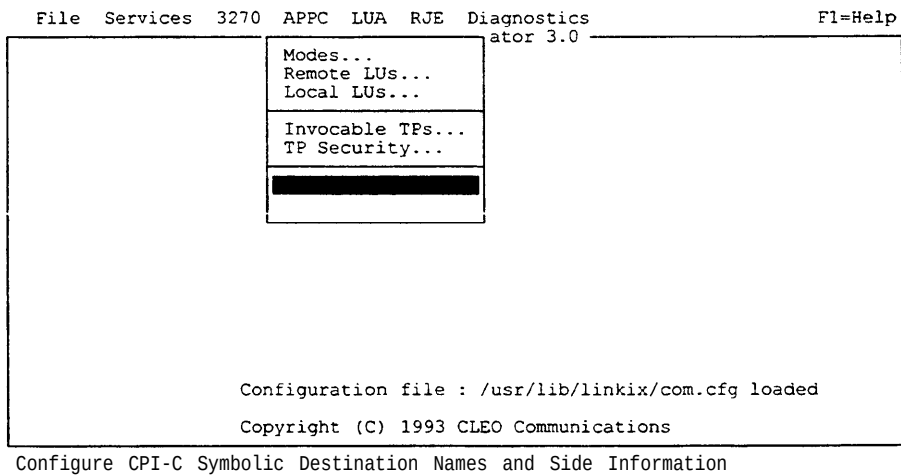
Note: User IDs and passwords are automatically converted from ASCII to EBCDIC for comparison with the security information received from the incoming Allocate generated by the invoking TP.

Configuring CPI-C Symbolic Destination Names;

Symbolic destination names are used in CPI-C programs to identify a particular combination of partner LU, mode, partner TP and security information. Instead of specifying each of these parameters individually on an Allocate call (as in APPC), a CPI-C application can simply specify a symbolic destination name. LINKix then obtains all the information required to access the partner TP from the configuration file entry for that name.

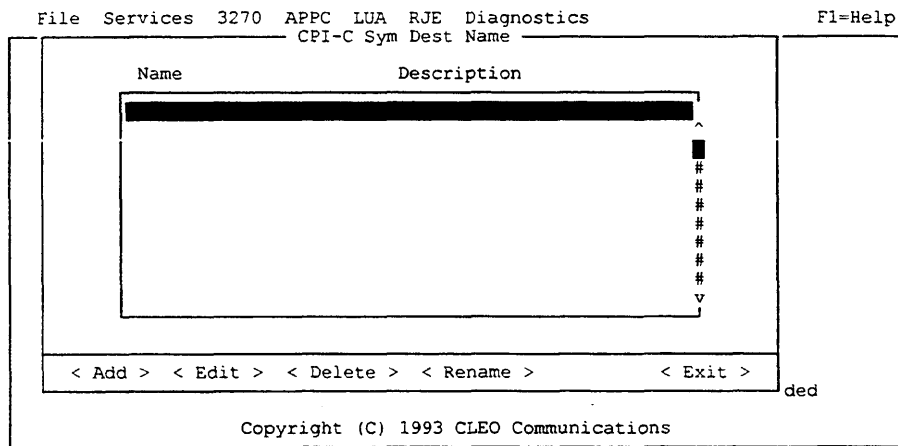
If CPI-C applications will be using symbolic destination names, you need to configure each name. (Note that a CPI-C application can set up the parameters for communications with its partner using a series of CPI-C calls, instead of using a symbolic destination name.)

To configure symbolic destination names, select Sym Dest Name from the APPC menu.



The CPI-C Symbolic Destination Name dialog is displayed.

Configuring for APPC



This dialog lists all configured symbolic destination names. The description of each symbolic destination name is also shown.

From this dialog you can add, modify, delete or rename CPI-C symbolic destination names.

Adding a Symbolic Destination Name

To add a symbolic destination name:

1. To add a symbolic destination name using the system defaults as the model for the new symbolic destination name, select [NEW] from the list box.

To add a symbolic destination name using an existing symbolic destination name as the model for the new name, select an existing symbolic destination name from the list box.

2. Choose < Add >. The Add a Symbolic Destination Name dialog is displayed.

File Services 3270 APPC LUA RJE Diagnostics F1=Help
CPI-C Sym Dest Name

Name	Description
[NEW]	
Add a CPI-C Symbolic Destination Name	
Symbolic destination name . . . [.....]	
< OK > <Cancel>	

< Add > < Edit > < Delete > < Rename > < Exit >

Copyright (C) 1993 CLEO Communications

3. Enter the name of the symbolic destination. This name can be one to eight characters long consisting of upper- or lowercase letters (lowercase letters are converted to uppercase) and the numbers 0 through 9.

This name cannot match any other symbolic destination name in this configuration.

4. Choose < OK >. The Configure Symbolic Destination Name dialog is displayed.
5. Complete the fields with the needed information. See **Configuring a Symbolic Destination Name** (p. 7-54) for details on the fields.

Editing a Symbolic Destination Name

To modify the configuration for an existing symbolic destination name:

1. Select the symbolic destination name you want to modify from the list box in the Symbolic Destination Names dialog.
2. Choose < Edit >. The Configure Symbolic Destination Name dialog is displayed with the existing configuration for the symbolic destination name.
3. Change the fields as necessary. See **Configuring a Symbolic Destination Name** (p. 7-54) for details on the fields.

Deleting a Symbolic Destination Name

To delete a symbolic destination name:

1. Select the symbolic destination name you want to delete from the list box in the Symbolic Destination Names dialog.
2. Choose < Delete >. The Delete Symbolic Destination Name dialog is displayed.
3. Choose < OK > to confirm the deletion or < Cancel > to abandon the deletion.

You return to the Symbolic Destination Names dialog (p. 7-52).

Renaming a Symbolic Destination Name

To rename a symbolic destination name:

1. Select the symbolic destination name you want to rename from the list box in the Symbolic Destination Names dialog.
2. Choose < Rename >. The Rename Symbolic Destination Name dialog is displayed, showing the existing symbolic destination name.
3. Enter the new symbolic destination name. This name can be one to eight characters long consisting of upper- or lowercase letters (lowercase letters are converted to uppercase) and the numbers 0 through 9.

This name cannot match any other symbolic destination name in this configuration.

4. Choose < OK >.

You return to the Symbolic Destination Names dialog (p. 7-52).

Configuring a Symbolic Destination Name

When you add or modify a symbolic destination name, the Configure Symbolic Destination Name dialog is displayed.

File Services 3270 APPC LUA RJE Diagnostics F1=Help
 CPI-C Sym Dest Name
 Configure Symbolic Destination Name FILEMGR

Description [Remote file management pr]

Partner TP name
 (o) Application TP [FILEMANAGER.....]
 () SNA Service TP (in hex) [.....]

Partner LU type (o) Alias () Fully Qualified

Partner LU: Mode Name:

TPLU1	^	LOCMODE	^
TPLU2	#		#
	v		v

< Ad

< Security > < OK > <Cancel>

1. Make whatever changes are necessary to the following options.

Description

The Description edit box contains the description of the symbolic destination name. This optional description may be up to 25 characters. It appears on the Symbolic Destination Names display screen and allows you to distinguish among confirmed symbolic destination names.

Partner TP name

The Partner TP name section specifies the TP name of the partner invocable TP (the TP name configured for this TP on the system where the partner TP runs).

Application TP

If the partner is an application TP, choose the Application TP radio button and type in the TP name. This name may be up to 64 characters long consisting of a-z, A-Z, 0-9 and . (period). For an APPC partner application, the following characters are also valid: # @ \$. For a CPI-C partner application, the following characters are also valid: < (+ & *); - / , % _ > ? : ' = " .

SNA Service TP

If the partner is an SNA service TP, choose the SNA Service TP radio button and enter the TP name. This name consists of two to eight hexadecimal byte values for the EBCDIC characters that make up the name.

Configuring for APPC

The first character must be between 0x00 and 0x3F. The other characters are from the EBCDIC AE character set: A - Z, a - z, 0 - 9, \$, #, @, and. (period).

The default type is Application TP, with no default for the TP name.

Partner LU type

The Partner LU type radio group specifies whether partner LUs are listed by LU alias or by fully qualified LU name. This affects the way LUs are displayed in the Partner LU list box (see below); the same LUs are displayed in both cases, but by different names.

Partner LU

This list box contains all remote LUs and all locally usable local LUs.

If you selected Alias for Partner LU type above, each LU is shown by its LU alias (up to eight characters). If you selected Fully Qualified, each LU is shown by its full network name (a network name of up to eight characters, a period and an LU name of up to eight characters).

Mode

This list box contains the modes that can be used to access the partner application. The list includes the names of all configured APPC modes. If a mode is configured with a blank mode name, the name appears as "()" at the top of the list.

When a local CPI-C application that accesses this partner application is started, it specifies a local APPC LU to use. The partner LU specified here must be configured as a partner of this local LU, using the mode specified here.

2. When you've entered all necessary information in this dialog, you're ready to configure the conversation-level security parameters. Choose < Security >.

The Configure Conversation Security for Symbolic Destination Name dialog is displayed. The title of this dialog box shows the symbolic destination name of the partner application.

```

File Services 3270 APPC LUA RJE Diagnostics Fl=Help
CPI-C Sym Dest Name
Configure Symbolic Destination Name FILEMGR
Configure Conversation Security for FILEMGR
Desc
Par Conversation Security:
  ( ) None
  ( ) Same
  (o) Program
Par User ID . . . . [fileadmin.]
  Password. . . . [.....]
  Check . . . . [.....]
  < OK > <Cancel>
< Ad
< Security > < OK > <Cancel>
  
```

- 3 . Make whatever changes are necessary to the following options.

Conversation Security

Specify whether or not the partner application uses conversation security and, if it does, how it uses it.

None	The partner application does not use conversation security.
Same	The partner application uses conversation security, but accepts an “already verified” indication from the local application (indicating that the local application was itself invoked by another application that specified a valid user ID and password).
Program	The partner application uses conversation security. If you select this option, you must supply a user ID and password in the other fields in this dialog.

User ID

This field contains the user ID required to access the partner application. This ID must match a conversation security user ID configured on the system where the partner application runs. This field may contain one to 10 characters consisting of A-Z, a-z, 0-9, . (period), #, \$ and @. This field is required if Conversation Security is set to Program.

Configuring for APPC

Password

This field contains the password required to access the partner application. This must match a conversation security password configured on the system where the partner application runs. Note that the password is not displayed on the screen as you type it. This field may contain one to 10 characters consisting of A-Z, a-z, 0-9, . (period), #, \$ and @. This field is required if Conversation Security is set to Program.

Password check

This field should contain the same password as that in the field above. It verifies whether or not the password was entered correctly. As before, the password is not displayed on the screen as you enter it. If the password you enter here does not match the password above, an error message is displayed. Choose the < Exit Message > pushbutton to acknowledge the message, then type the password in both boxes again.

10. Configuring Diagnostics

This chapter explains how to configure diagnostics for your system. The following topics are covered:

- ◆ How to specify whether NetView commands and alerts can be used, and which connection is used to communicate with the NetView host
- ◆ How to configure audit and error logs

For complete information on using the advanced diagnostics features, see chapter 13, *Advanced Diagnostics*.

Network Management Connection

Advanced NetView package required: Before you can access the Connection option of the Diagnostics menu and specify a network management connection, you must have purchased and installed the Advanced NetView package. If you find that you need the capabilities provided by the Advanced NetView package, call the CLEO sales department at (313) 662-2002.

The LINKix diagnostics features include the ability to designate a particular host connection as the Network Management connection. This connection is used for sending alerts to NetView and for executing NetView commands which allow you to log onto the host NetView program to view and control local software components or allow an operator at the NetView console to issue commands that will execute on the UNIX computer. This connection is also used for Network Management API (NM-API) applications, which can receive NMVTs from the NetView program and send NMVTs to it. For more information on NetView commands and NM-API applications, see the *NetView User's Guide*.

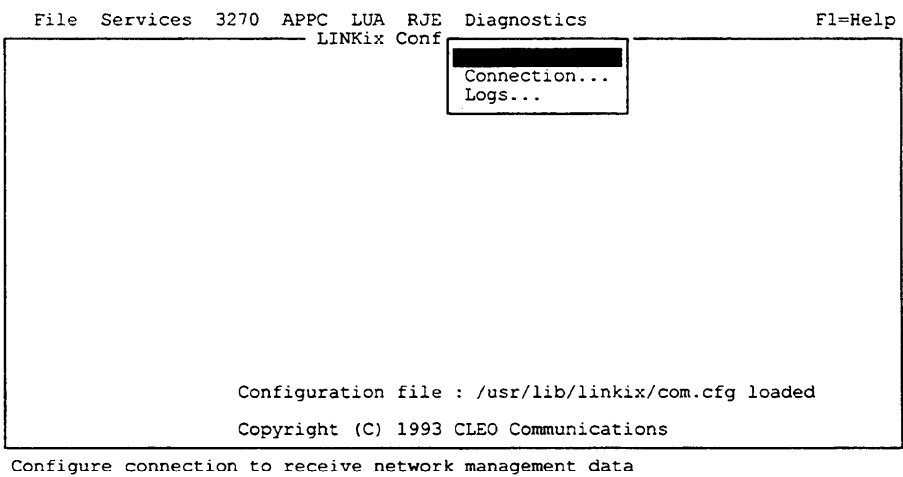
Alerts which are sent to the host on this Network Management connection can be created by the following methods:

- ◆ By LINKix internally (e.g., to notify of a link failure).
- ◆ By an application program using the TRANSFER_MS_DATA verb (one of the Common Service Verbs).
- ◆ By an NM-API application program.

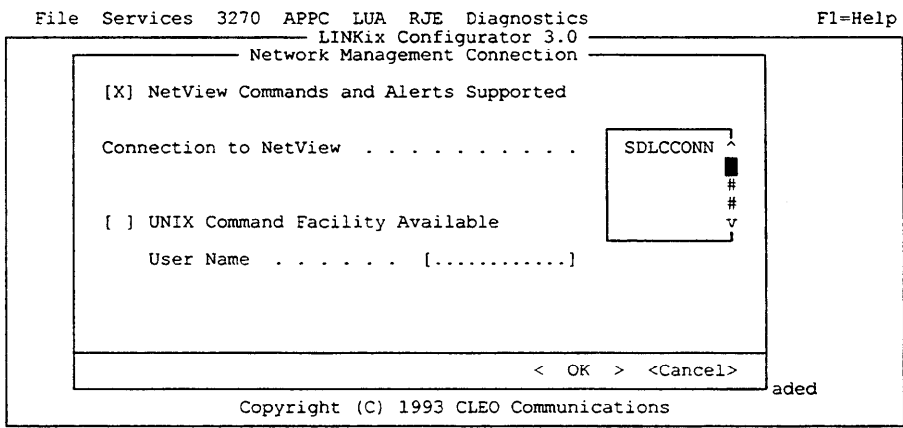
Note that 3270 user alerts are not sent on this connection; they are sent on a 3270 user's current emulation session, to the host for that session. See chapter 6, **Configuring for 3270**, for details.

You specify a connection to the host NetView program using the Connection option of the Diagnostics menu.

Configuring Diagnostics



The Network Management Connection dialog is displayed.



Configuring Diagnostics

Make whatever changes are necessary to the following options.

NetView Commands and Alerts Supported

This check box specifies whether or not network management features are enabled. If you do not select it, alerts will not be sent to a host and you will not be able to use NetView commands; also, all the other fields in this dialog will be disabled (grayed). If you select this option, you must use the Connection to NetView field to specify a connection for Network Management data.

Connection to NetView

The Connection to NetView list box displays the host connection to be used for Network Management data. All host connections in this configuration that are owned by a local node are listed in the list box. To specify the Network Management connection, select the connection from the list box.

This field is disabled (grayed) if the previous check box is not selected. Note also that Network Management data cannot be sent to peer connections or to a connection that is not owned by a local node.

UNIX Command Facility Available

This check box specifies whether or not the UNIX Command Facility (UCF) can be used. UCF is a subset of NetView Commands that allow an operator at the NetView console to issue commands that execute on the UNIX server on which the network management daemon is running. To ensure that NetView operators do not have unreasonable access to information on the UNIX computer or the ability to execute potentially destructive UNIX commands, the security features provided by UNIX must be used in conjunction with this facility. If you intend to allow use of this facility, refer to **UCF Security** for information.

This field is disabled (grayed) if the NetView Commands and Alerts Supported check box is not selected.

User Name

This edit box contains the user name to be used for executing UCF commands. This must be a valid user ID on the UNIX computer on which the network management daemon will run. When a command is issued using the UCF, a shell is started for this user ID (with the environment and permissions of this user), and the command is run in this shell. Refer to the section **UCF Security** for more information.

This field is disabled (grayed) if the UNIX Command Facility Available check box is not selected.

UCF Security

Because the UCF allows a remote operator to issue commands on the UNIX computer and to receive output from these commands, it is important to consider the security implications. For example, you need to ensure that the operator cannot access private information or issue UNIX commands which may disrupt other users.

These security considerations require you to configure a specific UNIX user name as the UCF user; all UCF commands are executed with this user's ID and, therefore, with the access permissions of this user. The commands run on the LINKix server computer on which the network management daemon is running; the UCF user name must be a valid login ID on this computer.

It is intended that you use the normal security features provided by UNIX to restrict the commands that are accessible by the UCF user, in order to allow only those commands you consider reasonable for use from UCF. The following guidelines may be useful:

- ◆ The UCF user name should be one which is used solely for UCF; you should not use an existing login which is also used for other purposes. This makes it easier to define the privileges of this user to include only those which are reasonable for UCF; it also allows you to identify processes which were started using UCF.
- ◆ You may need to restrict the users and groups to which the UCF user can change user ID or group ID. In particular, *the UCF user must not be allowed to become root or superuser.*

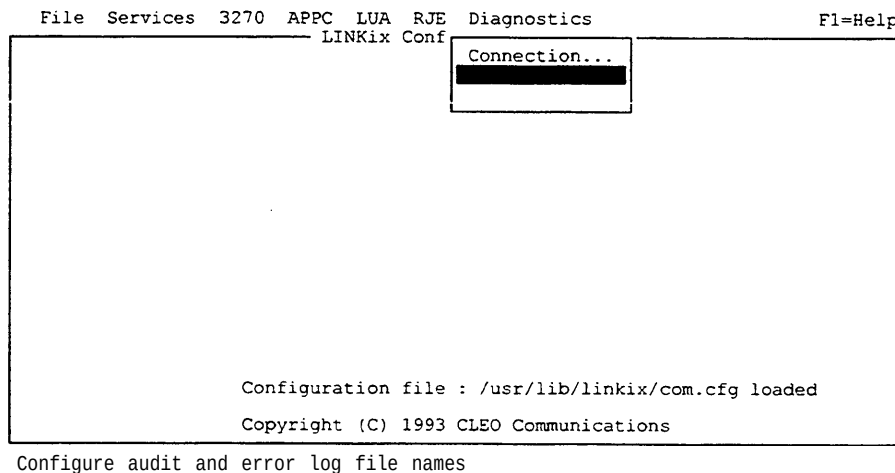
Configuring Audit and Error Logs

Diagnostic information is recorded at two levels: audit log (which provides a record of the normal operation) and error log (which provides warning and error information). This information is logged in the local audit and error log files. In addition, alerts can be sent to a host for analysis and diagnosis (these are also logged in the local audit log file for information).

You can specify files for the system to write all audit and error information to when you configure your system.

Audit log files provide information on the software's internal operation and are useful as system diagnostics. Error information reports error conditions in software operation.

Use the Logs option of the Diagnostics menu to specify where error and audit log messages will be logged on the local computer.



The Audit/Error Logs dialog is displayed.

```

File Services 3270 APPC LUA RJE Diagnostics      F1=Help
LINKix Configurator 3.0

Audit/Error Logs

Audit log . . . . [/usr/lib/linkix/com.aud.....]

Default audit log level:
(o) Detailed problem analysis data (6)
( ) General information messages (8)
( ) Significant system events (10)
( ) Audit logging disabled

Error log . . . . [/usr/lib/linkix/com.err.....]

< OK > <Cancel>

Configuration file : /usr/lib/linkix/com.cfg loaded
Copyright (C) 1993 CLEO Communications
  
```

Enter whatever information is necessary in the following fields.

Audit log

The Audit Log edit box specifies the path and the name of the file you want to use as an audit log. Audit log file names must have an extension of *.aud* or *.log*. The path name and file name can be up to 127 characters long.

You can send both audit log and error log information to the same file. To do so, you must provide the same name for both the error and audit log files, and include the extension *.log*.

If you do not specify a path for the file, the file is opened in the directory from which you started the control daemon (see chapter 11, **Managing LINKix Components**, for details).

Default audit log level

This radio group lets you choose the level of information you want recorded in the audit log.

Detailed problem analysis data (6) Log all messages at levels 6, 8 and 10.

General information messages (8) Log all messages at levels 8 and 10.

Configuring Diagnostics

Significant system events (10) Log all messages at level 10 only.

Audit logging disabled Suppress all level 6-10 audit log messages.

Level 6 messages provide very detailed information which is useful for problem tracing. Level 8 provides general informational messages of system events. Level 10 provides only significant system events. Note that level 12 and 16 messages are always recorded in the error log 6 you cannot suppress these messages.

Error log

The Error Log edit box specifies the path and the name of the file you want to use as an error log. Error log file names must have an extension of error log. The path name and file name can be up to 127 characters long.

You can send both audit log and error log information to the same file 6 to do so, you must provide the same name for both the error and audit log files, and include the extension. log.

If you do not specify a path, the error log file is opened in the directory from which you started the control daemon (see chapter 11, **Managing LINKix Components**, for details).

Resetting the Audit and Error Log Files

While the audit and error logs capture important information, that information becomes outdated as time passes and, as the file grows in size, it consumes an increasing amount of disk space. To ensure that log files do not become too large and take up too much disk space, when a log file exceeds one megabyte it is copied to *bak.err* (for an error log file) or *bak.aud* (for an audit log file), in the same directory as the existing file, and the file size is reset to zero. This ensures that a log file never takes up more than approximately two megabytes of disk space.

11. Managing LINKix Components

This chapter explains:

- ◆ How to start and stop the local node
- ◆ How to start and stop the control daemon
- ◆ How to deal with daemon startup problems
- ◆ How to manage services from the command line
- ◆ How to use the V.25bis autodial utilities

Starting/Stopping the Control Daemon

The control daemon is a UNIX user space process that is not attached to any terminal or user ID. It manages the initial interaction between processes in user space and those in kernel space, and it controls the error log daemon.

The control daemon, *snadaemon*, controls:

- ◆ Local nodes (PU 2.1)
- ◆ Links
- ◆ The diagnostics daemon program, *comerrlog*

The configuration program, however, can be used without the daemon.

After the control daemon is started, you can start any configured local nodes or links, and the network management daemon (provided you have configured a network management connection). You can then stop and restart services as necessary while the control daemon is running (see **Starting/Stopping LINKix Services** for details).

Starting the Control Daemon

Before starting the control daemon, you should ensure that the configuration file */usr/lib/linkix/com.cfg* (and its associated security file *com.see*) contains the correct information for your configuration and that they are located in the proper directory. The information in these configuration files is used when you start the control daemon; an error will be returned if the files cannot be opened.

To start the control daemon, at the UNIX command prompt type:

```
comstart [-c configfile] [-t string] [-k] [-s]
```

The *-c* option specifies the configuration file that the control daemon will use when it is started. If a configuration file is not specified when the daemon is started, the default configuration file (*/usr/lib/linkix/com.cfg*) is used, if it can be found. If the configuration file cannot be found, an error is returned.

If the `-t` option is specified, the control daemon will assign one or two trace files for tracing the kernel space components (local node and link). By default, trace data is stored to the two files `comfile1.trc` and `comfile2.trc` in the directory `/usr/lib/linkix`.

The string should contain:

```
file1 [ : file2]
```

where *file1* and *file2* are the names of trace files. The second file name is optional; if it is used, a colon must separate the two file names.

If you specify one file name only, there is no limit on the size of the trace file that will be created.

If you specify two file names, the first trace file will holdup to 250,000 bytes of trace data. When the first file reaches 250,000 bytes in size, trace information is written to the second file. When the second file reaches 250,000 bytes, trace information is redirected back to the first file and overwrites the existing data. Using two files allows tracing to be active for an extended period of time while using a maximum of just 500,000 bytes of disk space.

Warning~Trace data must always be directed to one or more text files. *Do not specify the name of a device (such as `/dev/tty`), or a print spooler, as a trace filename.* Doing so may cause corruption of vital UNIX system files.

Since tracing degrades performance, it should only be used when it's specifically needed for diagnosis.

If problems are encountered during control daemon initialization, these are reported in the following ways:

- ◆ If a fatal error occurs (one which prevents the control daemon from starting), an error message is written to the standard error device; the daemon then terminates with a non-zero return code. The return code indicates the nature of the error. See appendix C, Exit Codes, for more information.

If you create a shell script to start the control daemon, your script can test the exit code to determine whether the initialization was completed successfully.

Managing LINKix Components

- ◆ If a non-fatal error occurs (one which does not prevent the daemon from starting but which may affect its operation), the program writes a warning message to the error log file. Error log messages are explained in greater detail in chapter 13, **Advanced Diagnostics**.

The `-k` option enables tracing on the SNA LAN Interface Module (SLIM). The trace output is sent to the same files as for other service components. If this option is not specified, SLIM tracing is inactive. For information on tracing LINKix services, see **Managing Services from the Command Line**.

The `-s` option specifies that messages should not be written to the system console. (These messages are normally used to request operator action or to warn the operator of error conditions.)

Stopping the Control Daemon

Stopping the control daemon stops all processes that are using it, including any active services. You should warn any active users, including anyone using the configuration program, that the control daemon is about to be stopped, giving them time to finish their activities before you actually stop the system.

To stop the control daemon, at the UNIX command prompt type:

```
comstop
```

The *comstop* program returns the exit code 0 if it completes successfully; it returns 1 if the software was not active.

If the 3270 emulation program is running when you stop the control daemon, the 3270 emulation will automatically exit. Applications using APPC, CSV or LUA are notified by a `COMM_SUBSYSTEM_ABENDED` return code, CPI-C applications by a `CM_PRODUCT_SPECIFIC_ERROR` return code, HLLAPI applications by a `HARC_SYSTEM_ERROR` return code and NM-API applications by a `COMM_SUBSYSTEM_NOT_LOADED` return code.

Starting/Stopping LINKix Services

In order to use most LINKix features (e.g., 3270 emulation, 3770 emulation, APPC applications, etc.), both the control daemon and the appropriate node and link must be started.

Starting Services

To start a local node or link, enter the following command at the UNIX command prompt (after the control daemon has been started):

```
startsrv service [ -b boxname]
```

where the following items are specified:

service	This is the name of the service component, as specified in the LINKix configuration, that you wish to start. Specifically this would be the node or link name (e.g., SDLCNODE or SDLCLINK).
-b boxname	This option indicates that the service is to be started on a different server on the LAN. <i>boxname</i> is the name of the server (it is not case sensitive). This option is only needed in a client/server environment. If you do not specify this option, LINKix attempts to start the service on the computer on which you issued the <i>startsrv</i> command.

The *startsrv* command returns an exit code of zero if it completes successfully. If an error occurs, the exit code is non-zero. LINKix also writes a message to standard error, indicating the reason for the failure.

Example:

To start the control daemon, then a link and finally a node:

```
comstart
startsn SDLCLINK
startsrv SDLCNODE
```


Stopping Services

If you wish to stop a service without stopping the LINKix control daemon, enter the following command at the UNIX command prompt:

```
stopsrv service
```

Where *service* is the name of the service component that is currently running, that you wish to stop. Specifically this would be a node or link name (e.g., `SDLCNODE` or `SDLCLINK`).

The *stopsrv* command returns an exit code of zero if it completes successfully. If an error occurs, the exit code is non-zero. LINKix also writes a message to standard error, indicating the reason for the failure.

Note: Stopping the control daemon stops all processes that are using it, including any active services. Therefore, you do not need to use *stopsrv* before doing a *comstop*.

Daemon Startup Problems

If you cannot start the control daemon, check for one of the following errors.

Error:	The daemon is not running as root.
Correction:	The daemon must be owned by root and have the necessary access permissions (r-sr-s---) to start. Connect the access permissions, login as root and retry.
Error:	Unable to open the configuration file. Either the configuration file, <i>com.cfg</i> , is not in <i>/usr/lib/linkix/</i> , it does not have the proper access permissions or it's corrupted.
Correction:	Use the -c switch (<i>comstart -c cfgfile</i>) to specify the configuration file that should be used, repeat the configuration steps in part III of this manual or copy <i>/usr/lib/linkix/samples/com.cfg</i> to <i>/usr/lib/linkix/com.cfg</i> and make the necessary changes to file access permissions (rw-rw----).
Error:	Unable to open security file. Either the security file, <i>com.sec</i> or whatever name you specified during configuration (it must be the same as the configuration file), is not in <i>/usr/lib/linkix/</i> , the file does not have the proper access permissions or it is corrupted.
Correction:	Repeat the configuration steps in part III of this manual or copy <i>/usr/lib/linkix/samples/com.sec</i> to <i>/usr/lib/linkix/com.sec</i> and make the necessary changes to file access permissions (rw-rw----).
Error:	Unable to open the link device. The link device (e.g., <i>/dev/sna_SDLC</i>) or the local node device (e.g., <i>/dev/com_NODE</i>) could not be opened. Either the device name was changed during confirmation (and is now misspelled), the device is missing or the file does not have the proper access permissions.
Correction:	The device name that appears in the configuration file must match the name of the device in the <i>/dev/</i> directory. Correct the device name in the configuration if it's misspelled; if the device is missing from the <i>/dev/</i> directory, reinstall the product; or, make the necessary changes to file access permissions (rw-rw-rw-).

Managing Services from the Command Line

You can view and control the level of diagnostics information recorded for LINKix services in two ways: either from the command line or using the Manage program (the Manage program is provided with the LINKix Management Utilities package). This section explains how to manage services from the command line.

To view or change the diagnostics levels of a service, enter the following command at the UNIX command prompt:

```
comservice servicename [ - t trcflags | -c ] [ -a auditlevel ]
```

<i>servicename</i>	Identifies which service is being managed. This is the name of the local node or link (which must match the name of an active service), BSDSLIM for the LINKix SNA LAN Interface Module or NETMAN for the network management daemon. This name is not case sensitive.
<i>-t trcflags</i>	Specifies tracing options for the service. There are different options available for each service type; see the table below for more information. This option cannot be used with the -c option. For service types where there is more than one trace option, -t can be used more than once to specify multiple trace types at the same time.
<i>-c</i>	Disables all tracing for the service. This option cannot be used with the -t option.
<i>-a auditlevel</i>	Specifies the audit log level for the service: 6,8 or 10, or 0 to disable audit logging for the service.

If no options are specified, *comservice* prints messages describing the current trace level and audit level for the service.

Any other message indicates an error condition and begins either with *öSyntax error* or *öError,i* followed by information on the cause of the error. Syntax error messages indicate that you entered an invalid option or parameter or that you did not supply a required one; check for the correct syntax and retry. Error messages indicate that the command syntax was correct, but that the command could not be issued (for example because the service was not started or because an operating system error occurred); take the required action and retry.

The *comservice* command returns an exit code of zero if it completes successfully. If an error occurs, the exit code is non-zero. For explanations of the exit codes which may be returned, see the *LINKix Diagnostics Guide*.

Tracing Options

The following table describes the tracing options that can be used.

Service type	Options	Explanation
Node	sna	Trace SNA messages sent and received by the node. (Cannot be used with the internal trace flag.)
	internal	Trace all messages sent and received by the node - both SNA and LINKix internal messages. (Cannot be used with the sna trace flag.)
SDLC or X.25 link	Com	Connection tracing - trace data flowing between the node and this link.
	lev2	Link tracing - trace the data flowing on this link.
	all	Trace all messages flowing on this link.
Vendor X.25 link	com	Connection tracing - trace data flowing between the node and this link.

Managing LINKix Components

Service type	Options	Explanation
Token Ring link	c o m	Connection tracing - trace data flowing between the node and this link.
	dlpi	Trace the Data Link Provider Interface, which is an industry-standard interface between different levels of the Token Ring software.
	lli	Trace the data sent between the Token Ring software and the adapter card.
	all	Trace all messages flowing on this link.
Network management daemon	internal	Trace all messages sent and received by the network management daemon and the UNIX Command Facility daemon.
Queued TP	api	API tracing - trace data flowing between the TP and the APPC or CPI-C library.
SLIM	all	Trace all messages sent and received by the LINKix SNA LAN Interface Module.

Examples

To manage the SDLCNODE, tracing all messages sent and received by the node, and setting audit logging at level 6:

```
comservice sdlcnode - t internal -a 6
```

To manage the SDLCLINK, tracing connection and link data and setting audit logging disabled:

```
comservice sdlclink - t com -t lev2 -a 0
```

To manage the network management daemon, with both tracing and audit logging disabled:

```
comservice netman -c -a 0
```

V.25bis Dialer

Customers who are using a V.25bis autodial modem will find the `v25dial` and `v25cmd` utilities useful. These files are installed in `/usr/lib/hmod`, along with a number of other files. To use `v25dial` and `v25cmd`, you should add `/usr/lib/hmod` to your PATH, or copy the utilities to a directory that is in your PATH.

Using v25dial

The `v25dial` utility is used to send command strings (via the LINKix HMOD) to a modem that supports the V.25bis dialing protocol. Before you can use `v25dial`, a successful `comstart` must be performed and the LINKix control daemon must be running. You then run `v25dial` to temporarily gain access to the communications port to send a dial command string to the modem. When the modem sends its response string, `v25dial` relinquishes control back to LINKix and the modem's response is returned as an exit code. If you are operating `v25dial` in verbose mode, a text string is displayed on the screen.

The syntax for using `v25dial` is as follows:

```
v25dial [-d string] [-h] [-p port] [-t timeout] [-v]
```

where the options may be one or more of the following:

<code>-d <i>string</i></code>	Sends the specified command string to the modem. If the string has white space in it, you must enclose it in quote marks. If you do not specify a command string with this option, <code>v25dial</code> is started in interactive mode and displays the following prompt: Enter dial command string (q to quit) : >>
<code>-h</code>	Displays the usage message.
<code>-p <i>port</i></code>	Specifies the port number that the dial strings will be sent to. If not specified, the port defaults to 1.

- t timeout* Specifies a timeout value (in seconds). If the modem does not respond within this time, *v25dial* cancels execution and returns an appropriate return code. This value may range from 1 to 1000; the default is 60.
- v* Verbose mode. Specifying this mode causes *v25dial* to display a message on the screen with the return code for the operation.

Example Commands

The following are examples of *v25dial* commands.

v25dial

Starts *v25dial* in interactive mode, commands are sent to port 1 and the timeout value is 60 seconds. Enter one or more modem setup or V.25bis commands, pressing **ENTER** at the end of each command. Press **q** to quit.

v25dial -d"CRN 13135551212"

Sends the V.25bis command that dials the specified phone number (1.313.555.1212).

v25dial -d"PRN 1;13135551212"

Sends the V.25bis command that stores a phone number (1.313.555.1212) to address 1.

v25dial -d"CRS 1"

Sends the V.25bis command that dials the phone number stored at address 1.

v25dial -dRLN

Sends the V.25bis command that requests a list of stored numbers. The modem will return a list of all numbers that are currently stored.

Refer to your modem documentation for complete details on the V.25bis commands that you can use, or see **Some Common V.25bis Commands** on p. 11-15.

The dial Script

We've also provided a sample script, *dial*, that shows one way to use *v25dial* to dial a stored number. This script is installed in */usr/lib/hmod*.

```
while true
do
    /usr/lib/hmod/v25 dial -dcrl
    if [ $? -eq 254 ]
    then
        sleep 1
    else
        exit 0
    fi
done
```

Using v25cmd

The *v25cmd* utility opens the communications port for exclusive use & this prevents LINKix from opening a connection to the modem at the same time. This utility is useful for setting up the options for the modem.

The syntax for using *v25cmd* is as follows:

```
v25cmd [-p port] [-v] [-h]
```

where the options may be one or more of the following:

-p <i>port</i>	<i>Specifies the port</i> number that the command strings will be sent to. If not specified, the port defaults to 1.
-v	Verbose mode. Specifying this mode causes <i>v25cmd</i> to display messages on the screen for each operation.
-h	Displays the usage message.

After you've started *v25cmd*, you enter the modem command strings at an interactive prompt. If you specified the -v option, the results of each command are displayed in messages on the screen. When finished, enter *oqi* to quit.

Some Common V.25bis Commands

The following are some common V.25bis commands that you will find useful:

CRN <phone number>

Dial command. This command causes the modem to dial the specified phone number.

PRN <#>;<phone number>

Program number command. This command causes the modem to store the given phone number at the specified address*.

CRS <#>

Dial stored number. This command causes the modem to dial the phone number that's stored at the specified address*.

RLN

Request list of stored numbers. This command causes the modem to return a list of all the phone numbers that are currently stored.

For complete details on these and other V.25bis commands, refer to the documentation that accompanied your modem.

Return Codes

When sending a command to the modem, *v25dial* will return one of the following exit codes.

0	Command executed successfully
2	Call failure - engaged tone
3	Call failure - number not stored
4	Call failure - local DCE busy
5	Call failure - timeout on ring tone
6	Call failure - abort call on timeout
7	Call failure - answer tone not detected
8	Call failure - forbidden call
9	Call failure - no dial tone detected
10	Call failure - unknown response
11	Invalid command - command unknown
12	Invalid command - message syntax error
13	Invalid command - parameter syntax error
14	Invalid command - parameter value error
15	Invalid command - unknown response
16	No CTS - modem did not assert CTS
17	The <i>v25dial</i> utility is already running
18	Function has timed out
19	Connection is closed
20	Connection already established
21	Diagnostics being performed
22	Dial string not successfully transmitted - try
23	Internal error (HAPI closed)
24	The V25BIS service is not available

12.Troubleshooting

This chapter contains information that can be used to resolve problems you may encounter when installing, configuring and using LINK.ix. The information is divided into the following sections:

- ◆ Troubleshooting LINKix 3270
- ◆ Troubleshooting LINKix 3770
- ◆ It Was Working Just Fine . . .
(Dealing with Subsequent Problems)

Troubleshooting LINKix 3270

This section provides solutions to the most common problems that you may encounter when attempting to use LINKix 3270 for the first time. If the suggestions listed here do not solve your problem, you may wish to turn to chapter 13, **Advanced Diagnostics**, for more advanced troubleshooting instructions.

3270 Emulation Problems

The problems 3270 emulation program (*com3270*) allows your UNIX terminal to emulate an IBM 3278/3279 display terminal or a 3287 printer. The emulation program can be started without enabling a 3270 session. If a 3270 session is not automatically enabled, you'll see the main screen of the 3270 emulation program. If LINKix 3270 has been configured to enable a 3270 session at startup, however, you should see an enabled 3270 session screen with a status line at the bottom.

If the 3270 emulation program will not start, check for one of the following errors.

Error:	Unable to open the configuration or security file.
Correction:	Correct the access permissions to the files <i>/usr/lib/linkix/com.cfg</i> and <i>/usr/lib/linkix/com.sec</i> (rw-rw----).
Error:	<i>Program</i> not found. Your PATH environment variable does not include the directory that contains the program or the program is not installed on your system.
Correction:	Make sure that <i>c3270</i> and <i>com3270</i> are installed in <i>/usr/bin/</i> and that your PATH includes that directory.
Error:	If the 3270 emulation program runs when you are logged in as root but does not operate when logged in as a user, the file <i>c3270</i> or <i>com3270</i> may not have proper access permissions.
Correction:	Correct the file access permissions as needed (r-xr-xr-x).
Error:	No 3270 users in the configuration file. None of the users of the UNIX computer have been configured as 3270 users.

- Correction:** Use the configuration program to configure at least one 3270 user (this can be the <DEFAULT> user record). See chapter 6, **Configuring 3270 Emulation**, for more information.
- Error:** User not found in the configuration file. You are not included in the configured list of 3270 users or you've logged on with a different *userID* than the one specified in the configuration file.
- Correction:** Login using the proper *userID* or run the configuration program to add the appropriate user information.
- Error:** No 3270 LUs in the configuration file. You are confirmed as a 3270 user but there aren't any sessions configured or assigned for your use.
- Correction:** Use the configuration program to configure one or more LUs, or to assign a user to a configured LU. See chapter 6, **Configuring 3270 Emulation**, for more information.

If you're experiencing a problem that's not covered here, turn to chapter 13, **Advanced Diagnostics**, for information on advanced techniques to problem resolution.

3270 Session Problems

Once the 3270 emulation program is successfully started, you connect to a host system and enable a 3270 display session. Depending on the settings that were specified during configuration, LINKix 3270 may try to enable a session when the emulation program is initially started.

In the problem resolution advice that follows, the *communications port* refers to the port on the communications adapter board (for SNA products) or the port on the 3x74/76 cluster controller (for DFT products). In addition, take note that the DFT product has two different cables: 1) the single wire coax cable from the communications adapter board to the controller and 2) the synchronous cable that's attached between the controller and the other device (modem or host system).

If you are using the LINKix 3270 DFT product, an important preliminary step to problem resolution would be to check if any other devices attached to the 3x74/76 cluster controller are working. If the problem is wide spread among devices, this would indicate a problem with the controller itself.

Troubleshooting

Cannot Enable 3270 Session

If 3270 emulation is successfully started but the status line contains a communication check code message that indicates the session is not properly enabled (e.g., -+z_504), check for one of the following errors.

Error	The cable to the communications port maybe disconnected or connected to the wrong board.
Correction:	Inspect the cable attachment at the back of the UNIX computer and, in the case of the DFT product at the back of the 3x74 controller. If the cable is not connected, connect it. Also, make sure that the cable is connected to the proper communications port.
Error:	The cable attached to the communications port maybe the wrong type or configured incorrectly. The cable should be shielded, synchronous-compatible, RS232C of no more than 50 feet in length or single wire coaxial (RG62AU Coax), depending on the type of communications board installed. If the cable is RS232C that connects to a modem or modem eliminator, pins 1 through 8, 15, 17 and 20 must be connected in a straight-through fashion.
Correction:	If the cable is of the wrong type or configuration, replace it with one of the correct type with proper pin assignments where needed.
Error:	The cable to the communications port maybe damaged.
Correction:	The best approach, when possible, is to borrow a cable from a similar working synchronous communications device, replace the suspect cable with this cable and determine if cabling is the problem. If the cable attached to the communications port is damaged, replace the cable.
Error:	The modem is not turned on or the cable is not properly attached to the modem.
Correction:	Check to be sure that the modem is turned on and that the cable is properly attached.
Error:	The cluster controller (DFT product) is not turned on or one or more cables are not properly attached to the controller-

- Correction:** Check to be sure that the controller is turned on and that the cables are properly attached.
- Error:** The modem is not receiving a DTR (Data Terminal Ready) signal on pin 20, the communications port is not receiving a DSR (Data Set Ready) signal from the modem on pin 6 or there's no clocking signal on pins 15 and 17.
- Correction:** All three of these problems are typically caused by a damaged or misconfigured cable, although clocking can be the result of attaching an incompatible modem (asynchronous as opposed to synchronous) or using an incorrectly configured modem eliminator. Replace the cable as necessary. If a modem eliminator is being used, be sure it provides a proper clocking signal on pins 15 and 17.
- Error:** The modem is the wrong type (asynchronous) or it doesn't match the modem at the other end.
- Correction:** Be sure the modem you're using with LINKix 3270 is synchronous and that it matches the one at the other end of the line.
- Error:** Configuration parameters may be incorrect. These include the modem encoding scheme (NRZ/NRZI), duplex settings (full/half), LU/device numbers, CU (control unit) or SDLC address, etc.
- Correction:** Be sure that the configuration settings are correct for your installation. Check with host personnel where necessary, and refer back to chapters 4 through 6 for complete details on how to run the configuration program and correct the necessary settings.
- Error:** The host computer may be experiencing problems. For example, it may be down or the host port may be inactive.
- Correction:** Contact host personnel.

Troubleshooting

3270 Session Enabled, Keyboard Problems

If a 3270 session is successfully enabled, but you find the keyboard is locked or does not respond as it should (e.g, function keys are not working), check for one of the following errors.

Error	Message on status line: X - f (Function Unavailable) You pressed an invalid key; for example, you used a program key such as F1 when the host SSCP is waiting for a logon message.
Correction:	Press RESET (Esc R) and continue with login or other activity.
Error:	Message on status line: X ?+ (Keystroke Lost) A keystroke has been lost, usually as the result of typing at a faster rate than can be serviced.
Correction:	Press RESET (Esc R) and continue.
Error:	Message on status line: PROG nnn (Program Check code) A protocol error was detected in the data coming from the host. The program check code, <i>nnn</i> , indicates the type of error (see Program Check Codes for complete details).
Correction:	Press RESET (Esc R) and continue. If the error recurs, make note of the check code and contact your system administrator.
Error:	Function keys on an IBM 3151 terminal are not performing the correct actions.
Correction:	The turnaround character, defined in terminal setup, must be öCR.ì Specifying any other character causes the function keys to respond incorrectly. Refer to the documentation that accompanied the 3151 terminal or turn to chapter 3 of this manual for instructions on viewing and changing the status of the turnaround character.

If you're experiencing a problem that's not covered here, refer to **Communication Check Codes** in Appendix B of the LINKix 3270/3270 *User's Guide* or turn to chapter 13, Advanced Diagnostics, for information on advanced techniques to problem resolution.

Troubleshooting LINKix 3770

This section provides solutions to the most common problems that you may encounter when attempting to use LINKix 3770 for the first time. If the suggestions listed here do not solve your problem, you may wish to turn to chapter 13, Advanced Diagnostics, for more advanced troubleshooting instructions.

Error:	Workstation does not start, cannot access output directory. The output directory specified in the workstation configuration must be present when the workstation is started.
Correction:	The access permissions for the directory probably do not allow the workstation to write to it; change the access permissions as needed.
Error:	Workstation does not start, <i>username</i> is not a valid login ID on the UNIX computer. This error can also occur with several other RJE programs, such as the console program or one of the spool control commands.
Correction:	The primary user assigned to the RJE workstation is no longer a valid user. Either add this <i>username</i> to the list of login IDs on the UNIX computer or change the primary user for this workstation.
Error:	Workstation does not start, <i>groupname</i> does not exist. This error can also occur with several other RJE programs, such as the console program or one of the spool control commands.
Correction:	The RJE workstation configuration names a group ID that is no longer a valid group ID on the UNIX computer. Either create <i>groupname</i> on the UNIX computer (with all the necessary users) or change the group ID for this workstation.
Error:	Workstation does not start, <i>user-name</i> not in group. The primary user for an RJE workstation is not in the user group configured for the workstation. This message may also be logged by several other RJE programs, such as the console program or one of the spool control commands.
Correction:	Either change the user and group setup on the UNIX computer, or modify the RJE workstation's configuration so the primary user is a

Troubleshooting

member of the configured group.

Error: Workstation does not start. Cannot execute the program to which output is routed.

Correction: This error generally occurs when the access permissions on the program do not allow the workstation to execute it; change the access permissions if necessary.

Error RJE program failed to open the configuration file. This error can occur with several different RJE programs, such as the console program or one of the spool control commands.

Correction: Check that the configuration file exists and has the correct access permissions, and that it is not corrupted (try opening it using the configuration program).

Error: RJE program failed when attempting to read the configuration file. This error can occur with several different RJE programs, such as the console program or one of the spool control commands.

Correction: Check that the configuration file exists and has the correct access permissions, and that it is not corrupted (try opening it using the configuration program).

Error RJE program could not find workstation. This error can occur with several different RJE programs, such as the console program or one of the spool control commands.

Correction: The workstation name supplied to an RJE command was not found in the configuration file. Check the name of the workstation in the configuration file to be sure it is correct and retry the command.

Error: The workstation name supplied to an RJE command was longer than the maximum length of four characters. This error can occur with several different RJE programs, such as the console program or one of the spool control commands.

Correction: Check the name of the workstation in the configuration file to be sure it is correct and retry the command.

- Error:** RJE base directory does not exist on the UNIX computer. This directory was created during installation.
- Correction:** More than likely the directory has been renamed or removed. Add the missing directory or re-install LINKix 3770.
- Error:** Reader spool directory for the RJE workstation cannot be found. The directory may have been deleted.
- Correction:** Restart the workstation, or submit another RJE job for this workstation. Either of these actions will rebuild the workstation's directory structure.
- Error:** The RJE workstation could not open a job file. The file maybe corrupted or may have been deleted.
- Correction:** Cancel and resubmit the job. Note that the `snajobcan` command may not work with this error if so, delete the file manually.
- Error:** The RJE workstation encountered an error while trying to send a job to the host.
- Correction:** This error may indicate a problem with the host connection, or the host may have canceled the job. Check for connection problems. The RJE workstation will retry the submission (up to 10 times).
- Error:** The RJE workstation could not open the status file or it does not match the job file. The file may be corrupted, or may have been deleted.
- Correction:** Cancel and resubmit the job. Note that the `snajobcan` command may not work in this case; if so, delete the files manually.
- Error:** The RJE workstation could not read the code conversion table file */usr/lib/linkix/rjecctab*.
- Correction:** The file may have been deleted or corrupted, or it may have the wrong access permissions.
- Error:** An RJE program does not start because it does not have the correct file ownership and access permissions.

Troubleshooting

- Correction:** Check the file ownership of RJE programs and correct them as necessary; then retry the command.
- Error:** Host has attempted to write to an unconfigured device or a printer or punch device which is not included in the configuration file.
- Correction:** Check the details of the required device with your host personnel and add the necessary information to the configuration.
- Error:** RJE LU logon is rejected by the host. When the RJE workstation attempted to logon to the host, the host rejected the logon attempt. This error occurs only if the workstation startup method is configured as "Workstation logon."
- Correction:** Check that the logon string configured for the workstation is correct. This error may also occur if there is a configuration mismatch between LINKix 3770 and the host for this LU number.
- Error:** An error occurs while processing program output from the printer or punch device. Either an error occurred while attempting to execute the output program configured for the device (e.g., *lp*) or the output was successfully routed to the program, but an error occurred while attempting to delete a temporary file used for the output (e.g., *rm*).
- Correction:** If the problem is with *rm*, the output will have been received successfully, but the temporary file will be left in the output directory; remove the temporary file.
- Error:** Received a negative response from the host while using an RJE workstation.
- Correction:** This indicates that the host reader is inactive. Contact the host personnel to activate the host reader (or you may be able to activate it using a console command). On some host systems, you may need to reset the RJE LU before you can continue.

It Was Working Just Fine . . . (Dealing with Subsequent Problems)

If you have problems after the product has been installed (the system has been working fine, then suddenly stops working), consider the following:

Have you installed new hardware which conflicts with the LINKix product's IRQ, I/O or shared memory address?

If you've added any adapter boards or other hardware since installing the adapter used by the LINKix product, does the IRQ, I/O address or other setting, conflict with those of the LINKix product? This is especially true when using boards that are not configurable (IRQ and/or I/O cannot be changed with jumpers).

Has something happened to the cabling?

Check to ensure that all cabling is secured tightly and that cables are not damaged.

Has something happened to the modem?

Is the modem operational? Check to ensure that it's plugged in and powered on.

Has something happened to the controller?

If you are using the DFT Coax product, is the 3x74/76 cluster controller operational? Check to ensure that it's plugged in and powered on.

Has something happened to the phone line?

Check to ensure that the phone line is operational. In the case of a leased-line, have your host administrator check the line for problems.

What is the current state of your physical unit (PU)?

Contact your host administrator to see if your PU is inoperative (INOP), pending contact (meaning it is trying to start, but as yet is unsuccessful) or active (meaning that the host thinks everything is OK).

Have changes occurred at the host which you are not aware of?

Check with the host administrator to see if any changes have occurred at the host which affect the current configuration settings. Make whatever changes are needed.

Troubleshooting

Has the communications adapter board failed?

This problem rarely occurs. Make sure you check out all other possible problems before concluding that the board has failed.

13. Advanced Diagnostics

This chapter contains detailed information that can be used to help resolve problems you may encounter when using LINKix 3270, LINKix 3770, APPC, CPI-C or LUA. The information is divided into the following sections:

- ◆ Diagnostics Support
- ◆ Tracing

Diagnostics Support

Provided with all of CLEO's products area number of diagnostics tools to help identify problems encountered during operation.

Trace Files

You can produce trace files which record all the activity in any software component. This tracing provides detailed diagnostic information by recording the flow of data within our software, or between our software and the remote system. You can trace any of the following:

- ◆ Information supplied to a CLEO API by transaction programs
- ◆ Information sent to and from the remote system
- ◆ Internal control messages sent between software components

Audit and Error Log Files

Log files can be used to help you solve configuration and operation errors. The following information is recorded in audit/error log files:

- ◆ Software, hardware or link failures
- ◆ Damaged or missing files
- ◆ Connectivity statistics
- ◆ Configuration problems
- ◆ Status information
- ◆ Information from application programs

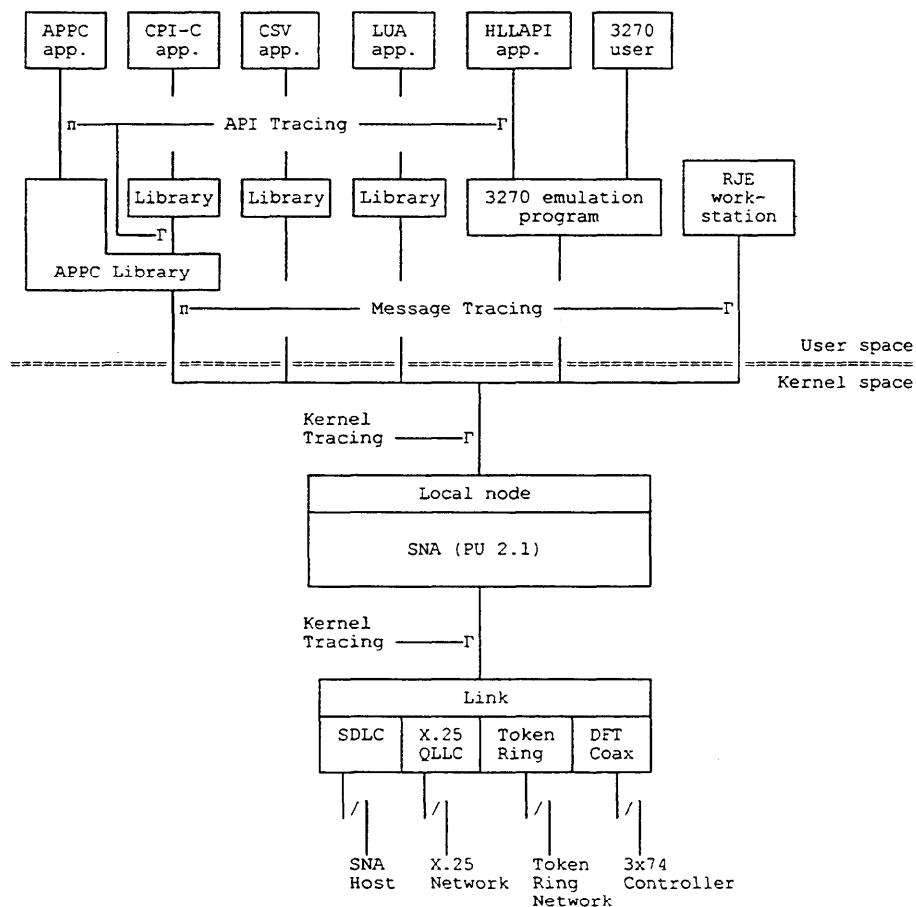
Two different types of log files are supported: an audit log file for general information and an error log file for serious warnings and fatal errors. For audit log information, you choose the detail of information you want to store; you may choose one of three different audit logging levels or disable audit logging altogether. You can also employ one log file to store both audit and error log information, instead of having separate files for each type.

All audit and error log messages have a given severity level. The two types of files (audit versus error) are identical in nature except for the severity level of messages:

Severity	Type	Description
6	audit	minor information
8	audit	significant event
10	audit	recoverable failure
12	error	error
16	error	fatal error

Tracing

The following diagram shows the overall structure of CLEO's LINKix products and the location of the tracing trace types: API tracing, message tracing and kernel tracing.



API Tracing

This trace type operates on application programs that use one of the programming interfaces: APPC, CPI-C, CSV, LUA or HLLAPI. This trace tracks all the parameters supplied to the API library by a transaction program and all the

parameters returned by the API library. Note that CPI-C API tracing includes both the CPI-C calls made by the application and the underlying APPC calls to which the CPI-C library converts them. For further details on tracing a specific API, refer to the ***Programmer's Guide*** for that interface.

Message Tracing

This trace type operates on the 3270 emulation program, RJE workstations and application programs that use the APPC, LUA or CSV APIs; HLLAPI application data can be traced by tracing the 3270 emulation program with which the HLLAPI application is being used. This trace captures all messages sent between the remote system and the LINKix 3270/LINKix 3770 product, as well as the control messages sent between software components. Options to suppress specific types of messages are available (for example, you can trace all HLLAPI data but not internal messages). See **Tracing User Space**, later in this chapter, for more information.

Kernel Tracing

This trace type operates on the tracingkernel space components (the local node and links). This trace captures the control messages sent between software components and messages sent between the remote system and local node. See **Tracing Kernel Space**, later in this chapter, for more information.

Tracing User Space

The tracing of the tracinguser space components is controlled with environment variables. These variables specify the file(s) to which the trace information will be stored, and they specify any limits or restrictions on the data that will be traced. If you're not familiar with environment variables and their use see the appropriate section of your UNIX system documentation.

SNATRC

This environment variable specifies the tracingHLLAPI applicationtrace file(s) that will be used for storing API trace data. It also indicates whether or not tracing is active when the application is started. Its syntax is:

```
file1 [ : [file2] [:]]
```

file1 is the name of the file to which API trace data will be written. A second file name, *file2*, is optional. If two file names are specified, you must use a colon to separate them. With two files specified, up to 250,000 bytes of trace data will be

Advanced Diagnostics

stored in the first file. When the first file is filled, trace information is written to the second file. When the second file contains 250,000 bytes, data is redirected to the first file, overwriting the previous 250,000 bytes of trace information.

Using two files, tracing can be active for an extended period of time and only 500,000 bytes of disk space will be used. If you specify only one file name, there is no limit on the size of the trace file.

Warning: Trace data must always be directed to one or more text files. *Do not specify the name of a device (such as /dev/tty), or a print spooler, as a trace filename.* Doing so may cause corruption of vital UNIX system files.

A second colon can be specified to indicate that tracing should be active when the application is started. If tracing is not active when the application is started, it can be activated from within an application with the DEFINE_TRACE Common Service Verb or the HLLAPI FunctionsSET SESSION PARAMETERS function. If you specify only one file name, two colons are required to make tracing active when the application is started.

Examples

Tracing to two files, inactive at start of 3270 emulation:

```
SNATRC=file1.trc:file2.trc
```

Tracing to one file, active at start of 3270 emulation:

```
SNATRC=file.trc::
```

API Tracing on Auto-Started Invocable TPs

Because you cannot set the SNATRC environment variable explicitly on an auto-started invocable TP, LINKix determines the tracing options by using the SNATRC setting that was used when starting the control daemon. The TP is started in the home directory of the owner of the executable file, so the trace files are created in this directory if a path is not specified.

SNAMSG

This environment variable specifies the tracing3270 emulationtrace file(s) that will be used for storing the data from message tracing (messages sent between the local LINKix software and the remote host system, as well as those sent between local

software components). It also indicates what types of messages are to be traced. Its syntax is:

```
file1 : [ file2 ] { : | :A | :P | :F }
```

The file(s) specified in this variable follow a similar format and operate in the same manner as with SNATRC.

Four parameters (of which one and only one must be specified) control which data is to be traced:

- : Trace all SNA data to and from the remote system, no internal control messages
- :A All messages are traced (SNA data and internal control messages)
- :P APPC data only, no internal control messages
- :F 3270, RJE, LUA and CSV data only, no internal control messages (CSV data applies only to the TRANSFER_MS_DATA verb, no other verbs involve remote message flow)

SNATRUNC

This environment variable lets you limit the number of bytes in each entry that's written to the trace file, and it applies to all trace types. By default, the length of each internal message is limited to a maximum of 256 bytes.

Set this variable to a decimal number, which represents the number of bytes to allow in each trace file entry (a typical size might be somewhere in the range of 512 to 1024). If a trace entry contains more characters than the value of SNATRUNC, the excess characters are truncated. For example, setting SNATRUNC to 512 limits tracing to 512 bytes of data per entry.

SNACTL

This environment variable can be used to suspend an application program's ability to control tracing. Normally all application programs can use the DEFINE_TRACE Common Service Verb, and HLLAPI applications can use the TRON/TROFF option of the function, to turn tracing on or off within the application. When SNACTL is set to any non-blank string, trace control commands issued by application programs are ignored. If tracing is on, it remains on; if tracing is off, it remains off. The return code from the HLLAPI SET SESSION PARAMETERS function still indicates successful completion of the function. To cancel SNACTL, set it to a blank character.

Controlling Tracing from within an Application

When creating an application with one of CLEO's APIs, you can control API tracing from within the. This allows you to trace a specific segment of a program, without having to trace the whole application. Tracing can be activated at the start of a segment (where a problem is being encountered) and deactivated at the end of it.

To use the trace facility within an API application, the SNATRC environment variable must be setup before the application program is started (see **SNATRC** earlier in this section). When setting this environment variable, you can cause the tracing facility to be active or inactive when an application is started; you can then activate or deactivate tracing from within the application as required.

To use this facility, the SNATRC environment variable must be set before the application program is started. Tracing is then controlled using one of the following:

- ◆ The Common Service Verb `DEFINE_TRACE` can be used to control tracing in applications that use the APPC, LUA, HLLAPI and CSV interfaces. See the *CSV Programmer's Guide* for complete information. (Note that HLLAPI tracing can also be controlled using the `SET SESSION PARAMETERS` function explained below.)
- ◆ The TRON/TROFF options of the HLLAPI Functions `SET SESSIONS PARAMETERS` function can be used to control tracing within HLLAPI applications. See the *HLLAPI Programmer's Guide* for complete information. (Note that HLLAPI tracing can also be controlled using the CSV `DEFINE_TRACE` verb explained above.)

Note that the SNACTL environment variable can override the ability to control tracing from within application programs. If SNACTL is set before the application program is started, any tracing control requests from within an application are ignored. If tracing is off and SNACTL is set to any non-blank string, you can prevent tracing in an application which normally uses it. In a similar manner, if tracing is on and SNACTL is set, you can force tracing on an entire application for which portions are normally turned off. In both cases, the API trace function still sends back a return code that indicates the trace request completed successfully.

Tracing Kernel Space

You can specify that tracing of kernel components (the local node and link) be activated when you start the component, or activate it later while the component is

running. You change the way tracing is handled by specifying command-line options when starting the control daemon:

```
comstart [-c configfile] [-t string] [-k] [-s]
```

See chapter 11, **Managing LINKix Components**, for more complete information on starting the daemon program.

Specifying Trace File Names

By default, trace data is stored to the two files *comfile1.trc* and *comfile2.trc* in the directory */usr/lib/linkix*. If the *-t* option is specified, the control daemon will send trace data to one or two specified trace files.

The *string* should contain:

```
file1 [: file2]
```

where *file1* and *file2* are the names of trace files. The second file name is optional; if it's used, a colon must separate the two file names.

When two files are used, trace information is directed to the first file until it reaches a size of 250,000 bytes. At that point, tracing continues with trace data being sent to the second file. When the second file also reaches 250,000 bytes, the first file is cleared and trace data is sent to the first file. This process continues, with each file being cleared and reused when the other file reaches a size of 250,000 bytes. Using two files means that tracing can be active for an extended period of time and will occupy no more than 500,000 bytes of disk space.

If you specify one file name only, the only limit on the size of the trace file that will be created is available disk space – use this option with caution.

Enabling Tracing at Initial Start

The *-k* option indicates that tracing on the SLIM (SNA LAN Interface Module) is active. The trace output is sent to the same files as for other service components. If this option is not specified, SLIM tracing is inactive. It can later be activated or deactivated for a specific component using the LINKix Manage.

Trace File Formats

The examples that follow show trace for the different trace types and notes to help you interpret the data. All trace types capture the following common data:

Process ID

For user space tracing, the five-digit process ID of the component being traced appears at the start of each line. Because kernel processes have no process ID, this field is replaced by one of the following:

KER.PU	indicates tracing on the local node
KER.SD	indicates tracing on SDLC links; this is replaced by <i>SDL.nn</i> when tracing messages specific to a particular link, where <i>nn</i> is an identifier for each link which is assigned when the link is activated
KER.X2	indicates tracing on X-25 links; this is replaced by <i>X25.nn</i> for a particular link (as for SDLC above)
KER.QL	indicates tracing on Vendor X.25 (QLLC) links; this is replaced by <i>QLL.nn</i> for a particular link (as for SDLC above)
KER.TR	indicates DLPI tracing on Token Ring links; this is replaced by <i>TOK.nn</i> for a particular link (as for SDLC above)
LL2.nn	indicates adapter interface tracing on a particular Token Ring link
USR.NM	indicates tracing on the network management daemon
USR.UX	indicates tracing on the UNIX Command Facility (UCF) daemon

The process ID is followed by an indicator of the type of component being traced (e.g. SDLC for an SDLC link or EHLL for a HLLAPI program).

Message Data

Message data is shown in hexadecimal and character formats. Each line shows a number of bytes of hexadecimal data followed by the equivalent ASCII or EBCDIC characters. Note that the translation is done character by character; that is, if the hex value represents a printable ASCII character, it is interpreted as ASCII, otherwise it is interpreted as EBCDIC. Some EBCDIC values which are also valid ASCII values may therefore be interpreted incorrectly.

APPC Trace File Format

For APPC, the verb control block (parameter block) supplied to the API library is traced when the verb is issued and again when it returns. (Note that the [MC_]RECEIVE_AND_POST verb, if issued successfully, returns twice; both returns are traced.)

The name of the APPC verb issued and the result on its return (taken from the primary return code) are shown at the top of each section of tracing. If data is being sent or returned, this is also traced following the parameter block, with the address at which the data is stored (taken from the parameter block).

```

297.22 APPC ----- 00:30:19.00
297.22 APPC MC_ALLOCATE request
297.22 APPC ---- Verb Parameter Block at address 40f3e8 ----
297.22 APPC 01000100 00000000 0000000F 00000040 . . . . .@>
297.22 APPC 61620000 00000000 00000000 00000000 <ab: . . . . .@>
297.22 APPC 00000000 424F554E 43454C55 D4D6C4C5 < BOUNCELUMODE>
297.22 APPC F1404040 C2D5C3C5 E3E3D740 40404040 <1000BNCETTP00000>
297.22 APPC 40404040 40404040 40404040 40404040 <ef464eee00w00001w>
297.22 APPC 40404040 40404040 40404040 40404040 <0000000000000000>
297.22 APPC 40404040 40404040 40404040 40404040 <0ee0www3ee>
297.22 APPC 40404040 00000000 00000000 00000000 <0000>
297.22 APPC 00000000 00000000 00000000 00000000 . . . . .@>
297.22 APPC 00000000 00000000 00000000 00000000 . . . . .@>
297.22 APPC 00000000 00000000 00000000 00000000 . . . . .@>
297.22 APPC 00000000
297.22 APPC ----- 00:30:20.00
297.22 APPC MC_ALLOCATE response, result = OK
297.22 APPC ---- Verb Parameter Block at address 40f3e8 ----
297.22 APPC 01000100 00000000 0000000F 00000040 . . . . .@!>
297.22 APPC 61620040 8F1C0100 00000000 00000000 <ab: . . . . .@>
297.22 APPC 00000000 424F554E 43454C55 D4D6C4C5 < BOUNCELUMODE>
297.22 APPC F1404040 C2D5C3C5 E3E3D740 40404040 <1000BNCETTP00000>
297.22 APPC 40404040 40404040 40404040 40404040 <0000000000000000>
297.22 APPC 40404040 40404040 40404040 40404040 <0000000000000000>
297.22 APPC 40404040 40404040 40404040 40404040 <0000000000000000>
297.22 APPC 40404040 00000000 00000000 00000000 <0000>
297.22 APPC 00000000 00000000 00000000 00000000 . . . . .@>
297.22 APPC 00000000 00000000 00000000 00000000 . . . . .@>
297.22 APPC 00000000 00000000 00000000 00000000 . . . . .@>
297.22 APPC 00000000
297.22 APPC ----- 00:30:24.00
297.22 APPC MC_FLUSH request
297.22 APPC ---- Verb Parameter Block at address 40f3e8 ----
297.22 APPC 06000100 00000000 0000000F 00000040 . . . . .@>
297.22 APPC 61620040 8F1C . . . . .@>
297.22 APPC ----- 00:30:24.00
297.22 APPC MC_FLUSH response, result = OK
297.22 APPC ---- Verb Parameter Block at address 40f3e8 ----
297.22 APPC 06000100 00000000 0000000F 00000040 . . . . .@>
297.22 APPC 61620040 8F1C . . . . .@>
297.22 APPC ----- 00:30:25.00
297.22 APPC Mc_SEND DATA request
297.22 APPC ---- Verb Parameter Block at address 40f3e8 ----
297.22 APPC 0F000100 00000000 0000000F 00000040 . . . . .@>
297.22 APPC 61620040 8F1C0000 02A30041 62E00000 <ab: . . . . .t.Ab: . . . . .@>
297.22 APPC ---- Data at address 4162e0 ----
297.22 APPC F1F1F1F1 4BF1F1F1 F14BF1F1 F1F14BF1 <llllK1lllK1lllK1>
297.22 APPC F1F1F14B F1F1F1F1 4BF1F1F1 F14BF1F1 <lllK1lllK1lllK1>
297.22 APPC F1F14BF1 F1F1F14B F1F1F1F1 4BF1F1F1 <llK1lllK1lllK1>
297.22 APPC F14BF1F1 F1F14BF1 F1F1F14B F1F1F1F1 <llK1lllK1lllK1>
297.22 APPC 4BF1F1F1 F14BF1F1 F1F14BF1 F1F1F14B <K1lllK1lllK1>
297.22 APPC F1F1F1F1 4BF1F1F1 F14BF1F1 F1F14BF1 <llllK1lllK1lllK1>

```


CPI-C Trace File Format

Because the LINKix implementation of CPI-C is implemented using APPC, the CPI-C library converts most CPI-C functions into APPC verb calls and CPI-C tracing includes both the CPI-C parameters and the APPC parameters. First the CPI-C request is traced, then the APPC request, then the APPC return and finally the CPI-C return. For other CPI-C functions that deal only with local information, such as checking or setting the receive type or the synchronization level, no APPC verbs are executed, so the tracing shows only the CPI-C parameters.

The name of the CPI-C call, or APPC verb issued, and the return code are shown at the top of each section of tracing. If data is being sent or returned, this is also traced following the parameter block, with the address at which the data is stored.

```

41.62 CPIC ----- Mar 12 1993 11:24:13.85
41.62 CPIC CMINIT request
41.62 CPIC   Sym dest name = WALL
41.62 APPC ----- Mar 12 1993 11:24:14.08
41.62 APPC TP.STARTED request
41.62 APPC ---- Verb Parameter Block at address 7ffffb60 ----
41.62 APPC 14000000 00000000 00000000 4C554350 < . . LUCP>
41.62 APPC 31202020 00000000 00000000 83819595 <1";~;;~..Cann~
41.62 APPC 9695A397 40404040 40404040 40404040 <ontp >
41.62 APPC 40404040 40404040 40404040 40404040 < >
41.62 APPC 40404040 40404040 40404040 40404040 < >
41.62 APPC 40404040 40404040 40404040
41.62 APPC ----- Mar 1; 1993 11:24:14.59
41.62 APPC TP_STARTED response, result = OK
41.62 APPC ---- Verb Parameter Block at address 7ffffb60 ----
41.62 APPC 14000000 00000000 00000000 4C554350 < . . LUCP>
41.62 APPC 31202020 0F000000 089D4000 83819595 <1,--:::-- -C->
41.62 APPC 9695A397 40404040 40404040 40404040 < ontp >
41.62 APPC 40404040 40404040 40404040 40404040 < >
41.62 APPC 40404040 40404040 40404040 40404040 < >
41.62 APPC 40404040 40404040 40404040
41.62 CPIC ----- Mar 1; 1993 11:24:14.67
41.62 CPIC CMINIT response, result = CM_OK
41.62 CPIC   Conversation ID = 01000000
41.62 CPIC   Conversation characteristics
41.62 CPIC   Conversation type = CM_MAPPED_CONVERSATION
41.62 CPIC   Deallocate type = CM_DEALLOCATE_SYNC_LEVEL
41.62 CPIC   Error direction = CM_RECEIVE_ERROR
41.62 CPIC   Sync level = CM_NONE
41.62 CPIC   Fill type = CM_FILL_LL
41.62 CPIC   Prepare to receive type = CM_PREP_TO_RECEIVE_SYNC_LEVEL
41.62 CPIC   Receive type = CM_RECEIVE_AND_WAIT
41.62 CPIC   Send t-e = CM_BUFFER_DATA
41.62 CPIC   Conversation security type = XC_SECURITY_NONE
41.62 CPIC   Log data pointer = 0X0000
41.62 CPIC   Log data length = 0
41.62 CPIC   Sym dest name = WALL
41.62 CPIC   Partner LU name =
41.62 CPIC D8E6C5D9 E3E8E4C9 4BC1E2C4 C6C7C8D1 <QWERTYUIKASDFGHJ>
41.62 CPIC D2 <K >
41.62 CPIC   Mode name =
41.62 CPIC D4D6C4C5 F1404040 <MODE1 >
41.62 CPIC   Partner TP name =
41.62 CPIC E6C1D3D3 E3D74040 40404040 40404040 <wALLTP >
41.62 CPIC 40404040 40404040 40404040 40404040 < >
41.62 CPIC 40404040 40404040 40404040 40404040 < >
41.62 CPIC 40404040 40404040 40404040 40404040
41.62 CPIC ----- Mar 1; 1993 11:24:14.76
41.62 CPIC CMECS request

```

Advanced Diagnostics

```
41.62 CPIC      Conversation ID = 01000000
41.62 CPIC      ----- Mar 12 1993 11:24:14.77
41.62 CPIC      CMECS response, result = CM_OK
41.62 CPIC      Conversation state = CM_INITIALIZE_STATE
41.62 CPIC      ----- Mar 12 1993 11:24:14.78
41.62 CPIC      CMSCT request
41.62 CPIC      Conversation ID = 01000000
41.62 CPIC      Conversation type = CM_BASIC_CONVERSATION
41.62 CPIC      Conversation characteristics
41.62 CPIC      Conversation type = CM_MAPPED_CONVERSATION
41.62 CPIC      ----- Mar 12 1993 11:24:14.91
41.62 CPIC      CMSCT response, result = CM_OK
41.62 CPIC      Conversation characteristics
41.62 CPIC      Conversation type = CM_BASIC_CONVERSATION
41.62 CPIC      ----- Mar 12 1993 11:24:14.93
41.62 CPIC      CMSED request
41.62 CPIC      Conversation ID = 01000000
41.62 CPIC      Error direction = CM_SEND_ERROR
41.62 CPIC      Conversation characteristics
41.62 CPIC      Error direction = CM_RECEIVE_ERROR
41.62 CPIC      ----- Mar 12 1993 11:24:15.00
41.62 CPIC      CMSED response, result = CM_OK
41.62 CPIC      Conversation characteristics
41.62 CPIC      Error direction = CM_SEND_ERROR
41.62 CPIC      ----- Mar 12 1993 11:24:15.24
41.62 CPIC      CMSSL request
41.62 CPIC      Conversation ID = 01000000
41.62 CPIC      Sync level = CM_NONE
41.62 CPIC      Conversation characteristics
41.62 CPIC      Sync level = CILNONE
41.62 CPIC      ----- Mar 12 1993 11:24:15.29
41.62 CPIC      CMSSL response, result = CM_OK
41.62 CPIC      Conversation characteristics
41.62 CPIC      Sync level = CM_NONE
41.62 CPIC      ----- Mar 12 1993 11:24:15.30
41.62 CPIC      XCMESI request
41.62 CPIC      Entry number = 0
41.62 CPIC      Sym dest name = WALL
41.62 CPIC      Entry length = 124
41.62 CPIC      ----- Mar 12 1993 11:24:15.53
41.62 CPIC      XCMESI response, result = CM_OK
41.62 CPIC      Sym dest name = WALL
41.62 CPIC      Partner LU name =
41.62 CPIC      51574552 54595549 2E415344 4647484A <QWERTYUI. ASDFGHJ>
41.62 CPIC      4B <K >
41.62 CPIC      TP name type = XC_APPLICATION_TP
41.62 CPIC      Partner TP name =
41.62 CPIC      57414C4C 54502020 20202020 20202020 <WALLTP >
41.62 CPIC      20202020 20202020 20202020 20202020 < >
41.62 CPIC      20202020 20202020 20202020 20202020 < >
41.62 CPIC      20202020 20202020 20202020 20202020 < >
41.62 CPIC      Mode name =
41.62 CPIC      4D4F4445 31202020 <MODE1 >
41.62 CPIC      Conversation security type = XC_SECURITY_NONE
41.62 CPIC      Security user ID =
41.62 CPIC      00000000 00000000
41.62 CPIC      ----- Mar 12 1993 11:24:15.67
41.62 CPIC      XCMSSI request
41.62 CPIC      Conversation ID = 00000000
41.62 CPIC      Sym dest name = WALLNAME
41.62 CPIC      Partner LU name =
41.62 CPIC      51574552 54595549 2E415344 4647484A cQWERTYUI .ASDFGHJ>
41.62 CPIC      4B <K >
41.62 CPIC      TP name type = XC_APPLICATION_TP
41.62 CPIC      Partner TP name =
41.62 CPIC      57414C4C 54502020 20202020 20202020 <WALLTP >
41.62 CPIC      20202020 20202020 20202020 20202020 < >
41.62 CPIC      20202020 20202020 20202020 20202020 < >
41.62 CPIC      20202020 20202020 20202020 20202020 < >
41.62 CPIC      Mode name =
41.62 CPIC      4D4F4445 31202020 <MODE1 >
41.62 CPIC      Conversation security type = XC_SECURITY_NONE
```

Advanced Diagnostics

```

41.62 CPIC Security user ID =
41.62 CPIC 00000000 00000000 <.....>
41.62 CPIC Entry length = 124
41.62 CPIC ----- Mar 12 1993 11:24:15.86
41.62 CPIC XCMSSI response, result = CM_OK
41.62 CPIC ----- Mar 12 1993 11:24:15.96
41.62 CPIC CMALLC request
41.62 CPIC Conversation ID = 01000000
41.62 CPIC Conversation characteristics
41.62 CPIC Conversation type = CM_BASIC CONVERSATION
41.62 CPIC Return control = CM_WHEN_SESSION_ALLOCATED
41.62 CPIC Sync level = CM.NONE
41.62 CPIC Conversation security type = XC_SECURITY_NONE
41.62 CPIC Partner LU name =
41.62 CPIC D8E6C5D9 E3E8E4C9 4BC1E2C4 C6C7C8D1 <QWERTYUI KASDFGHJ>
41.62 CPIC D2 <K>
41.62 CPIC Mode name =
41.62 CPIC D4D6C4C5 F1404040 <MODE1>
41.62 CPIC Partner TP name =
41.62 CPIC E6C1D3D3 E3D74040 40404040 40404040 <WALLTP>
41.62 CPIC 40404040 40404040 40404040 40404040 <
41.62 CPIC 40404040 40404040 40404040 40404040 <
41.62 CPIC 40404040 40404040 40404040 40404040
41.62 APPC ----- Mar 1; 1993 11:24:15.98
41.62 APPC ALLOCATE request
41.62 APPC ---- Verb Parameter Block at address 7ffffb70 ----
41.62 APPC 01000000 00000000 00000000 0F000000 <
41.62 APPC 089D4000 00000000 00000000 00000000 <S-:::.....>
41.62 APPC 00000000 00000000 00000000 0000D4D6 <.....MC>
41.62 APPC C4C5F140 4040E6C1 D3D3E3D7 40404040 <DE1 WALLTP>
41.62 APPC 40404040 40404040 40404040 40404040 <
41.62 APPC 40404040 40404040 40404040 40404040 <
41.62 APPC 40404040 40404040 40404040 40404040 <
41.62 APPC 40404040 40400000 00000000 00000000 <.....>
41.62 APPC 00000000 00000000 00000000 00000000 <
41.62 APPC 00000000 00000000 00000000 00D8E6C5 <::: :::: ::::QWE
41.62 APPC D9E3E8E4 C94BC1E2 C4C6C7C8 D1D20000 <RTYUIKASDFGHJK. >
41.62 APPC 00000000 00000000
41.62 APPC ----- Mar 12 1993 11:24:16.75
41.62 APPC ALLOCATE response, result = OK
41.62 APPC ---- Verb Parameter Block at address 7ffffb70 ----
41.62 APPC 01000000 00000000 00000000 0F000000 <.....>
41.62 APPC 089D4000 BCCE4000 00000000 00000000 <.....>
41.62 APPC 00000000 00000000 00000000 0000D4D6 <.....MO>
41.62 APPC C4C5F140 4040E6C1 D3D3E3D7 40404040 <DE1 WALLTP>
41.62 APPC 40404040 40404040 40404040 40404040 <
41.62 APPC 40404040 40404040 40404040 40404040 <
41.62 APPC 40404040 40404040 40404040 40404040 <
41.62 APPC 40404040 40400000 00000000 00000000 <.....>
41.62 APPC 00000000 00000000 00000000 00000000 <
41.62 APPC 00000000 00000000 00000000 00D8E6C5 <.....QWE>
41.62 APPC D9E3E8E4 C94BC1E2 C4C6C7C8 D1D20000 <R&L&DFGHJK. >
41.62 APPC 00000000 00000000
41.62 CPIC ----- Mar 12;1993; 11:24:16.77
41.62 CPIC CMALLC response, result = CM_OK
41.62 CPIC ----- Mar 12 1993 11:24:16.78
41.62 CPIC CMECS request
41.62 CPIC Conversation ID = 01000000
41.62 APPC ----- Mar 12 1993 11:24:16.81
41.62 APPC GET_STATE request
41.62 APPC ---- Verb Parameter Block at address 7ffffb7c ----
41.62 APPC 09000000 00000000 00000000 0F000000 <.....>
41.62 APPC 089D4000 BCCE4000 00000000 <
41.62 APPC ----- Mar 12 1993 11:24:16.81
41.62 APPC GET_STATE response, result = OK
41.62 APPC ---- Verb Parameter Block at address 7ffffb7c ----
41.62 APPC 09000000 00000000 00000000 0F000000 <.....>
41.62 APPC 089D4000 BCCE4000 02000000
41.62 CPIC ----- Mar 12 1993 11:24:16.82
41.62 CPIC CMECS response, result = CM_OK
41.62 CPIC Conversation state = CM_SEND_STATE
41.62 CPIC ----- Mar 12 1993 11:24:16.89

```

Advanced Diagnostics

```
41.62 CPIC      CMSEND request
41.62 CPIC      Conversation ID = 01000000
41.62 CPIC      Buffer empty
41.62 CPIC      Send length = 0
41.62 CPIC      Conversation characteristics
41.62 CPIC      Send type = CM_BUFFER_DATA
41.62 CPIC      Prepare to receive type = CM_PREP_TO_RECEIVE_SYNC_LEVEL
41.62 CPIC      Deallocate type = CM_DEALLOCATE_SYNC_LEVEL
41.62 APPC      ----- Mar 12 1993 11:24:16.90
41.62 APPC      SEND_DATA request
41.62 APPC      ---- Verb Parameter Block at address 7ffffb64 ----
41.62 APPC      0F000000 00000000 00000000 0F000000 . . . . < . . . . .>
41.62 APPC      089D4000 BCCE4000 00000000 E8B14100 . . . . < . . . . .Y.A.>
41.62 APPC      00000000
41.62 APPC      ----- Mar 12 1993 11:24:16.94
41.62 APPC      SEND_DATA response, result = OK
41.62 APPC      ---- Verb Parameter Block at address 7ffffb64 ----
41.62 APPC      0F000000 00000000 00000000 0F000000 . . . . < . . . . .>
41.62 APPC      089D4000 BCCE4000 00000000 E8B14100 . . . . < . . . . .Y.A.>
41.62 APPC      00000000
41.62 CPIC      ----- Mar 12 1993 11:24:16.95
41.62 CPIC      CMSEND response, result = CM_OK
41.62 CPIC      Request to send received = CM_REQTO_SEND_NOT_RECEIVED
```

LUA and CSV Trace File Format

For LUA, the verb control block (parameter block) supplied to the API library is traced when the verb is issued. It is then traced when the verb returns and, if it returns IN_PROGRESS to indicate asynchronous completion, it's traced again when it completes.

The name of the LUA verb issued and the result on its return (taken from the primary and secondary return codes) are shown at the top of each section outracing. If data is being sent or returned, this is also traced following the parameter block, with the address at which the data is stored (taken from the parameter block).

CSV tracing is similar to both APPC and LUA, except that there are no asynchronous verb returns; the verb control block (parameter block) supplied to the API library is traced twice, when the verb is issued and when it returns.

The following is an example of LUA API tracing, which also includes CSV API tracing for the CONVERT verb.

```
33.93 LUA      ----- - - - - - 17:41:05.00
33.93 LUA      RUI_INIT request
33.93 LUA      ---- Verb Parameter Block at address 41658c ----
33.93 LUA      520000CE 00000000 00000180 00000000 <R . . . . .>
33.93 LUA      4C49534F 4C484920 00000000 00000000 <LISOLHI . . . . .>
33.93 LUA      00000000 00000000 0000262A 00000000 . . . . < . . . . .&* . . . .>
33.93 LUA      00000000 00000000 00000000 00000000 . . . . < . . . . .>
33.93 LUA      00000000 00000000 00000000 00000000 . . . . < . . . . .>
33.93 LUA      00000000 00000000 00000000 00000000 . . . . < . . . . .>
33.93 LUA      00000000 00000000 00000000 00000000 . . . . < . . . . .>
33.93 LUA      00000000 00000000 00000000 00000000 . . . . < . . . . .>
33.93 LUA      00000000 00000000 00000000 00000000 . . . . < . . . . .>
33.93 LUA      00000000 00000000 00000000 00000000 . . . . < . . . . .>
```

Administration Guide
13-17

13-18

Advanced Diagnostics

```
150.95 EHLL -----19:37:57.00
150.95 EHLL RESET SYSTEM request (21), length 0, PS position 0
150.95 EHLL -----19:37:57.00
150.95 EHLL RESET SYSTEM response, length 0, return code 0
150.95 EHLL -----19:37:57.00
150.95 EHLL QUERY SESSIONS request (10), length 64, PS Position 0
150.95 EHLL -----19:37:57.00
150.95 EHLL QUERY SESSIONS response, length 2, return code 2
150.95 EHLL ---- Data at address 4092a0 ----
150.95 EHLL 31312020 20202020 2050D007 45534553 <11 P. .ESES>
150.95 EHLL 53303120 20480000 cS01 H..
150.95 EHLL -----19:37:57.00
150.95 EHLL CONNECT PSPACE request (1), length 1, PS position 0
150.95 EHLL ---- Data at address 4092a0 ----
150.95 EHLL 45 <E
150.95 EHLL -----19:37:58.00
150.95 EHLL CONNECT PSPACE response(1), length 1, return code 0
150.95 EHLL ---- Data at address 4092a0 ----
150.95 EHLL 45 <E
150.95 EHLL -----19:37:58.00
150.95 EHLL SET SESSION PARAMETERS request (9), length 30, PS position 0
150.95 EHLL ---- Data at address 4092a0 ----
150.95 EHLL 53524348 46524F4D 20535243 48465257 <SRCHFROM SRCHFRW>
150.95 EHLL 44205354 524C454E 20454F54 3D24 <D STRLEN EOT=$ >
150.95 EHLL -----19:37:58.00
150.95 EHLL SET SESSION PARAMETERS response(9), length 30, return code 0
150.95 EHLL -----19:37:58.00
150.95 EHLL WAIT request (4), length 0, PS position 0
150.95 EHLL -----19:37:59.00
150.95 EHLL WAIT response(4), length 0, return code 0
150.95 EHLL -----19:37:59.00
150.95 EHLL SEND KEY request (3), length 42, PS position 0
150.95 EHLL ---- Data at address 4092a0 ----
150.95 EHLL 6C6F676F 6E206170 706C6964 3D74736F clogon applid=tso>
150.95 EHLL 2C6C6F67 6D6F6465 3D633234 2C646174 <, logmOde=c24 ,dat>
150.95 EHLL 613D7864 636C3034 4045 <a=xdc104@E
150.95 EHLL -----19:38:08.00
150.95 EHLL SEND KEY response(3), length 42, return code 0
150.95 EHLL -----19:38:08.00
150.95 EHLL WAIT request (4), length 0, PS position 0
150.95 EHLL -----19:38:11.00
150.95 EHLL WAIT response(4), length 0, return code 0
150.95 EHLL -----19:38:11.00
150.95 EHLL SEARCH FIELD request (30), length 8, PS position 23
150.95 EHLL ---- Data at address 4092a0 ----
150.95 EHLL 50415353 574F5244 < PASSWORD
150.95 EHLL -----19:38:11.00
150.95 EHLL SEARCH FIELD response, length 0, return code 24
150.95 EHLL -----19:38:13.00
```

Kernel Trace File Formats

The start of each kernel trace entry shows the type of message being traced and its direction; some of the header information is also decoded. Any header information is then shown separately, followed by the message data.

Kernel Trace Abbreviations

+RSP positive response
-RSP negative response

BB, BBI	begin bracket indicator
BC, BCI	begin chain indicator
DAF	destination address field
CEB	conditional end bracket
DLC	data link control
DR1, DR2	definite response 1 and 2 indicators
EB, EBI	end bracket indicator
EC, ECI	end chain indicator
ER, ERI	exception response indicator
OAF	origin address field
ODAI	OAF/DAF assignor indicator
RH	request header or response header
RU	request unit or response unit
SC	session control
TH	transmission header

SNA Message Trace

In the SNA message trace, the first two lines of each entry show if the message flow is from the LINKix software (>>SEND>>), to the LINKix software (<<RCVE<<) or in a local APPC conversation (>>LOOP<<). Some of the decoded bits from the Transmission Header (TH) and Request/Response Header (RH) are also displayed here. The TH and RH are then shown separately, followed by the message data.

```

KER . SD SNA -----13:44:46.00
KER . SD SNA >> SEND>> DAF:01 OAF:01 ODAI:0FF Expedited
KER . SD SNA BIND +RSP SC FI BC EC DR1
KER . SD SNA
KER . SD SNA TH: 2d0001010000
KER . SD SNA RH: eb8000
KER . SD SNA
KER . SD SNA ---- Data ----
KER . SD SNA 31001307 B0B050B1 04048787 84040602 C1 . . . . P. .ggd. .>
KER . SD SNA 00000000 00000000 20000000 1F000902 < . . . . .>
KER . SD SNA 0306c304 D6C4C540 09030000 00000000 <LOCOMODE@ . . . .>
KER . SD SNA 00010905 E3D7D3E4 F2404040 00 <....TPLu2@@@.
KER . SD SNA -----13:44:46.00
KER . SD SNA <<RCVE<< DAF:01 OAF:01 ODAI:0FF
KER . SD SNA RQE FMD FI BC EC DR1 PI CD
KER . SD SNA
KER . SD SNA TH: 2C0001010001
KER . SD SNA RH: 0b9120
KER . SD SNA
KER . SD SNA ---- Data ----
KER . SD SNA 120502FF 0003D100 0007E3D7 D5C1D4C5 . < . . . .J. .TPM>
KER . SD SNA F2000013 12FF2E2E 2F73312F 73317069 <2 . . . . . /sl/slpi>
KER . SD SNA 6D696E2E 63 <rein. c >

```

Internal Message Trace

This type of tracing contains the messages sent between software components, in the internal format used by the LINKix product. Internal message tracing is provided mainly for use by CLEO Technical Support personnel, who may ask you to provide it to assist them in diagnosing problems. You'll generally find SNA message tracing is more useful for your own problem diagnosis and for investigating problems with remote systems.

```

KER . PU DLC                                     13:44:44.00
KER . PU DLC 02160010->01110021  DLC DATA
KER . PU DLC      DAF:01 OAF:01 Expedited
KER . PU DLC      BIND   RQD SC   FI BC EC DR1
KER . PU DLC
KER . PU DLC ---- Header at address 406a4c, 1 elements ----
KER . PU DLC 00000000 000C001l 2D000101 00000000 . . . .<. . . - . . . . .>
KER . PU DLC 0000                                <..
KER . PU DLC
KER . PU DLC ---- Element at address 406a73, start 10, end 90 ----
KER . PU DLC 6B800031 001307B0 B050B104 04878784 <k..l. . . .p.. . ggd>
KER . PU DLC 04060200 00000000 00000020 000008E3 <
KER . PU DLC D7D3E4F1 4040401F 000902D3 D6C3D4D6 <G-Gib&I :: :L&&
KER . PU DLC C4C54009 03000000 00000000 010904E3 <0E@
KER . PU DLC D7D3E4F1 40404000 08E3D7D3 E4F24040 <PLuL@@@. .TPLu2W!>
KER . PU DLC 40 ~@
KER . PU DLC -----13:44:44.00
KER . PU DLC 01110021->02160010  DLC DATA
KER . PU DLC      DAF:01 OAF:01 Expedited
KER . PU DLC      BIND   +RSP SC   FI BC EC DR1
KER . PU DLC
KER . PU DLC ---- Header at address 406a4c, 1 elements ----
KER . PU DLC 00000506 00010000 2D000101 00000000 . . . .<. . . - . . . . .>
KER . PU DLC 0000                                <..
KER . PU DLC
KER . PU DLC ---- Element at address 406a73, start 10, end 73 ----
KER . PU DLC EB800031 001307B0 B050B104 04878784 . . l<. . . . .P.. . ggd>
KER . PU DLC 04060200 00000000 00000020 0000001F . . . .<. . . . .>
KER . PU DLC 000902D3 D6C3D4D6 C4C54009 03000000 .<..LOCMODE@. . . :>
KER . PU DLC 00000000 010905E3 D7D3E4F2 40404000 <....TPLu2@@@@.>
KER . PU DLC -----13:44:46.00
KER . PU DLC 02160010->01110021  DLC DATA
KER . PU DLC      DAF:01 OAF:01 Normal
KER . PU DLC      RQE   FMD FI BC EC DR1 PI CD
KER . PU DLC
KER . PU DLC ---- Header at address 406a4c, 1 elements ----
KER . PU DLC 00000101 00000000 2C000101 00010000 . . . .<. . . . .>
KER . PU DLC 0000                                <..
KER . PU DLC
KER . PU DLC ---- Element at address 406a73, start 10, end 49 ----
KER . PU DLC 0B912012 0502FF00 03D10000 07E3D7D5 <-j----- .J-. .TpN>
KER . PU DLC C1D4C5F2 00001312 FF2E2E2F 73312F73 <AME2 . . . . .s/s>
KER . PU DLC 3170960D 696E2E63 <lpimin.c

```

Token Ring

The example below includes both types of Token Ring tracing: DLPI tracing and adapter interface tracing. (Adapter interface tracing is identified as LLI.) In both cases, the first line of each trace entry indicates the message type, and is followed by the message data.

```

=====
===== Initialized Fri Mar  5 12:51:33 1993 =====
=====
KER . TR DLPI ----- Mar  5 1993 12:52:14:04
KER . TR DLPI OPEN STREAM
TOK .01 DLPI ----- Mar  5 1993 12:52:19:03
TOK .01 DLPI TX M_PCPROTO INFO_REQ
TOK .01 DLPI 00000000
TOK .01 DLPI ----- Mar  5 1993 12:52:19:00
TOK .01 DLPI TX M_PROTO TRC_SET_LINK
TOK .01 DLPI 01010000 01000000 54524C20 20202020 . . < . . .-TRL >
TOK .01 DLPI 00000000 03000000 06000000
TOK .01 DLPI ----- Mar  5 1993 12:52:19:00
TOK .01 DLPI RX M_PCPROTO INFO_ACK
TOK .01 DLPI 03000000 00100000 00000000 00000000 . . . < . . .-...->
TOK .01 DLPI 02000000 00000000 04000000 00000000 . . . < . . .-...->
TOK .01 DLPI 01000000 00000000 00000000 00000000 . . . < . . .-...->
TOK .01 DLPI 00000000 01050000 00000000 02000000 . . . < . . .-...->
TOK .01 DLPI 00000000 00000000 00000000
TOK .01 DLPI ----- Mar  5 1993 12:52:19:05
TOK .01 DLPI TX M_PROTO ATTACH_REQ
TOK .01 DLPI 0B000000 47000000
TOK .01 DLPI ----- Mar  5 1993 12:52:19:06
TOK .01 DLPI RX M_PCPROTO OK_ACK
TOK .01 DLPI 06000000 0B000000
TOK .01 DLPI ----- Mar  5 1993 12:52:19:07
TOK .01 DLPI TX M_PCPROTO INFO_REQ
TOK .01 DLPI 00000000
TOK .01 DLPI ----- Mar  5 1993 12:52:19:07
LL2.01 LLI TX M_PCPROTO INT?O_REQ
LL2.01 LLI 00000000
LL2.01 LLI ----- Mar  5 1993 12:52:19:08
LL2.01 LLI RX M_PCPROTO INFO_ACK
LL2.01 LLI 03000000 D0070000 00000000 06000000 . . . < . . .-...->
LL2.01 LLI 02000000 03000000 00000000 B6B7FFFF . . . < . . .-...->
LL2.01 LLI FFFFFFFF 00000000 00160141 00000000 . . . < . . .-...->
LL2.01 LLI 00000000 00000000 00000000 00000000
TOK .01 DLPI ----- Mar  5 1993 12:52:19:09
TOK .01 DLPI RX M_PCPROTO INFO_ACK
TOK .01 DLPI 03000000 D0070000 00000000 00000000 . . . < . . .-...->
TOK .01 DLPI 02000000 00000000 00000000 00000000 . . . < . . .-...->
TOK .01 DLPI 01000000 00000000 00000000 00000000 . . . < . . .-...->
TOK .01 DLPI 00000000 01050000 00000000 02000000 . . . < . . .-...->
TOK .01 DLPI 00000000 00000000 00000000
TOK .01 DLPI ----- Mar  5 1993 12:52:19:10
TOK .01 DLPI TX M_PROTO BIND_REQ
TOK .01 DLPI 01000000 04000000 01000000 01000000 . . . < . . .-...-='
TOK .01 DLPI 00000000
TOK .01 DLPI ----- Mar  5 1993 12:52:19:11
LL2.01 LLI TX M_PROTO BIND_REQ

```

Advanced Diagnostics

```

LL2.01 LLI 01000000 04000000 47000000 0200FF7F .....G.....>
LL2.01 LLI ----- Mar 05 1993 12:52:19:12
LL2.01 LLI RX M_PCPROTO BIND_ACK
LL2.01 LLI
LL2.01 LLI 04000000 04000000 06000000 18000000 <.....>
LL2.01 LLI 00000000 00000000 10005AFF 29B0 ..Z.)..>
TOK.01 DLPI ----- Mar 05 1993 12:52:19:13
TOK.01 DLPI RX M_PCPROTO BIND_ACK
TOK.01 DLPI
TOK.01 DLPI 04000000 04000000 07000000 18000000 <.....>
TOK.01 DLPI 01000000 00000000 10005AFF 29B004 ..2.)..>
KER.TR DLPI ----- Mar 05 1993 12:52:26:93
KER.TR DLPI OPEM STREAM
TOK.01 DLPI ----- Mar 05 1993 12:52:26:95
TOK.01 DLPI TX M_PCPROTO INFO_REQ
TOK.01 DLPI
TOK.01 DLPI 00000000
TOK.01 DLPI ----- Mar 05 1993 12:52:26:96
TOK.01 DLPI TX M_PROTO TRC_SET_LINK
TOK.01 DLPI
TOK.01 DLPI 01010000 01000000 54524C20 20202020 <.....TRL>
TOK.01 DLPI 00000000 03000000 06000000
TOK.01 DLPI ----- Mar 05 1993 12:52:26:97
TOK.01 DLPI RX M_PCPROTO INFO_ACK
TOK.01 DLPI
TOK.01 DLPI 03000000 00100000 00000000 00000000 <.....>
TOK.01 DLPI 02000000 00000000 04000000 00000000 <.....>
TOK.01 DLPI 01000000 00000000 00000000 00000000 <.....>
TOK.01 DLPI 00000000 01050000 00000000 02000000 <.....>
TOK.01 DLPI 00000000 00000000 00000000
TOK.01 DLPI ----- Mar 05 1993 12:52:26:98
TOK.01 DLPI TX M_PROTO ATTACH_REQ
TOK.01 DLPI
TOK.01 DLPI 0B000000 47000000
TOK.01 DLPI ----- Mar 05 1993 12:52:26:98
TOK.01 DLPI RX M_PCPROTO OK_ACK
TOK.01 DLPI
TOK.01 DLPI 06000000 0B000000
TOK.01 DLPI ----- Mar 05 1993 12:52:26:99
TOK.01 DLPI TX M_PCPROTO INFO_REQ
TOK.01 DLPI
TOK.01 DLPI 00000000
TOK.01 DLPI ----- Mar 05 1993 12:52:27:00
TOK.01 DLPI RX M_PCPROTO INFO_ACK
TOK.01 DLPI
TOK.01 DLPI 03000000 D0070000 00000000 00000000 <.....>
TOK.01 DLPI 02000000 00000000 00000000 00000000 <.....>
TOK.01 DLPI 01000000 00000000 00000000 00000000 <.....>
TOK.01 DLPI 00000000 01050000 00000000 02000000 <.....>
TOK.01 DLPI 00000000 00000000 00000000
TOK.01 DLPI ----- Mar 05 1993 12:52:27:01
TOK.01 DLPI TX M_PROTO BIND_REQ
TOK.01 DLPI
TOK.01 DLPI 01000000 04000000 00000000 01000000 <.....>
TOK.01 DLPI 00000000
TOK.01 DLPI ----- Mar 05 1993 12:52:27:02
TOK.01 DLPI RX M_PCPROTO BIND_ACK
TOK.01 DLPI
TOK.01 DLPI 04000000 04000000 07000000 18000000 <.....>
TOK.01 DLPI 00000000 00000000 10005AZ'F 29B004 ..Z.)..>
TOK.01 DLPI ----- Mar 05 1993 12:52:27:03
TOK.01 DLPI TX M_PROTO SET_ALL_REQ
TOK.01 DLPI
TOK.01 DLPI F0000000 02000000 0A000000 01000000 <.....>
TOK.01 DLPI 03000000
TOK.01 DLPI ----- Mar 05 1993 12:52:27:04
TOK.01 DLPI RX M_PCPROTO OK_ACK
TOK.01 DLPI
TOK.01 DLPI 06000000 F0000000
LL2.01 LLI ----- Mar 05 1993 12:52:37:71
LL2.01 LLI RX M_PROTO and 1 M_DATA UNITDATA_IMD

```

```

LL2.01 LLI 08000000 06000000 18000000 06000000 . . . . . < . . . . . >
LL2.01 LLI 1E000000 00000000 08005A14 06191000 < . . . . . Z . . . . . >
LL2.01 LLI 5AFF29B0 <Z.) . . . . . >
LL2.01 LLI
LL2.01 LLI 0404BF <...
KER.TR DLPI ----- Mar 05 1993 12:52:37:73
KER.TR DLPI RX M_PROTO XID_IND
KER.TR DLPI
KER.TR DLPI 2A000000 01000000 07000000 18000000 . . . . . >
KER.TR DLPI 07000000 1F000000 10005AFF 29B00408 . . . . . < . . . . . Z.) . . . >
KER.TR DLPI 005A1406 1904 <Z . . . . . >
TOK.01 DLPI ----- Mar 05 1993 12:52:37:81
TOK.01 DLPI TX M_PROTO and 1 M_DATA XID.RES
TOK.01 DLPI
TOK.01 DLPI 2B000000 01000000 07000000 10000000 <+ . . . . . >
TOK.01 DLPI 08005A14 061904 < . . Z . . . . . >
TOK.01 DLPI
TOK.01 DLPI 325D05DF FFFF0000 B0880000 00000000 <2] . . . . . h . . . . . >
TOK.01 DLPI 00010B40 00078900 00000007 00103B00 . . . . . <1 . . . . . >
TOK.01 DLPI 27110C08 04F0F3F0 F0F0F013 06E2D5C1 . . . . . <* . . . . 030000 . . SNA>
TOK.01 DLPI D760C9E7 40C3E240 40404040 40400908 <P'IX CS . . . . . >
TOK.01 DLPI F0F0F0F0 F0F0F013 11031000 10F0F0F0 <0000000 . . . . . 000>
TOK.01 DLPI F0F0F0F0 F0F0F0F0 F0F00E01 F4 <0000000000 . . 4 >

```


Coptool

coptool is a utility provided with the coprocessor version of LINKix and is installed in the `/usr/lib/linkix` directory. Its primary functions are to download the operational environment to the board and manage the coprocessor resources.

When the *comstart* script is executed, *coptool* is started and it downloads the onboard code to the board in your system. (Under AIX, *coptool* is executed when the *comload* script is executed.)

If problems arise, *coptool* can be used to examine the onboard environment. This secondary function allows CLEO technicians to diagnose link problems and determine the cause.

coptool can be run in either interactive or command line mode. If *coptool* is invoked with a command line argument, it will execute that command and return to the UNIX prompt. If it's invoked without a command line argument, *coptool* enters interactive mode and prompts for a command. It continues to prompt for a command until you EXIT or QUIT.

The following is a summary of *coptool* commands. Many of the commands maybe abbreviated.

AUDIT	Displays an audit buffer. Maybe abbreviated "A".
BOARD	Displays configuration details for the installed board.
BOOT	Loads and runs the bootstrap.
DUMP	Displays some of the coprocessor memory in hex. Maybe abbreviated "D".
EDIT	Modify coprocessor memory. May be abbreviated "E".
HELP	Use help to get information on available commands. This command can be abbreviated "H" or "?".
LOAD	Loads the specified program onto the coprocessor.

MEM	Displays the coprocessor Memory Control Block list.
PRINT	Displays some of the coprocessor memory in text. Maybe abbreviated "P".
QUIT or EXIT	Exit the program. Can be abbreviated "Q".
RESTART	Resets the bootstrap program to restart the board.
RUN	Loads the specified program onto the coprocessor and Runs it.
SET	Set or display coprocessor environment variables.

Numerical arguments can be entered in decimal or hexadecimal format; hexadecimal numbers must be preceded by 0x.

A detailed description of each command follows.

AUDIT <name> [<filename>]

Displays the coprocessor audit buffer identified by <name> (see MEM command). If the optional filename is specified, the audit listing is printed to the file.

BOARD

Displays all configuration details for the installed board.

BOOT

Resets the coprocessor, initializes all coprocessor memory, loads the bootstrap program and runs.

DUMP

Continue the memory dump (in hex format) from the previously specified address.

DUMP *

Displays the previously specified memory dump (in hex format) again. May be abbreviated "D *" or "DS".

DUMP <name>

Displays (in hex format) the coprocessor memory identified by <name> (see the MEM command).

Advanced Diagnostics

DUMP address [+offset] [length]

Displays (in hex format) the coprocessor memory beginning at the specified 32 bit address+offset for the length. If the optional length specification is omitted, 128 bytes is assumed.

DUMP segment: off set [length]

Displays (in hex format) the coprocessor memory beginning at the specified segment: offset. If the optional length specification is omitted, 128 bytes is assumed.

EDIT address [+offset] <data>

Stores all the specified data bytes beginning at the specified 32 bit address+offset.

EDIT segment: offset <data>

Stores all the specified data bytes beginning at the specified segment:offset.

HELP [<command> | all]

Produces a list of all available commands. If a specific command is specified, information on that particular command is displayed. If the keyword ALL is given, the detail information for all commands is listed.

LOAD [@addr] <filename>

Loads the specified .COM or .EXE file either at the specified address or at the next available memory block. The program does not start executing.

M E M

Displays a map of all the allocated and free Memory Control Blocks on the coprocessor. Each memory segment is named, for example: COPLHMOD, AUDIT, LHMODRSP, LHMODREQ, LHMODFRE. This is the name you use for those commands that require a <name>.

PRINT

Continues a memory dump (in ASCII format for all printable characters – 0x20 through 0x7E) from the previously specified address.

PRINT *

Displays the previously specified memory again (in ASCII format for all printable characters – 0x20 through 0x7E). May be abbreviated “P *” or “PS”.

PRINT <name>

Displays the coprocessor memory (in ASCII format for all printable characters – 0x20 through 0x7E) identified by <name> (see MEM command).

PRINT address [+offset] [length]

Displays the coprocessor memory (in ASCII format for all printable characters – 0x20 through 0x7E) beginning at the specified 32 bit address+offset for the length. If length is omitted, 128 bytes is assumed.

PRINT segment: off set [length]

Displays the coprocessor memory (in ASCII format for all printable characters – 0x20 through 0x7E) beginning at the specified segment:offset. If the optional length is omitted, 128 bytes is assumed.

QUIT or EXIT

Release all temporary resources back to the system, then exit.

RESTART

Resets the coprocessor and restarts the bootstrap, but does not re-initialize coprocessor memory. This is typically used to allow the loading of a post-crash utility or debugger for examining the board memory.

RUN [@addr] <filename>

Loads the specified .COM or .EXE file either at the specified address or at the next available memory block. The program begins execution upon loading.

SET

Display all environment variables.

SET <variable>

Display the specified environment variable.

SET <variable>=

Delete an environment variable.

SET <variable>=<string>

Set an environment variable for use by any program running on the coprocessor.

Problem Determination

If any processor, memory addressing or IRQ errors occur during the initial execution of *comload*, various error messages will be reported. You should proceed no further until you've verified that the I/O address, shared memory address and IRQ are available (not used by or assigned to other hardware) and that the necessary jumpers and switches are set correctly on the communication adapter board(s).

When you issue *comload*, the board is booted and the onboard software, *coplhmod*, is downloaded and executed. This action could detect one of the following conditions.

Incorrect or Conflicting I/O Address

Upon running *comload*, *coptool* performs a memory test of the coprocessor board. This memory test requires proper I/O addressing of the hardware. If the I/O address is not configured correctly or is in conflict with another product that was not detected during configuration, the following messages are displayed by *comload*

```
Testing/Clearing Memory. . . . Done
CLL Error: Illegal on board address specified
CLL Error: Illegal on board address specified
Failed in processing the configuration file coptool.ini
at line . . .
```

Correct the configuration, reboot the system and execute *comload* again.

Incorrect or Conflicting Memory Address

Before *comload* loads the coprocessor, it performs a memory test on the hardware- If there is a detectable shared memory address conflict or the board is addressed incorrectly, then the following messages are displayed:

```
Testing/Clearing Memory .... CLL Error: Illegal on board
address . . .
Failed in processing the configuration file coptool.ini
at line . . .
```

Not all shared memory conflicts, however, are detectable at this time. Additional products may have to be run before the conflict actually occurs, and this type of conflict may produce unpredictable system behavior. If the system is not operating as expected, try to determine what products used recently could cause a memory

conflict. More often than not, this type of problem is uncovered and corrected through a trial-and-error process.

Correct the configuration, unload the drivers (*comunload*), reboot the system and execute *comload* again.

Determine if the Board is Running

If you question whether or not the board even started, use the MEM command. It produces a memory map of the coprocessor. If there is a memory segment named COPLHMOD, then the board environment was downloaded and started. If there are memory segments named AUDIT, LHMODRSP, LHMODREQ and LHMODFRE, *coplhmod* properly determined the board type and created its shared memory queues.

At this point, the most useful command is AUDIT. Enter:

```
audit audit
```

or

```
a audit
```

The named segment (which happens, in this case, to be called AUDIT) will be formatted and displayed. There will be several unconditional messages in the audit buffer indicating the proper start of the coprocessor:

```
<coplhmod start : txdelay=0, rtsdelay=1, debug=0000><CLOCK>  
<clock wait><clock start OK>
```

These messages indicate memory, processor and timer operation.

Incorrect or Conflicting IRQ

After the board is downloaded and *coplhmod* is executing, a link will be opened between the LINKix code in the kernel and a specific port on the board. If the configuration specified using an interrupt from the board, additional conflicts may be detected at this time. Messages similar to the following will be displayed:

Advanced Diagnostics

```
{coptool boot stuff}  
.  
.  
SNA software initialized  
{several seconds later}  
Board 1 is not responding for port n.  
LINKix device driver error on function . . .
```

By enabling audits and examining the audits after a failure like this, you can determine whether or not the IRQ is indeed in conflict. The easiest test is to reconfigure the software to run in polled mode until you discover an available IRQ.

Correct PU Address Configured

If the board is functioning properly and receiving data from a host line, the audit buffer contains a message for each new poll address detected. When the configured poll address is encountered, a message to this effect is reported and all address display is halted:

```
<Poll: port=2,addr=01>  
<Poll : port=2,addr=03>  
<Poll : port=3,addr=01><configured address found>  
<Poll: port=2, addr=02><configured address found>
```

This whole process allows you to quickly determine if the host line is polling and if you are configured for the correct host address.

If there are no poll messages in the audit buffer, this may indicate other problems. By re-running *comload*, you can turn on additional levels of audit messages that will be reported. Before running *comload*, enter

```
CDEBUG="-d=0xn timer"  
export CDEBUG
```

where 0xn timer is a hexadecimal bit mask for the various audit levels:

8000	scc interrupts (very intrusive, rarely used)
4000	286i error
2000	RX data display
1000	TX data display

0800	open/close
0400	TX request/RX request
0200	polls
0100	board requests /responses
0010	time stamp
0008	RX errors

When testing with these audits is done, you can later delete the CDEBUG environment variable with the UNIX *unset* command.

The most common audit level value would be `-d=0x3D08`. This shows all the command and response flow between the host PC and the coprocessor board. With this information CLEO Technical Support staff can determine if there is a shared memory or interrupt conflict problem:

- If there is a `svphopen()` but no corresponding response, then DSR or DCD have not been asserted (depending on leased/switched line).
- If there is a `svphopen()` response but no `svphrx()` request, then the PC did not see the interrupt. There is either a conflict or a hardware failure. Configure the board for polled mode and try again.
- The display of RX/TX data is a shortcut to using other diagnostic tools. This is the most convenient trace when determining a board level problem. If, however, you're tracking a protocol/product problem, other diagnostic tools are a better choice.

Audit level 8000 is very processor intrusive at most baud rates. However, it is useful in determining if the board communication chip is detecting any activity on the line. It will be used in special circumstances.

Keep in mind that the audit buffer is a circular buffer and data within it can be overwritten. Whether or not, and how quickly, the data is overwritten depends on the amount of data going into the audit buffer (determined by the audit level). A snapshot of the audit buffer can be taken at any time by using the optional filename with the `AUDIT` command.

Advanced Diagnostics

Index

3270

3270 LUs

- adding, 6-4
 - adding a range, 6-5
 - assigning a connection, 6-8, 6-12
 - configuring, 6-3
 - deleting, 6-10
 - description, 6-8, 6-11
 - LU number, 6-12
 - LU pool, 6-13
 - adding, 6-14
 - deleting, 6-15
 - description, 6-17
 - model type (display LUs), 6-17
 - modifying, 6-15
 - name, 6-14
 - overriding model type, 6-17
 - pool size, 6-17
 - renaming, 6-16
 - model type (display LUs), 6-9, 6-12
 - overriding model type, 6-9, 6-12
 - renaming, 6-10
 - type (display or printer), 6-9, 6-12
- 3270 sessions
- APPC multiple sessions, 7-5
 - APPC parallel sessions, 7-5
 - remapping, 6-M
 - session numbers, 6-23
- 3270 terminal emulation, 1-1
- 3270 user alerts, 10-2
- 3270 users, 6-18
- adding, 6-20
 - configuring, 6-19
 - default 3-24, 6-18
 - default style file, 6-25
 - deleting, 6-21
 - description, 6-22

- maximum sessions, 6-22
- modifying the style file, 6-25
- NetView alerts, 6-25
- renaming, 6-21
- session remapping, 6-24
- viewing RTM data, 6-25

A

- Accelerator key, 2-22
- direct accelerator key, 2-8
- Alerts, 10-6
- Alternate character mode, 2-2
- API tracing
- APPC, 13-11
 - CPI-C, 13-12
 - CSV, 13-13
 - HLLAPI, 13-4
 - LUA, 13-13
- APPC
- API tracing, 13-11
 - VTAM compatibility, 7-4
- APPC LUs
- associating a mode with a pair of LUs, 7-38
 - dependent local LUs 7-26
 - dependent LUs, 7-5
 - implicit partner LUs, 7-4
 - implicit remote LUs, 7-27, 7-38
 - naming convention, 7-27
 - independent LUs, 7-5
 - local LUs, 7-27, 7-29
 - adding, 7-30
 - alias, 7-31
 - configuring, 7-29
- APPC LUs (*continued*)
- local LUs (continued)*
 - conversation-level security, 7-37
 - default APPC local LU pool, 7-36

- deleting, 7-32
- description, 7-34
- local network address, 7-35
- locally usable, 7-36
- maximum parallel sessions, 7-35
- modifying, 7-31
- partner LUs, 7-37
- renaming, 7-33
- security prevalidation, 7-37
- SNA network name, 7-34
- remote LUs, 7-20
 - adding, 7-21
 - alias, 7-22
 - configuration, 7-20
 - conversation-level security, 7-26
 - deleting, 7-23
 - description, 7-25
 - modifying, 7-22
 - parallel sessions, 7-26
 - renaming, 7-24
 - security prevalidation, 7-27
 - session-level security, 7-27
 - SNA network name, 7-25
 - SSCP alias, 7-26
- APPC mode, 7-2
 - adding, 7-10
 - associating with a pair of LUs, 7-38
 - configuring, 7-9
 - connection, 7-15
 - contention winner limits, 7-15
 - default mode parameters, 7-18
 - deleting, 7-12
 - description, 7-14
 - explicit mode names, 7-4
 - frame pacing, 7-16, 7-17
 - implicit modes, 7-3
 - configuring, 7-39
 - local mode, 7-7
 - mode name, 7-4, 7-14
 - normal mode, 7-4
 - parallel sessions, 7-15, 7-17
 - priority, 7-15
 - renaming, 7-13
 - request unit (RU) length, 7-16
 - negotiation, 7-18
 - session limit 7-15
 - single session, 7-15, 7-35
 - SNASVCMG, 7-17, 7-26, 7-36
- APPC TPs, 7-3
 - conversation-level security
 - adding a new user ID, 7-50
 - deleting a user ID, 7-51
 - invocable TPs
 - adding, 7-41
 - alias, 7-42
 - configuring, 7-40
 - conversation-level security, 7-40, 7-45, 7-48
 - deleting, 7-43
 - description, 7-44
 - environment variables, 7-46
 - executable file name, 7-46
 - full name, 7-45
 - loading method, 7-44
 - modifying, 7-42
 - naming format 7-45
 - non-queued, autostarted, 7-45
 - queued, autostarted, 7-45
 - queued, operator started, 7-44
 - renaming, 7-43
 - security verification, 7-45
 - start-up parameters, 7-46
 - target machine, 7-46
 - TP type, 7-45
 - wait timeout, 7-46

- Audit and error logs, 10-6, 13-2
 - configuring, 10-1
 - default message logging level, 10-8
 - name, 10-7
 - resetting, 10-9
- Autodial support (V.25bis), 11-12

C

- .configlk*, 4-13
- cconfig*, 4-7, 4-10, 4-16
- Check box, 2-13
- cleohw.cfg*, 3-16
- Client/server, 5-35
- com.cfg*, 4-18
- comsec*, 4-18
- comconfig*, 4-7, 4-10, 4-16
- Common Service Verbs (CSV), 1-2
- comprintcfg*, 4-19
- comservice*, 11-8
 - tracing options, 11-9
- comstart*, 3-25, 4-5, 11-2, 13-9
- comstop*, 3-25, 11-4
- Configuration
 - hardware
 - see hardware configuration
 - overview, 4-4
 - starting, 4-8
- Configuration file, 3-14, 4-5, 11-2
 - backups, 4-5
 - contents, 4-5
 - default 4-5
 - opening, 4-10
 - saving, 4-14
- Connection
 - adding, 5-46
 - deleting, 5-47
 - renaming, 5-48
 - to NetView program, 10-3
- Connections, 4-4, 5-9
 - SDLC, 5-11
 - activation, 5-12
 - adding/removing link services, 5-14
 - contact retry limit 5-17
 - contact timeout 5-17
 - data/poll retry limit 5-17
 - data/poll timeout 5-16
 - description, 5-11
 - full control point name, 5-14
 - half- or full-duplex modem, 5-16
 - link role with remote, 5-12
 - modem encoding parameter, 5-15
 - node IDs, 5-13
 - poll address, 5-16
 - poll rate, 5-17
 - remote end type, 5-11
 - switched connection establishment timeout 5-18
 - telephone number, 5-16
 - XID type, 5-13
 - SDLC parameters, 5-15
- Token Ring, 5-19
 - activation, 5-20
 - adding/removing link services, 5-22
 - description, 5-19
 - full control point name, 5-22
 - link role with remote, 5-20
 - maximum BTU length, 5-24
 - node IDs, 5-21
 - receive ack threshold 5-23
 - remote end type, 5-19
 - remote network address, 5-23
 - remote SAP address, 5-23
 - retry limit 5-23
 - unacknowledged send limit 5-24
- Token Ring parameters, 5-22

Connections (*continued*)

- X.25, 5-25
 - activation, 5-26
 - adding/removing link services, 5-29
 - default packet size, 5-33
 - default window size, 5-33
 - description, 5-25
 - facility data, 5-30
 - full control point name, 5-28
 - link role with remote, 5-26
 - node IDs, 5-27
 - PVC Alias, 5-33
 - PVC connection parameters, 5-33
 - remote address, 5-30
 - remote end type, 5-25
 - SVC connection parameters, 5-30
 - user data, 5-32
 - virtual circuit type, 5-28
 - XID type, 5-27
- Console messages
 - suppressing, 11-4
- Control daemon, 11-2, 13-9
 - initialization problems, 11-3
 - starting, 11-2, 11-7
 - stopping, 11-4
- convertcfg*, 4-7
- CPI-C
 - API tracing, 13-12
 - conversation security, 7-59
 - password, 7-60
 - user ID, 7-59
 - symbolic destination name, 7-53
 - adding, 7-54
 - configuring, 7-56
 - deleting, 7-56
 - description, 7-57
 - mode, 7-58
 - partner LU, 7-58
 - partner LU type, 7-58
 - partner TP name, 7-57
 - renaming, 7-56

CSV

- API tracing, 13-13
- CSV DEFINE_TRACE verb, 13-8
- Cursor addressability, 2-2
- custom*, 3-9
- cxconfig*, 4-10, 4-17

D

- Device file, 5-4, 5-6, 5-8
- Diagnostics
 - configuring, 10-1
 - daemon, 11-2
 - tools, 13-2
- Dialog fields
 - controlling, 4-1
 - moving between, 4-1
- Dialogs, 2-10
 - 3270 LU pools, 6-13
 - 3270 LUs, 6-3
 - 3270 RTM Settings, 6-28
 - 3270 Users, 6-19
 - Add a 3270 LU, 6-5
 - Add a 3270 User, 6-20
 - Add a Local APPC LU, 7-31
 - Add a Local Node, 5-49
 - Add a Range of 3270 LUs, 6-6
 - Add a Range of RJE LUs, 9-6
 - Add a Remote APPC LU, 7-22
 - Add a Symbolic Destination Name, 7-54
 - Add a TP User ID, 7-50
 - Add an Invocable TP, 7-42
 - APPC Modes, 7-10
 - Audit/Error Logs, 10-7
 - Configure *n* 3270 LUs, 6-7
 - Configure *n* RJE LUs, 9-7
 - Configure 3270 Alert Descriptions, 6-30
 - Configure 3270 LU, 6-11
 - Configure 3270 LU Pool, 6-16

Dialogs (*continued*)

- Configure 3270 User, 6-22
- Configure APPC Mode with ID, 7-11
- Configure Conversation Security for Symbolic Destination Name, 7-58
- Configure Descriptions for Alert, 6-31
- Configure Invocable TP, 7-44
- Configure Local APPC LU with Alias, 7-34
- Configure Local Node, 5-37
- Configure LUA LU, 8-10
- Configure Printer, 9-26
- Configure Printers for Workstation, 9-25
- Configure Punch, 9-31, 9-32
- Configure Punches for Workstation, 9-30
- Configure Remap List, 6-24
- Configure Remote APPC LU, 7-25
- Configure RJE FCB, 9-15, 9-17
- Configure RJE LU, 9-11
- Configure RJE Workstation, 9-22
- Configure SDLC Connection, 5-11
- Configure SDLC Link, 5-4
- Configure SDLC Parameters, 5-15
- Configure Session, 6-23
- Configure Symbolic Destination Name, 7-55, 7-56
- Configure TP Security, 7-49
- Configure TP Security Information, 7-52
- Configure TR Connection, 5-19
- Configure TR Link, 5-6
- Configure TR Parameters, 5-22
- Configure Vendor X.25 Link, 5-8
- Configure X.25 Connection, 5-25
- Configure X.25 Parameters (PVC), 5-33

- Configure X.25 Parameters (SVC), 5-30
- Connections, 5-9
- CPI-C Symbolic Destination Name, 7-53
- Delete a TP User ID, 7-51
- Invocable TPs, 7-41
- Link Usage Parameters, 5-39
- Links, 5-3
- Local APPC LUs, 7-30
- Local Nodes, 5-36
- LUA LU pool, 8-13
- LUA LUs, 8-4
- Network Management Connection, 10-3
- Remote APPC LUs, 7-21
- RJE Forms Control Buffers, 9-14
- RJE LUs, 9-4
- RJE Workstations, 9-20
- Save, 4-14
- Select Modes for Use with Partner LU, 7-38
- Usage Parameters for TR Link, 5-41
- Usage Parameters for X.25 Link, 5-42
- View Response Times for session, 6-27

Direct accelerator key, 2-8

Disabled option, 2-24

Display redrawing, 2-27

Dual list box, 2-21 moving items, 2-22

E

- Edit box, 2-16
 - associated with list boxes, 2-20
 - entering information, 2-16
 - operation, 4-1
- Elevator bar, 2-19

- Error log, 10-6
 - name, 10-8
 - resetting, 10-9

- Exit codes, C-1

- starting, 11-5
- stopping, 11-6
- tracing, 11-9

F

- FCB, 9-13

- File access permissions, 3-25

- Forms Control Buffer
 - see FCB

- Full control point name, 5-14, 5-22, 5-28

H

- Hardware configuration, 3-16

- HLLAPI

- API tracing, 13-4

- Host response time monitor, 6-25
 - configuring, 6-28

I

- Installation, 3-1

- additional setup, 3-14

- multiple hardware adapters, 3-1

- removal

- see Removing Cleo

- SCO UNIX, 3-9

- UnixWare, 3-4

L

- Leased line, 5-14, 5-21, 5-28

- Link, 4-4, 5-2

- Link failure alerts, 10-2

- Link service

- adding, 5-43

- deleting, 5-44

- renaming, 5-45

- Link services
 - SDLC, 5-4
 - constant carrier, 5-5
 - description, 5-4
 - device name, 5-4
 - physical line type, 5-5
 - port number, 5-4
 - Token Ring, 5-6
 - adapter device name, 5-6
 - description, 5-6
 - device name, 5-6
 - port number, 5-7
 - service access point (SAP)
 - address, 5-41
 - X.25, 5-8
 - description, 5-8
 - device name, 5-8
 - interface card, 5-8
- Link usage parameters, 5-38
- linkdump*, 13-22
- List box, 2-17
 - and associated edit boxes, 2-20
- Local node, 4-4, 5-35
 - adding, 5-49
 - Configuring, 5-35
 - deleting, 5-50
 - renaming, 5-52
 - starting, 11-5
 - stopping, 11-6
- LU pools, 6-2, 6-13
- LU type 0, 8-3
- LU types
 - for LUA, 8-3
- LUA
 - API tracing, 13-13
- LUA application programs, 8-2
- LUA LUs
 - adding, 8-4
 - adding a range, 8-5
 - assigning a connection, 8-11

- LUA LUs (*continued*)
 - assigning priority, 8-11
 - configuring, 8-10
 - deleting, 8-9
 - description, 8-8,8-10
 - number, 8-10
 - pools, 8-12
 - adding, 8-13
 - available LUs, 8-16
 - configuring, 8-15
 - deleting, 8-14
 - description, 8-16
 - LU priority, 8-16
 - pool size, 8-16
 - renaming, 8-15
 - renaming, 8-9
- LUA, 1-2

M

- Managing services
 - from the command line, 11-8
- Menu bar, 2-5
 - activating, 4-1
 - choosing a menu, 2-5
- Menus, 2-7
 - choosing an option, 2-8
- Message boxes, 2-26
 - errors, 2-26
 - information, 2-26
 - warnings, 2-26
- Modem encoding
 - NRZ/NRZI, 5-15

N

- NetView, 1-2
 - connection, 10-4
- NetView commands and alerts, 10-1
 - connection, 10-1
 - permissions, 10-1

- NetView user alerts, 6-25, 6-30
 - alert numbers, 6-31
 - configuring, 6-30
 - description, 6-32
 - user prompts, 6-32
- Network management connection
 - 10-2
- Network management daemon
 - tracing, 11-10
- Node ID, 5-13, 5-21, 5-27
- Nodes
 - starting, 11-5
 - stopping, 11-6
 - tracing, 11-9

O

- On-line help, xiv, 2-3, 2-24, 4-1

P

- Peer node, 5-16
- Problems
 - 3270 emulation, 12-2
 - 3270 sessions, 12-3
- Process ID, 13-10
- Pushbutton, 2-11
 - choosing, 2-11, 4-1

R

- Radio button, 2-14
- Radio groups, 2-14
- Removing Cleo
 - SCO UNIX, 3-13
 - UnixWare, 3-7
- Response Time Monitor (RTM), 6-27
- RJE FCBS, 9-13
 - adding, 9-14
 - configuring, 9-17
 - deleting, 9-15

- description, 9-17
- RJE FCBS (*continued*)
 - first line, 9-17
 - last line, 9-17
 - page length, 9-17
 - renaming, 9-16
 - vertical channel positions, 9-18
- RJE LUs, 9-3
 - adding, 9-4
 - adding a range, 9-5
 - assigning a connection, 9-12
 - associated workstation, 9-12
 - configuring, 9-11
 - deleting, 9-9
 - description, 9-11
 - LU number, 9-11
 - renaming, 9-10
- RJE menu, 9-13
- RJE workstations, 9-19
 - adding, 9-20
 - compress files to host 9-24
 - computer name, 9-23
 - configuring, 9-22
 - deleting, 9-21
 - description, 9-22
 - group id, 9-23
 - logon text 9-23
 - LUs assigned to workstation, 9-24
 - primary user, 9-23
 - printers, 9-25
 - adding, 9-25
 - append output to file 9-28
 - associated FCB, 9-27
 - configuring, 9-26
 - convert transparent data, 9-28
 - deleting, 9-26
 - description, 9-27
 - EBCDIC/ASCII translation, 9-28
 - output file name, 9-27
 - output path name, 9-28
 - output routing, 9-27
 - pass through PDIR, 9-29

- punches, 9-30
 - adding, 9-31
 - append output to file, 9-34
 - configuring, 9-32
 - convert transparent data, 9-34
 - deleting, 9-31
 - description, 9-32
 - EBCDIC/ASCII translation, 9-33
 - newline between records, 9-34
 - output file name, 9-33
 - output path name, 9-33
 - routing output 9-32
- renaming, 9-21
- session startup, 9-23

S

- Sample VTAM definition for SDLC, B-2
- Screen
 - redrawing, 2-27
 - size, 2-2
- SDLC
 - sample VTAM definition, B-2
- Security file, 3-14, 4-6
- Services
 - managing from the command line, 11-8
 - starting, 11-5
 - stopping, 11-6
 - tracing, 11-8
- Single-user mode, 3-1
- SLIM tracing, 11-4
- startsrv*, 11-5
- stopsrv*, 11-6
- STREAMS components, 4-2
- Style file, 3-15
 - default 6-25
 - defined, 6-33
 - purpose, 3-15

- Switched line, 5-14, 5-21, 5-28
- Switched SDLC line, 5-18
- Symbolic destination name, 7-53

T

- Technologies and guidelines
 - Common User Access (CUA), 2-1
- Terminal requirements, 2-2
- Terminfo database, 2-2
- Trace files, 13-2
- Tracing, 13-2
 - 3270 emulation
 - trace file(s), 13-6
 - APPC application, 13-4
 - APPC/CPI-C, 11-10, 13-11
 - application programs, 13-8
 - CPI-C, 13-12
 - CSV application, 13-4
 - entry size, 13-7
 - environment variables
 - SNACTL, 13-7
 - SNATRC, 13-5, 13-6
 - SNATRUNC, 13-7
 - HLLAPI application, 13-4
 - output sample, 13-16
 - trace file(s), 13-5
 - kernel space, 11-3, 13-5
 - limiting trace file size, 13-6
 - links, 11-9
 - LUA application, 13-4
 - messages
 - ASCII/EBCDIC translation, 13-10
 - network management daemon, 11-10
 - nodes, 11-9
 - output samples, 13-10
 - performance issues, 11-3
 - services, 11-8
 - SNA LAN Interface Module (SLIM), 11-4, 11-10

- trace files, 11-3
- trace types, 13-4
- UNIX Command Facility, 11-10
- user space, 13-5

Troubleshooting

- Cleo 3270, 12-2
- Cleo 3770, 12-7

U

UCF, 10-4

- security, 10-5
- tracing, 11-10
- user, 10-4

UNIX command facility

- see UCF

User access permissions, 3-24

User interface, 2-1

- terminal requirements, 2-2

V

V.25bis autodial, 11-12

v25cmd, 11-14

v25dial, 11-12

VTAM parameters

- ADDR, 5-16
- IDBLK, 5-14, 5-21, 5-28
- IDNUM, 5-14, 5-21, 5-28
- LINE/GROUP, 5-15
- LOCADDR, 6-12, 7-5, 8-7, 8-11, 9-12
- XID, 5-14, 5-22, 5-28

X

X.25 interface card, 5-8

XID

- type 0, 5-13, 5-27
- type 3, 5-13, 5-27

Common Service Verbs Programmer's Guide

June 1994

RESTRICTED RIGHTS

Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (C)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013. Cleo Communications.

LIMITED RIGHTS

Limited rights shall be effective indefinitely and are not subject to expiration as set forth in paragraph (3) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013.

Interface Systems, 5855 Interface Drive, Ann Arbor MI 48103



INTERFACE
SYSTEMS, INC.

June 10, 1994

Common Service Verbs Programmer's Guide

June 1994

Cleo Communications reserves the right to, without notice, modify or revise all or part of this document and/or change product features or specifications and shall not be responsible for any loss, cost or damage, including consequential damage, caused by reliance on these materials.

This document may not be reproduced, stored in a retrieval system or transmitted, in whole or in part, in any form or by any means (electronic, mechanical, photocopied or otherwise) without the prior written permission of Cleo Communications.

© 1991, 1992 Cleo Communications Æ All rights reserved.

Portions:

© 1989, 1990 Data Connection Limited

© 1989, 1990 Digital Communications Associates, Inc.

© 1989, 1990 Microsoft Corporation

Table of Contents

1. Introduction to CSV Concepts

Before You Begin	1-2
Intended Audience	1-2
Prerequisite Knowledge	1-2
Documentation Conventions	1-2
Summary of Common Service Verbs	1-3
Common Service Verbs in C Programs	1-4
Verb Control Block	1-4
ACSSVC_C Definition	1-5
Issuing a Verb	1-5
Compiling and Linking Programs	1-7
Writing Portable Applications	1-8

2. Common Service Verbs

CONVERT	2-2
COPY_TRACE_TO_FILE	2-6
DEFINE_TRACE	2-9
GET_CP_CONVERT_TABLE	2-14
LOG_MESSAGE .	2-18
TRANSFER_MS_DATA	2-22

A. Header File: acssvcc.h

B. Creating a Message File

1. Introduction to CSV Concepts

This chapter covers the following topics:

- ◆ Summary of Common Service Verbs
- ◆ Common Service Verbs in C Programs
- ◆ Compiling and Linking Programs
- ◆ Writing Portable Applications

Before You Begin

This manual contains the information you'll need to incorporate Cleo's Common Service Verbs (CSV) API in C-language application programs.

These Common Service Verbs are compatible with the Common Service Verbs provided by the DCA/Microsoft Communications Server and Communications Workstation, and those provided with the Common Services Programming Interface of IBM's OS/2 Extended Edition, version 1.1.

Intended Audience

The primary audience for this manual is the programmer who will be writing application programs that use Common Service Verbs.

Prerequisite Knowledge

Before reading this manual, you should have a knowledge of the following subjects:

- ◆ General product concepts — these are presented in the *Administration Guide*, the *3270 User's Guide*, the *3770 User's Guide* and the *Programmer's Guides* for the other advanced APIs.
- ◆ UNIX, version V.3 or newer
- ◆ The C language and a C compiler

Documentation Conventions

The following documentation conventions are used in this manual:

0xnn	A number in this format (for example 0x20) indicates a hexadecimal value.
<i>italics</i>	Italicized words are used for emphasis and to indicate the titles of manuals.
bold	Bold is used for cross-references to other chapters within this manual or to other sections within a chapter.

Summary of Common Service Verbs

This section provides a summary of the Common Service Verbs. Chapter 2 contains a detailed description of each verb.

CONVERT

Converts a character string from ASCII to EBCDIC or from EBCDIC to ASCII.

COPY_TRACE_TO_FILE

Copies the current contents of the trace file(s) to another file for storage.

DEFINE_TRACE

Enables or disables tracing for specific Application Program Interfaces (APIs).

GET_CP_CONVERT_TABLE

Creates and returns a 256-byte conversion table to translate character strings from a source code page to a target code page.

LOG_MESSAGE

Takes a message from a message file, adds specified data to it and records the message in the error or audit log file. This verb also optionally displays the message on the system console.

TRANSFER_MS_DATA

Builds a Systems Network Architecture (SNA) request unit (RU) containing Network Management Vector Transport (NMVT) data. The verb can send the NMVT data to NetView for centralized problem diagnosis and resolution. The data can also be logged in the local audit log file.

Common Service Verbs in C Programs

Common Service Verbs can be used in conjunction with application programs written in C. A C program calls Common Service Verbs through one of the following external functions:

- ◆ ACSSVC.C
- ◆ ACSSVC_P
- ◆ ACSSVC

Any one the three function names may be used. ACSSVC.C is the function name used in previous versions of LINKix; ACSSVC.P and ACSSVC are provided for compatibility with other CSV implementations.

Verb Control Block

The only parameter passed to the function is the address of a verb control block (VCB). The VCB is a structure made up of variables that:

- ◆ identify the verb to be executed,
- ◆ supply information to be used by the verb, and
- ◆ contain information returned by the verb when execution is complete.

Each verb has alike named VCB structure (declared in acssvcc.h):

CONVERT:	struct convert
COPY_TRACE_TO_FILE :	struct copy_trace_to_file
DEFINE_TRACE:	struct define_trace
GET_CP_CONVERT_TABLE :	struct get_cp_convert_table
LOG_MESSAGE:	struct log_message
TRANSFER_MS_DATA :	struct transfer_ms_data

The *acssvcc.h* file is delivered as part of the CSV support package. The file can be found in the */usr/include/linkix* directory. The content of the file is listed in Appendix A.

In order to use Common Service Verbs in a C application, you must `#include` the header file (*acssvcc.h*) in the application source program.

ACSSVC_C Definition

The three functions are defined as follows:

```
void ACSSVC_C (vcbptr)
struct svc_hdr * vcbptr;

void ACSSVC_P (vcbptr)
struct svc_hdr * vcbptr;

void ACSSVC (vcbptr)
struct svc_hdr * vcbptr;
```

In each case, the only parameter passed to the function is a pointer to the VCB.

Issuing a Verb

To issue a request for a Common Service Verb:

1. Create a structure variable (from the VCB structure) for the verb to be issued.

```
#include <acssvcc.h>

struct convert conv_block;
```

The VCB structures are declared in *acssvcc.h*; one of these structures is named `convert`

2. Clear (set to zero) the variables within the structure.

```
memset (conv_block, '\0' , sizeof ( conv_block ) ) ;
```

Introduction to CSV Concepts

This step is not required; however, it helps in debugging and reading memory dumps, and it eliminates the danger of future versions of a verb being sensitive to fields that are ignored in the current version.

3. Assign values to the required VCB variables.

```
conv_block.opcode = SV_CONVERT;
conv_block.direction = SV_ASCII_TO_EBCDIC;
conv_block.char_set = SV_AE;
conv_block.len = sizeof (tpstart_name) ;
conv_block.source = (char *) tpstart_name;
conv_block.target = (char *) tpstart.tp_name;
```

SV_CONVERT, SV_ASCII_TO_EBCDIC and SV_AE are symbolic constants representing integers. These constants are defined in *acssvcc.h*.

The character array *tpstart_name* contains an ASCII string to be converted to EBCDIC and placed in the character array *tpstart.tp_name*.

4. Invoke the verb. The only parameter is a pointer to the structure containing the VCB for the verb. Any of the following three entry points may be used:

```
ACSSVC_C (&conv_block) ;

ACSSVC_P (&conv_block) ;

ACSSVC (&conv_block) ;
```

5. Use the values returned by the verb.

```
i f ( conv_block.primary_rc == SV_OK ) {
    /* other statements */
}
```

Compiling and Linking Programs

The following is a sample command using the `cc` compiler to compile a program:

```
cc -I /usr/include/linkix -o program program. c -lcom
```

The `-I` option indicates the include path to the header file `acssvcc.h`; in the syntax above, `/usr/inchude/linkix`. (The program may also include other header files if the application makes use of HLLAPI, APPC, LUA, etc.)

The `-o` option gives the name of the output executable file, in this case *program*.

program.c is the name of the C program source file

The `-l` option indicates the name of the library to be used. In the syntax above, this option indicates the API library file `/usr/lib/libcom.a`.

Compiling Under AIX

If you'll be compiling and linking programs under AIX 3.2 or later, you should use one of the following samples as a guide:

```
cc -qlanglvl=ansi -I/usr/include/linkix -o program program. c -lcom
```

or

```
xlc -I/usr/include/linkix -o program program. c -lcom
```

Using the `-q` option with `cc` tells the compiler to set the language level to “ansi” during compilation (`cc`'s default language level is “extended”). This option is required when compiling some of our sample programs with `cc`.

Alternatively, you can use `xlc` which uses “ansi” as the default language level.

Writing Portable Applications

Cleo's implementation of Common Service Verbs is designed to be compatible with the verbs provided by the DCA/Microsoft Comm Server and Comm Workstation, and with those provided by IBM's OS/2 Extended Edition. However, those implementations are different in several ways from this implementation:

- ◆ The Comm Server and OS/2 Extended Edition implementations generate certain additional return codes which are not returned by Cleo's implementation; they may also make use of parameters which are shown as reserved in this manual.
- ◆ The Comm Server and OS/2 Extended Edition implementations use far pointers (far *) in all cases; UNIX does not have a concept of far and near pointers, so the word far must be omitted for LINKix.
- ◆ The OS/2 Extended Edition implementation includes two additional verbs, `DEFINE_DUMP` and `SET_USER_LOG_QUEUE`. These verbs are not included in the Cleo implementation because the functions they provide are specific to OS/2 Extended Edition and are not applicable. If you attempt to issue either verb, the primary return code `SV_INVALID_VERB` is returned.

The following guidelines provide some recommendations for writing CSV applications that will be portable to other environments:

- ◆ Include the header file `acssvcc.h` without a pathname prefix. Use the `-I /usr/include/linki.x` option on `cc` to locate the file. This allows the application to be used in an environment with a different file system.
- ◆ Specify the file name for the `COPY_TRACE_TO_FILE` verb without any pathname prefix, so that the file is always in the current directory. This allows the application to be used in an environment with a different file system.
- ◆ Use the symbolic constant names for parameter values and return codes, not the numeric values shown in the file `acssvcc.h`; *this* ensures that the correct value will be used regardless of the way these values are stored in memory. (See Appendix A, Listing of `acssvcc.h`, for more information.)
- ◆ Include a check for return codes other than those listed here and provide appropriate diagnostics.

- ◆ Ensure that any parameters shown as reserved are set to zero.

2. Common Service Verbs

This chapter contains a description of each of the Common Service Verbs, including the following information for each verb:

- ◆ Definition of the verb.
- ◆ Structure defining the verb control block (VCB) used by the verb. The structure is declared in the file *acssvcc.h*.
- ◆ Parameters (VCB fields) supplied to and returned by the verb. For each parameter, the following information is provided:
 - ◆ Description
 - ◆ Possible values
 - ◆ Additional information
- ◆ Additional information describing the use of the verb.

Most parameter values supplied to and returned by the Common Service Verbs are in hexadecimal format. To simplify coding, these values are represented by meaningful symbolic constants, which are established by **#define** statements in the header file *acssvcc.h*. For example, the operation code parameter value for the CONVERT verb is the hexadecimal value represented by `SV_CONVERT`.

It is important that you use the symbolic constants and not the hexadecimal values when setting values that are supplied to the CSV function, or when testing values returned by the CSV function. Because different UNIX machines store these values differently, using the symbolic value helps ensure portability to another machine.

If you're writing applications that will be used in other environments as well as with Cleo's implementation, see **Writing Portable Applications** in chapter 1.

CONVERT

The CONVERT verb translates an ASCII character string to EBCDIC format or an EBCDIC character string to ASCII format.

The string to be converted is called the source string. The converted string is called the target string.

CONVERT

Field	Type	Length	Notes
opcode	unsigned short	2	
opext	unsigned char	1	Reserved
reserv2	unsigned char	1	Reserved
primary_rc	unsigned short	2	
secondary_rc	unsigned long	4	
direction	unsigned char	1	
char_set	unsigned char	1	
len	unsigned short	2	
source	unsigned char *	4	
target	unsigned char *	4	

Supplied Parameters

opcode	SV_CONVERT
direction	SV_ASCII_TO_EBCDIC Convert from ASCII to EBCDIC characters. SV_EBCDIC_TO_ASCII Convert from EBCDIC to ASCII characters.
char_set	The character set to use in converting the source string. SV_A Use the “A” character set: Uppercase letters

Numerals 0-9

Special characters \$, #, @ and space

This character set is supported by a system-supplied type “A” conversion table. The first character of the source string must be an uppercase letter or the special character \$, # or @. Spaces are allowed only in trailing positions. Lowercase letters may be supplied in positions other than the first character, but are translated to uppercase.

SV_AE

Use the “AE” character set:

Uppercase letters

Lowercase letters

Numerals 0-9

Special characters \$, #, @, period and space

This character set is supported by a system-supplied type “AE” conversion table. The first character of the source string can be any character in the character set. Spaces are allowed only in trailing positions, unless the string consists entirely of spaces. No case conversion is performed.

SV_G

Use the “G” character set, supported by a custom conversion table. The file containing the table must be specified by the environment variable SNATBLG; set this variable to the full pathname of the file. (If the file is not found, the system returns the SV_TABLE_ERROR return code.)

The table must be an ASCII file 32 lines long. Each line must consist of 32 hexadecimal digits, representing 16 characters. The first 16 lines (256 characters) specify the EBCDIC characters to which ASCII characters are converted; the remaining 16 lines specify the ASCII characters to which EBCDIC characters are converted. The hexadecimal digits A through F can be either upper- or lowercase. However, you may want to use uppercase to ensure compatibility with IBM’s OS/2 Extended Edition.

You can use the GET_CP_CONVERT_TABLE verb to build a

Common Service Verbs: CONVERT

type “G” conversion table.

The file *comtblg.dat* (in */usr/lib/linkix/samples*), contains a sample type “G” conversion table. This table converts the first 127 characters of an ASCII code page to EBCDIC.

len	The number of characters to be converted.
source	The address of the buffer containing the source string.
target	The address of the buffer for the target string. This buffer can overlap or coincide with the source buffer and overwrite the source string.

Returned Parameters

Successful Execution

If the verb executes successfully, CSV returns:

primary_rc	SV_OK

Parameter Check

If the verb does not execute successfully because of a parameter error, CSV returns:

primary_rc	SV_PARAMETER_CHECK
secondary_rc	SV_CONVERSION_ERROR One or more characters in the source string were not found in the conversion table, or embedded spaces were found in a type “A” or “AE” string. These characters or spaces were converted to nulls (0x00). The verb still executed. SV_INVALID_CHARACTER_SET The char_set parameter contained an invalid value. SV_INVALID_DIRECTION The direction parameter contained an invalid value. SV_INVALID_FIRST_CHARACTER The first character of a type “A” source string was invalid.

SV_TABLE_ERROR

One of following occurred:

The file containing the type “G” custom conversion table, specified by the environment variable SNATBLG, was not found.

The table was not in the correct format.

SNATBLG was defined incorrectly.

Other Conditions

Other conditions can result in the following:

primary_rc

SV_INVALID_VERB

The opcode parameter did not match the operation code of any verb. No verb executed.

SV_UNEXPECTED_DOS_ERROR

The operating system has encountered an error while processing the verb. The operating system return code is returned through the secondary_rc. (For the meaning of this return code, see the file *errno.h*, which is specific to your particular UNIX computer). If the problem persists, consult the system administrator for corrective action.

COPY_TRACE_TO_FILE

The COPY_TRACE_TO_FILE verb copies the current contents of the API trace file or files to a new file, and clears the trace files. This allows you to save a copy of the current trace data for this application. For more information on API tracing, see chapter 13 of the *Administration Guide*, **Advanced Diagnostics**.

All API tracing on this application (for APPC, CPI-C, CSV, HLLAPI and/or LUA) must be stopped before you issue COPY_TRACE_TO_FILE. If any tracing is active, use the DEFINE_TRACE verb to stop it before using this verb.

COPY_TRACE_TO_FILE

Field	Type	Length	Notes
opcode	unsigned short	2	
opext	unsigned char	1	Reserved
reserv2	unsigned char	1	Reserved
primary.rc	unsigned short	2	
secondary.rc	unsigned long	4	
reserv3	unsigned char[8]	8	Reserved
file_name	unsigned char[64]	64	
file_option	unsigned char	1	
reserv4	unsigned char[12]	12	Reserved

Supplied Parameters

The program using this verb supplies the following parameters.

opcode	SV_COPY_TRACE_TO_FILE
file_name	The name (and optionally the path) of the file to hold the trace information; up to 64 characters. If the file is not in the current directory, specify the full path. If you set the file_option parameter to SV_NEW (see below), the file name specified must not be the name of an existing file.

Common Service Verbs: COPY_TRACE_TO_FILE

file_option	SV_NEW Create a new file with the name specified in file_name above. An error is returned if this file already exists.
	SV_OVERWRITE Overwrite the file if it exists, or create the file if it does not exist.

Returned Parameters

Successful execution
If the verb executes successfully, CSV returns:

primary_rc	SV_OK

Parameter check
If the verb does not execute successfully because of a parameter error, CSV returns:

primary_rc	SV_PARAMETER_CHECK
secondary_rc	SV_INVALID_FILE_OPTION The file_option parameter contained an invalid value.

State check
If the verb does not execute successfully because of a state error, CSV returns:

primary_rc	SV_STATE_CHECK
secondary_rc	SV_TRACE_BUFFER_EMPTY There was no trace information to copy to file. Either the trace files were empty, or the SNATRC environment variable was not set up. This environment variable must be set up before the application is started. For information on how to run and use traces, see chapter 13 of the <i>Administration Guide</i> , Advanced Diagnostics . SV_TRACE_NOT_STOPPED Tracing was still active when the verb was issued. Before issuing COPY_TRACE_TO_FILE, tracing for the CSV, APPC, CPI-C, HLLAPI and RUI interfaces must be turned off. Use

Common Service Verbs: COPY_TRACE_TO_FILE

DEFINE_TRACE to turn off any active tracing before issuing COPY_TRACE_TO_FILE; see the section on DEFINE_TRACE later in this chapter for more information.

Other conditions

Other conditions can result in the following:

primary.rc

SV_FILE_ALREADY_EXISTS

You specified the option SV_NEW for the file_option parameter (to create a new output file), but a file with the specified name already exists.

SV_INVALID_VERB

The opcode parameter did not match the operation code of any verb. No verb executed.

SV_OUTPUT_DEVICE_FULL

There was insufficient space in the output file's disk or directory to hold the trace information. The trace files were not reset the output file may contain some of the available trace information, but is not complete.

SV_UNEXPECTED_DOS_ERROR

The operating system has encountered an error while processing the verb. The operating system return code is returned through the secondary_rc. (For the meaning of this return code, see the file *erro.h*, which is specific to your particular UNIX computer). If the problem persists, consult the system administrator for corrective action. This error will be returned if you specify an invalid name or path for the output file, if you do not have write access to the output file, or if the file you specify is one of the original trace files.

DEFINE_TRACE

The DEFINE_TRACE verb enables or disables tracing for specified Application Program Interfaces (APIs).

The trace files must be set up (using the SNATRC environment variable) before the application which issues this verb is started. For information on how to run and use traces, see chapter 13 of the *Administration Guide*, **Advanced Diagnostics**.

DEFINE_TRACE

Field	Type	Length	Notes
opcode	unsigned short	2	
opext	unsigned char	1	Reserved
reserv2	unsigned char	1	Reserved
primary_rc	unsigned short	2	
secondary_rc	unsigned long	4	
reserv3	unsigned char [8]	8	Reserved
dt_set	unsigned char	1	
appc	unsigned char	1	
reserv4	unsigned char	1	Reserved
srpi	unsigned char	1	Not used
sdlc	unsigned char	1	Not used
tkn_mg_dlc	unsigned char	1	Not used
pcnet_dlc	unsigned char	1	Not used
dft	unsigned char	1	Not used
acdi	unsigned char	1	Not used
reserv5	unsigned char	1	Reserved
comrn.serv	unsigned char	1	
ehllapi	unsigned char	1	
x25_api	unsigned char	1	Not used
x25_dlc	unsigned char	1	Not used
twinax	unsigned char	1	Not used
reserv6	unsigned char	1	Reserved
rui	unsigned char	1	

Common Service Verbs: DEFINE_TRACE

Field	Type	Length	Notes
etherand	unsigned char	1	Not used
subsym	unsigned char	1	Not used
reserv7	unsigned char [8]	8	Reserved
reset_trc	unsigned char	1	
trunc	unsigned short	2	
strg_size	unsigned short	2	Reserved
reserv8	unsigned char [1]	1	Reserved
phys_link	unsigned char [8]	8	Reserved
reserv9	unsigned char [56]	56	Reserved

Parameters marked “Not used” are included for compatibility with OS/2 Extended Edition, but are not applicable to Cleo’s implementation. They maybe set in the same way as the appc or comm_serv parameters (see below); LINKix ignores them and does not generate an error return code.

Supplied Parameters

opcode	SV_DEFINE_TRACE
dt_set	Specifies whether tracing is turned on or off. SV_ON Enable tracing for a particular API if the parameter for that API (appc, comm_serv, rui or ehallpi) is set to SV_CHANGE. SV_OFF Disable tracing for a particular API if the parameter for that API is set to SV_CHANGE.
appc	Specifies whether the state of APPC tracing is to be changed. Note that this option controls both APPC and CPI-C tracing; they cannot be controlled independently. LINKix checks only the most significant bit (bit 0) of this byte; other bits are ignored. SV_CHANGE Enable or disable tracing for the Advanced Program-to-Program Communication (APPC) API and the Common

	programming Interface for Communications (CPI-C), depending on the dt_set parameter. SV_IGNORE Leave tracing in its current state for APPC and CPI-C.
comm_serv	Specifies whether the state of tracing for the Common Service Verbs is to be changed. SV_CHANGE Enable or disable tracing for Common Service Verbs, depending on the dt_set parameter. SV_IGNORE Leave tracing in its current state for Common Service Verbs.
ehllapi	Specifies whether the state of HLLAPI tracing (on or off) is to be changed. (Note that HLLAPI tracing can also be controlled using the TRON/TROFF options on the HLLAPI SET SESSION parameters function.) LINKix checks only the most significant bit (bit 0) of this byte; other bits are ignored. SV_CHANGE Enable or disable tracing for the High-Level Language Application programming Interface (HLLAPI) API, depending on the dt_set parameter. SV_IGNORE Leave tracing in its current state for HLLAPI.
rui	Specifies whether the state of tracing for the RUI interface of LUA (on or off) is to be changed. LINKix checks only the most significant bit (bit 0) of this byte; other bits are ignored. SV_CHANGE Enable or disable tracing for the RUI interface, depending on the dt_set parameter.

Common Service Verbs: DEFINE_TRACE

	SV_IGNORE Leave tracing in its current state for the RUI interface.
reset_trc	Specifies whether to reset the trace file or files. SV_YES Reset the trace file or files; that is, empty the files and discard their current contents. SV_NO Do not reset the trace files.
trunc	The length at which each trace record is to be truncated. Specify zero if you do not want truncation.

Returned Parameters

Successful Execution

If the verb executes successfully, CSV returns:

primary_rc	SV_OK

Parameter Check

If the verb does not execute successfully because of a parameter error, CSV returns:

primary_rc	SV_PARAMETER_CHECK
secondary_rc	SV_INVALID_SET The dt_set parameter contained an invalid value. SV_INVALID_RESET_TRACE The reset_trc parameter contained an invalid value.

Other Conditions

Other conditions can result in the following:

primary_rc	SV_INVALID_VERB The opcode parameter did not match the operation code of any verb. No verb executed.

SV_UNEXPECTED_DOS_ERROR

The operating system has encountered an error while processing the verb. The operating system return code is returned through the `secondary_rc`. (For the meaning of this return code, see the file *erro.h*, which is specific to your particular UNIX computer). If the problem persists, consult the system administrator for corrective action.

SNACTL Environment Variable

The SNACTL environment variable is provided for debugging application programs which use the DEFINE_TRACE verb. If this variable is set, DEFINE_TRACE verbs issued by the program have no effect on tracing (although they still return SV_OK unless an error occurs). This can be used to force tracing of a program which normally turns tracing off or to suppress tracing of a program which normally uses it. For more information on tracing and on this environment variable, see chapter 13 of the *Administration Guide*, **Advanced Diagnostics**.

GET_CP_CONVERT_TABLE

The GET_CP_CONVERT_TABLE verb creates and returns a 256-byte conversion table for translating character strings from a source code page to a target code page. A code page is a table that associates specific ASCII or EBCDIC values with specific characters, thus providing a national language variant of ASCII or EBCDIC.

Currently, most versions of UNIX do not provide support for different code pages and the national language characters associated with them and, therefore, is of limited use. It's provided here for compatibility with the Common Service Verbs provided by IBM's OS/2 Extended Edition and the DCA/Microsoft Communications Server or Communications Workstation. A list of supported code pages, and the national languages for which they are used, are at the end of this section.

GET_CP_CONVERT_TABLE

Field	Type	Length	Notes
opcode	unsigned short int	2	
opext	unsigned char	1	Reserved
reserv2	unsigned char	1	Reserved
primary_rc	unsigned short	2	
secondq_rc	unsigned long	4	
source_cp	unsigned short	2	
target_cp	unsigned short	2	
conv_tbl_addr	unsigned char *	4	
char_not_fnd	unsigned char	1	
substitute_char	unsigned char	1	

Supplied Parameters

opcode	SV_GET_CP_CONVERT_TABLE
source_cp	A decimal number which identifies the source code page to be used. See Supported Code Pages , at the end of this section, for a list of valid code page numbers.

Common Service Verbs: GET_CP_CONVERT_TABLE

target_cp	A decimal number which identifies the target code page to be used. See Supported code pages , at the end of this section, for a list of valid code page numbers.
conv_tbl_addr	The address of the buffer for the 256-byte conversion table.
char_not_fnd	Specifies the action to take if a character in the source code page does not exist in the target code page. SV_ROUND_TRIP Store a unique value in the conversion table for each source code-page character. This value is useful only if you build a second conversion table to convert between the same two code pages in the reverse direction. If you specify the SV_ROUND_TRIP value in building both conversion tables, any character translated from one code page to the other and then back is unchanged. SV_SUBSTITUTE Store a substitute character (specified by the substitute_char parameter) in the conversion table. Converting the translated character string back to the original code page does not necessarily recreate the original character string.
substitute_char	Specifies the character to store in the conversion table when a character from the source code page has no equivalent in the target code page. Use this parameter only if the char_not_fnd parameter is set to SV_SUBSTITUTE. When the target code page is an EBCDIC code page, this parameter should be set to the EBCDIC value of the character you want to use, not to the actual character. For example, to use the I character as the substitute character in an ASCII to EBCDIC conversion table, you should supply the value 4F (the value associated with the character I in EBCDIC) and not the actual character. When the target code page is an ASCII code page, you can specify either the character or its ASCII value.

Common Service Verbs: GET_CP_CONVERT_TABLE

Returned Parameters

Successful Execution

If the verb executes successfully, CSV returns:

primary_rc SV_OK

Parameter Check

If the verb does not execute successfully because of a parameter error, CSV returns:

primary_rc SV_PARAMETER_CHECK

secondary_rc SV_INVALID_CHAR_NOT_FOUND
The `char_not_fnd` parameter contained an invalid value.

SV_INVALID_SOURCE_CODE_PAGE
The code page specified by the `source_cp` parameter is not supported.

SV_INVALID_TARGET_CODE_PAGE
The code page specified by the `target_cp` parameter is not supported.

Other Conditions

Other conditions can result in:

primary_rc SV_INVALID_VERB
The opcode parameter did not match the operation code of any verb. No verb executed.

SV_UNEXPECTED_DOS_ERROR
The operating system has encountered an error while processing the verb. The operating system return code is in `secondary_rc` (see *errno.h* for the meaning).

Supported Code Pages

The following is a list of the supported code pages and the national language variants of ASCII or EBCDIC which use each code page.

ASCII code pages

8859	Generalized ASCII code page defined by ISO 8859, used to support all language variants.
437	US English
850	International code page (contains characters for US English, UK English, French, German, Italian, Spanish, Finnish, Netherlands, Swedish, Swiss, Belgian, Latin American)
865	Danish, Norwegian
860	Portuguese
863	Canadian French

EBCDIC code pages

037	US English, Canadian Bilingual, Netherlands, Portuguese
273	German
277	Danish, Norwegian
278	Finnish, Swedish
280	Italian
284	Spanish, Latin American
285	UK English
297	French
500	Belgian (New), Swiss French, Swiss German

LOG_MESSAGE

The LOG_MESSAGE verb records a message in the error or audit log file. The message is taken from an existing message file. This verb can also insert additional information in the message. The message can optionally be displayed on the system console.

LOG_MESSAGE

Field	Type	Length	Notes
opcode	unsigned short	2	
opext	unsigned char	1	Reserved
reserv2	unsigned char	1	Reserved
primary_rc	unsigned short	2	
secondary_rc	unsigned long	4	
msg_num	unsigned short	2	
origintr_id	unsigned char [8]	8	
msg_file_name	unsigned char [3]	3	
msg_act	unsigned char	1	
msg_ins_len	unsigned short	2	
msg_ins_ptr	unsigned char *	4	

Supplied Parameters

opcode	SV_LOG_MESSAGE
msg_num	Number of the message in the message file specified by msg_file_name.
origintr_id	Name of the component issuing the LOG_MESSAGE verb; a string of up to eight characters. LINKix accepts up to eight characters for compatibility with IBM's OS/2 Extended Edition; the first six characters will be used in the log file to identify the originator of the message.

	This parameter is optional; set the first byte to 0x00 if you do not want to display the originator ID in the log file.
msg_file_name	Name of the file containing the message to be logged. The file name must be three characters long. Pad with blanks if necessary. The <i>.msg</i> extension is added automatically. To specify the default message file, <i>/usr/lib/linkix/com.msg</i>, set the first byte to 0x00. If you specify your own message file rather than the default, the file must be in the directory <i>/usr/lib/linkix</i>. (Appendix B, Creating a Log Message File, contains complete information for creating your own message file.)
msg_act	Action to be taken when processing the message. SV.INTRV Log the message with a severity level of 12; display the message on the system console. SV_NO_INTRV Log the message with a severity level of 12; do not display the message. SV_INTRV_16 Log the message with a severity level of 16; display the message on the system console. SV_NO_INTRV_16 Log the message with a severity level of 16; do not display the message. SV_NO_INTRV_10 Log the message with a severity level of 10; do not display the message. SV_NO_INTRV_8 Log the message with a severity level of 8; do not display the message. SV_NO_INTRV_6 Log the message with a severity level of 6; do not display the message.

Common Service Verbs: LOG_MESSAGE

For information on severity levels, see the chapter on **Troubleshooting** in the *Administration Guide*.

Note that the system administrator may specify that messages not be displayed on the system console (this option is specified when starting the control daemon). If this is the case, the message will be logged but will not be displayed.

Note also that values other than SV_INTRV and SV_NO_INTRV are specific to the LINKix implementation and are not supported by other CSV implementations.

msg_ins_len Length of data to be inserted into the message. (See msg_ins_ptr below for more information on this.) Specify zero if no data is to be inserted.

The total length of the message (40 byte header, message text and inserted data) should not exceed 256 bytes. If the message is longer than 256 bytes, only the header and inserted text are logged; the message text is left out.

rmsg_insptr Address of the data to be inserted into the message. The data consists of a series of up to nine null-terminated strings. (Because IBM's OS/2 Extended Edition, versions 1.1 and 1.0, supports only three data strings, you may want to limit the inserted text to three strings to ensure compatibility with that product.)

Use this parameter only if msg_ins_len is greater than zero.

When you create the log message file, you can specify where in the message the additional data is to be inserted. For further information, see Appendix B, **Creating a Log Message File**.

Returned Parameters

Successful Execution

If the verb executes successfully, CSV returns:

primary_rc SV_OK

Common Service Verbs: LOG_MESSAGE

Parameter Check

If the verb does not execute successfully because of a parameter error, CSV returns:

primary_rc	SV_PARAMETER_CHECK
secondary_rc	SV_INVALID_MESSAGE_ACTION The msg_act parameter contained an invalid value.

Note that there is no PARAMETER_CHECK secondary return code indicating that the message file name was incorrect or that the message file was not found; this error causes a return code of SV_UNEXPECTED_DOS_ERROR.

Other Conditions

Other conditions can return the following:

primary_rc	SV_COMM_SUBSYSTEM_NOT_LOADED The control daemon has not been started. Contact the system administrator for corrective action. Note that this return code only occurs when using a user-defined message file, not when using the default message file <i>com.msg</i> .
	SV_INVALID_VERB The opcode parameter did not match the operation code of any verb. No verb executed.
	SV_UNEXPECTED_DOS_ERROR The operating system has encountered an error while processing the verb. The operating system return code is returned through the secondary_rc. (For the meaning of this return code, see the file <i>errno.h</i> , which is specific to your particular UNIX computer). If the problem persists, consult the system administrator for corrective action.

TRANSFER_MS_DATA

The TRANSFER_MS_DATA verb builds a Systems Network Architecture (SNA) request unit (RU) containing Network Management Vector Transport (NMVT) data. The verb can send the NMVT data to NetView for centralized problem diagnosis and resolution. The data can also be logged in the local audit file.

Note that the CSV library uses the UNIX **sigpoll** signal during the processing of this verb; if you need to use this signal in your application, see **Use of Signals** at the end of this section. (This signal is not used during processing of the other CSV verbs.)

TRANSFER_MS_DATA

Field	Type	Length	Notes
opcode	unsigned short	2	
data_type	unsigned char	1	
reserv2	unsigned char	1	Reserved
primary_rc	unsigned short	2	
second_rc	unsigned long	4	
options	unsigned char	1	
reserv3	unsigned char	1	Reserved
origintr_id	unsigned char [8]	8	
dlen	unsigned short	2	
dptr	unsigned char *	4	

Supplied Parameters

opcode SV_TRANSFER_MS_DATA

data_type	Type of data provided by this verb. SV.NMVT Network Management Vector Transport (NMVT): this data includes the Network Services (NS) header, the major Network Management Vector and subvectors. SV_ALERT_SUBVECTORS RU containing data for an alert, in the appropriate format, without the NS header or major NMVT vector. SV_PDSTATS_SUBVECTORS RU containing data for problem determination statistics, in the suitable format, without the NS header or major NMVT vector. SV_USER_DEFINED User-defined data; this data is recorded in the audit log file but cannot be sent to NetView.
options	This parameter is a one-byte value, with individual bits indicating the options selected. Bit 0 is the most significant bit and bit 7 the least significant bit. (Bits 1Æ 3 are ignored if data_type contains SV_USER_DEFINED.) SV_ADD SV_NO_ADD Bit 0: Add date/time subvector to data. Bit 1: Add Product_Set_ID subvector to data. This allows Network Management Services to identify the sender of an alert. SV_SEND SV_NO_SEND Bit 2: Send the data on the active SSCP-PU session using the connection configured for diagnostics if the session is active. SV_ADD SV_NO_ADD Bit 3: Log the data in the audit log file.

Common Service Verbs: TRANSFER_MS_DATA

If bit 2 is set to SV_SEND, this option specifies whether or not to log the data to the audit log file. (If this option is selected, the data is still logged even if the attempt to send the data fails.)

If bit 2 is set to NO_SEND, or if data_type is set to SV_USER_DEFINED, this option is ignored; the data is always logged.

Bits 4-7: Not used.

origintr_id	Name of the component issuing the TRANSFER_MS_DATA verb. This name is included when the data is logged in the audit log file. This parameter is optional. Set the first character of the array to 0x00 if you want the system to ignore it.
dlen	Length of data to be supplied to this verb. The total length of the data (user-supplied data and any added headers or subvectors) must fit into one RU. The maximum RU length is 512 bytes.
dptr	Address of the data to be sent.

Returned Parameters

Successful Execution

If the verb executes successfully, CSV returns:

primary_rc	SV_OK

Parameter Check

If the verb does not execute successfully because of a parameter error, CSV returns:

primary_rc	SV_PAFLWIETER_CHECK

secondary_rc	SV_DATA_EXCEEDS_RU_SIZE

The data that was to be sent was too long. The length of the user-supplied data plus headers and added subvectors must fit in a single RU that is not more than 512 bytes long.

Common Service Verbs: TRANSFER_MS_DATA

SV_INVALID_DATA_TYPE

The data_type parameter contained an invalid value.

State check

If the verb does not execute successfully because of a state error, CSV returns:

primary_rc SV_STATE_CHECK

secondary_rc SV_SSCP_PU_SESSION_NOT_ACTIVE
The NMVT was not sent; the local node owning the network management connection was not active, the SSCP-PU session was not active or no network management connection was configured.

Other Conditions

Other conditions can result in the following:

primary_rc SV_COMM_SUBSYSTEM_NOT_LOADED
The LINKix software has not been loaded. Contact your System Administrator for corrective action.

SV_INVALID_VERB
The opcode parameter did not match the operation code of any verb. No verb executed.

SV_UNEXPECTED_DOS_ERROR
The operating system has encountered an error while processing the verb. The operating system return code is returned through the secondary_rc. (For the meaning of this return code, see the file *errno.h*, which is specific to your particular UNIX computer). If the problem persists, consult the system administrator for corrective action.

Use of Signals

To detect work arriving from other software components, the CSV library makes use of the UNIX **sigpoll** signal while processing the TRANSFER_MS_DATA verb. Because applications may require this signal for other purposes, VLRs daisy-chain to any existing signal catchers.

When a signal arrives, it may or may not indicate work for your application to process. The CSV library sets up a signal catcher for the signal, but remembers the

Common Service Verbs: TRANSFER_MS_DATA

address of the signal catcher previously in force. When the library has finished processing the signal, it calls the previous signal catcher.

Each TRANSFER_MS_DATA verb initializes the signal catcher and daisy-chains the old signal catcher. The new signal catcher is unlinked when the verb completes.

An application must not issue another TRANSFER_MS_DATA verb from within a signal catcher.

A. Header File `acssvcc.h`

This appendix contains a listing of the header file `acssvcc.h`. The contents of this file are as follows:

- ◆ A **#include** statement for the file `svcon.g.h`. This file is not listed here because its contents are specific to your particular type of UNIX system. If you cannot find this file, see your system administrator.
- ◆ The definition of the `PFLIPI` and `PLFLIPIPLFLIPI` macros. These macros ensure that hexadecimal values are stored in the correct format. The definition depends on the setting of `PFLIPRQD`, which is in the file `svconfig.h`.
- ◆ The Common Service Verbs function prototype.
- ◆ The structure declarations for the Common Service Verbs verb control blocks (VCBs).
- ◆ The **#define** statements that substitute meaningful symbolic constants for the actual hexadecimal values that are supplied to and returned by the Common Service Verbs. Each **#define** statement uses the `PFLIPI` or `PLFLIPI` macro (defined at the start of the file) to indicate how the hexadecimal value is stored in memory.

When setting or testing Common Service Verbs parameters, use the symbolic constants defined in this file rather than the hexadecimal values. This ensures portability to any type of UNIX computer.

Header File: acssvcc.h

```
/**INC+*****  
/*  
/* Include: ACSSVCC.H  
/*  
/* Purpose: Common Service Verb library standard header file for  
/*          EEI.2 compatibility.  
/*  
/* (C) COPYRIGHT Cleo COMMUNICATIONS 1993  
/*  
/**INC-*****  
  
/*  
/* For a byte swapped CPU such as Intel 80x86 set PFLIPRQD to 1  
/* For unswapped machine such as Motorola 68k set PFLIPRQD to 0  
/*  
#include <svconfig.h>  
  
#ifndef PFLIPI  
#undef PFLIPI  
#undef PLFLIPI  
#endif  
  
#ifndef PFLIPRQD  
#if PFLIPRQD==1  
#define PFLIPI(X) ((unsigned short)((((unsigned long)(X) & 0xffff) * 0x10001) \  
    >> 8))  
#define PLFLIPI(X) (PFLIPI(((unsigned long)X & 0x0000ffff)) << 16 | \  
    PFLIPI((unsigned long)X >> 16))  
#else  
#define PFLIPI(X) (X)  
#define PLFLIPI(X) (X)  
#endif  
#else  
**** PFLIPRQD not specified in svconfig.h  
#endif  
  
/*  
/* Function prototype for the Common Service Verbs  
/*  
extern void ACSSVC_C();  
extern void ACSSVC_P();  
extern void ACSSVC();  
  
/*  
/* VCB structures  
/*  
/**STRUCT+*****  
/* Structure: svc_hdr  
/*  
/* Description: Common verb control block (vcb) header  
/*  
struct svc_hdr  
{  
    unsigned short opcode; /* Verb identifying operation code. */  
    unsigned char opext; /* Verb extension code - reserved. */  
    unsigned char reserv2; /* Reserved. */  
    unsigned short primary_rc; /* Primary return code from verb. */  
    unsigned long secondary_rc; /* Secondary (qualifying) return code. */  
};  
/**STRUCT-*****  
  
/**STRUCT+*****  
/* Structure: convert  
/*  
/* Description: verb control block for convert verb  
/*  
struct convert  
{  
    unsigned short opcode; /* Verb identifying operation code. */
```

```

    unsigned char opext;          /* Verb extension code - reserved. */
    unsigned char reserv2;        /* Reserved. */
    unsigned short primary_rc;    /* Primary return code from verb. */
    unsigned long secondary_rc;   /* Secondary (qualifying) return code. */
    unsigned char direction;      /* Direction of conversion - ASCII to */
                                /* EBCDIC or vice-versa. */
    unsigned char char_set;       /* Character set to use for the */
                                /* conversion A, AE, or user-defined G. */
    unsigned short len;           /* Length of string to be converted. */
    unsigned char *source;        /* Pointer to string to be converted. */
    unsigned char *target;        /* Address to put converted string at. */
};
/**STRUCT-*****/

/**STRUCT+*****/
/* Structure: define_trace */
/*
/* Description: verb control block for define_trace verb
/* *****/
struct define_trace
{
    unsigned short opcode;        /* Verb identifying operation code. */
    unsigned char opext;          /* Verb extension code - reserved. */
    unsigned char reserv2;        /* Reserved. */
    unsigned short primary_rc;    /* Primary return code from verb. */
    unsigned long secondary_rc;   /* Secondary (qualifying) return code. */
    unsigned char reserv3[8];     /* Reserved. */
    unsigned char dt_set;         /* Trace state to be set (on/off). */
    /* *****/
    /* In the following fields, used for specifying which types of tracing are */
    /* required, only the most significant bit is currently used. Bits 1 to 7 */
    /* are reserved. Bit 0 can be set to 'ignore' or to 'change'. */
    /* *****/
    unsigned char appc;           /* Tracing for APPC. */
    unsigned char reserv4;        /* Reserved. */
    unsigned char srpi;           /* Reserved. */
    unsigned char sdic;           /* Reserved. */
    unsigned char tkn_rng_dlc;     /* Reserved. */
    unsigned char pcnet_dlc;      /* Reserved. */
    unsigned char dft;            /* Reserved. */
    unsigned char acdi;           /* Reserved. */
    unsigned char reserv5;        /* Reserved. */
    unsigned char comm_serv;      /* Tracing for Comm_Serv_API. */
    unsigned char ehllapi;        /* Tracing for HLLAPI. */
    unsigned char x25_api;        /* Reserved. */
    unsigned char x25_dlc;        /* Reserved. */
    unsigned char twinax;         /* Reserved. */
    unsigned char reserv6;        /* Reserved. */
    unsigned char rui;            /* Tracing for RUI interface of LUA */
    unsigned char etherand;       /* Reserved. */
    unsigned char subsym;         /* Reserved. */
    unsigned char reserv7[8];     /* Reserved. */
    unsigned char reset_trc;      /* Flag to reset the trace files. */
    unsigned short trunc;         /* Truncation size for trace records. */
    unsigned short strg_size;     /* Reserved. */
    unsigned char reserv8[1];     /* Reserved. */
    unsigned char phys_link[8];   /* Reserved. */
    unsigned char reserv9[56];    /* Reserved. */
};
/**STRUCT-*****/

/**STRUCT+*****/
/* Structure: get_cp_convert_table */
/*
/* Description: verb control block for get_cp_convert_table verb
/* *****/
struct get_cp_convert_table
{
    unsigned short opcode;        /* Verb identifying operation code. */
    unsigned char opext;          /* Verb extension code - reserved. */
    unsigned char reserv2;        /* Reserved. */
    unsigned short primary_rc;    /* Primary return code from verb. */
}

```

Header File: acssvcc.h

```
    unsigned long secondary_rc;      /* Secondary (qualifying) return code. */
    unsigned short source_cp;        /* Source code page for conversion table. */
    unsigned short target_cp;        /* Target code page for conversion table. */
    unsigned char *conv_tbl_addr;    /* Address to put conversion table at. */
    unsigned char char_not_fnd;      /* Character not found option: either */
    /* substitute character or round trip. */
    unsigned char substitute_char;    /* Substitute character to use. */
};
/**STRUCT-*****/

/**STRUCT-*****/
/* Structure: log_message */
/*
/* Description: verb control block for log_message verb
*****/
struct log_message
{
    unsigned short opcode;           /* Verb identifying operation code. */
    unsigned char opext;             /* Verb extension code - reserved. */
    unsigned char reserv2;           /* Reserved. */
    unsigned short primary_rc;       /* Primary return code from verb. */
    unsigned long secondary_rc;      /* Secondary (qualifying) return code. */
    unsigned short msg_num;          /* Number of message to log. */
    unsigned char origntr_id[8];     /* ID of the originator of the message. */
    unsigned char msg_file_name[3]; /* Message file to search for the required */
    /* message number. */
    unsigned char msg_act;           /* Message action - whether to inform */
    /* operator. */
    unsigned short msg_ins_len;       /* Length of data for insertion in msg. */
    unsigned char *msg_ins_ptr;      /* Address of data for insertion in msg. */
};
/**STRUCT-*****/

/**STRUCT-*****/
/* Structure: transfer_ms_data */
/*
/* Description: verb control block for transfer_ms_data verb
*****/
struct transfer_ms_data
{
    unsigned short opcode;           /* Verb identifying operation code. */
    unsigned char data_type;         /* Data type supplied to the verb - can */
    /* be: user-defined; alert subvectors; */
    /* pdstats subvectors; NMVT. */
    unsigned char reserv2;           /* Reserved. */
    unsigned short primary_rc;       /* Primary return code from verb. */
    unsigned long secondary_rc;      /* Secondary (qualifying) return code. */
    unsigned char options;           /* Verb options. Bits 4 - 7 are reserved. */
    /* Bit 0: Time stamp subvector option. */
    /* Bit 1: Product set id subvector option. */
    /* Bit 2: Option to send NMVT to SSCP. */
    /* Bit 3: Option for local logging. */
    unsigned char reserv3;           /* Reserved. */
    unsigned char origntr_id[8];     /* ID of originator of NMVT. */
    unsigned short dlen;             /* Length of data supplied to verb. */
    unsigned char *dptr;             /* Address at which supplied data resides. */
};
/**STRUCT-*****/

/**STRUCT-*****/
/* Structure: copy_trace_to_file */
/*
/* Description: verb control block for copy_trace_to_file verb
*****/
struct copy_trace_to_file
{
    unsigned short opcode;           /* Verb identifying operation code. */
    unsigned char opext;             /* Verb extension code - reserved. */
    unsigned char reserv2;           /* Reserved. */
    unsigned short primary_rc;       /* Primary return code from verb. */
    unsigned long secondary_rc;      /* Secondary (qualifying) return code. */
    unsigned char reserv3[8];        /* Reserved. */
};
```

```

    unsigned char file_name[64];      /* Trace file name. */
    unsigned char file_option;        /* File options. New or overwrite. */
    unsigned char reserv4[12];        /* Reserved. */
};
/**STRUCT-*****/

/* The following two verbs are not implemented, but their structures are
 * present for compatibility with EE.
 */
/**STRUCT+*****/
/* Structure: set_user_log_queue */
/*
 * Description: verb control block for set_user_log_queue verb
 */
struct set_user_log_queue
{
    unsigned short opcode;            /* Verb operation code */
    unsigned char opext;              /* Reserved. */
    unsigned char reserv2;            /* Reserved. */
    unsigned short primary_rc;        /* Primary RETURN_CODE */
    unsigned long secondary_rc;       /* Secondary RETURN_CODE */
    unsigned char reserv3[8];         /* Reserved. */
    unsigned char queue_name[64];     /* Queue on which messages are to be put */
    unsigned char forward;            /* Types of messages to be put on queue */
    unsigned char suppress;           /* Messages to be suppressed */
    unsigned char selection;          /* Messages to be forwarded */
    unsigned short numbers[10];       /* Message No.s to be forwarded */
};
/**STRUCT-*****/

/* Structure: define_dump */
/*
 * Description: verb control block for define_dump verb
 */
struct define_dump
{
    unsigned short opcode;            /* Verb operation code */
    unsigned char opext;              /* Reserved. */
    unsigned char reserv2;            /* Reserved. */
    unsigned short primary_rc;        /* Primary RETURN_CODE */
    unsigned long secondary_rc;       /* Secondary RETURN_CODE */
    unsigned char reserv3[8];         /* Reserved. */
    unsigned char component_01;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_02;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_03;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_04;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_05;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_06;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_07;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_08;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_09;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_10;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_11;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_12;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_13;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_14;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_15;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_16;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_17;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_18;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_19;       /* ACTIVATE or IGNORE component dump */
    unsigned char component_20;       /* ACTIVATE or IGNORE component dump */
    unsigned char reserv4[11];        /* Reserved. */
    unsigned char x25_control_blocks; /* ACTIVATE or IGNORE x25 cb dump */
    unsigned char x25_user_buffers;   /* ACTIVATE or IGNORE x25 ub dump */
    unsigned char x25_adapter_name[8]; /* Adaptor name for x25 dumps */
    unsigned char file_name[64];      /* File to dump to */
    unsigned char file_option;        /* OVERWRITE old file or create NEW */
};

```

Header File: acssvcc.h

```
/**STRUCT-*****/
/*****
/*      Verb op codes.      */
*****/

#define SV_CONVERT          PFLIPI(0x1A00)    /* X'1A00'    */
#define SV_DEFINE_TRACE     PFLIPI(0x1D00)    /* X'1D00'    */
#define SV_GET_CP_CONVERT_TABLE PFLIPI(0x1900) /* X'1900'    */
#define SV_LOG_MESSAGE      PFLIPI(0x1F00)    /* X'1F00'    */
#define SV_TRANSFER_MS_DATA PFLIPI(0x1C00)    /* X'1C00'    */

#define SV_DEFINE_DUMP      PFLIPI(0x4D00)    /* X'4D00'    */
#define SV_COPY_TRACE_TO_FILE PFLIPI(0x4C00) /* X'4C00'    */
#define SV_SET_USER_LOG_QUEUE PFLIPI(0x4B00) /* X'4B00'    */

/*****
/*      Parameters for the Common Service Verbs.      */
*****/

#define SV_ADD                0
#define SV_NO_ADD             1
#define SV_CHANGE             1
#define SV_IGNORE             0
#define SV_SEND               0
#define SV_NO_SEND            1

#define SV_NO                  0x00
#define SV_YES                 0x01
#define SV_A                   0x01
#define SV_AE                  0x00
#define SV_ALERT_SUBVECTORS   0x02
#define SV_ASCII_TO_EBCDIC    0x00
#define SV_INTRV              0x00
#define SV_NO_INTRV           0x01
#define SV_INTRV_16           0x02
#define SV_NO_INTRV_16        0x03
#define SV_NO_INTRV_10        0x04
#define SV_NO_INTRV_8          0x05
#define SV_NO_INTRV_6          0x06
#define SV_EBCDIC_TO_ASCII    0x01
#define SV_G                   0x02
#define SV_NMVT                0x01
#define SV_OFF                  0x00
#define SV_ON                   0x01
#define SV_PDSTATS_SUBVECTORS 0x03
#define SV_SUBSTITUTE           0x00
#define SV_ROUND_TRIP          0x01
#define SV_USER_DEFINED        0x00
#define SV_NEW                  0x00
#define SV_OVERWRITE           0x01

/*****
/*      Return Codes from the Common Service Verbs.      */
*****/

#define SV_OK                  PFLIPI(0x0000) /* X'0000'    */
#define SV_PARAMETER_CHECK     PFLIPI(0x0001) /* X'0001'    */
#define SV_STATE_CHECK         PFLIPI(0x0002) /* X'0002'    */
#define SV_FILE_ALREADY_EXISTS PFLIPI(0x0030) /* X'0030'    */
#define SV_OUTPUT_DEVICE_FULL PFLIPI(0x0031) /* X'0031'    */
#define SV_INVALID_VERB_SEGMENT PFLIPI(0xF008) /* X'F008'    */
#define SV_UNEXPECTED_DOS_ERROR PFLIPI(0xF011) /* X'F011'    */
#define SV_COMM_SUBSYSTEM_NOT_LOADED PFLIPI(0xF012) /* X'F012'    */
#define SV_INVALID_VERB       PFLIPI(0xFFFF) /* X'FFFF'    */

#define SV_CONVERSION_ERROR    PLFLIPI(0x00000406) /* X'00000406' */
#define SV_DATA_EXCEEDS_RU_SIZE PLFLIPI(0x00000302) /* X'00000302' */
#define SV_INVALID_CHARACTER_SET PLFLIPI(0x00000402) /* X'00000402' */
#define SV_INVALID_DATA_SEGMENT PLFLIPI(0x00000006) /* X'00000006' */
#define SV_INVALID_DATA_TYPE   PLFLIPI(0x00000303) /* X'00000303' */
```



```
#define SV_INVALID_DIRECTION          PLFLIPI(0x00000401) /* X'00000401' */
#define SV_INVALID_FIRST_CHARACTER    PLFLIPI(0x00000404) /* X'00000404' */
#define SV_INVALID_MESSAGE_ACTION     PLFLIPI(0x00000621) /* X'00000621' */
#define SV_INVALID_SET                PLFLIPI(0x00000624) /* X'00000624' */
#define SV_INVALID_STORAGE_SIZE       PLFLIPI(0x00000627) /* X'00000627' */
#define SV_SSCP_PU_SESSION_NOT_ACTIVE PLFLIPI(0x00000301) /* X'00000301' */
#define SV_INVALID_NMVT_ERROR         PLFLIPI(0x00000304) /* X'00000304' */
#define SV_TABLE_ERROR                PLFLIPI(0x00000405) /* X'00000405' */
#define SV_INVALID_RESET_TRACE        PLFLIPI(0x0000062F) /* X'0000062F' */
#define SV_INVALID_CHAR_NOT_FOUND     PLFLIPI(0x00000630) /* X'00000630' */
#define SV_INVALID_SOURCE_CODE_PAGE   PLFLIPI(0x00000631) /* X'00000631' */
#define SV_INVALID_TARGET_CODE_PAGE   PLFLIPI(0x00000632) /* X'00000632' */
#define SV_TRACE_NOT_STOPPED          PLFLIPI(0x0000062A) /* X'0000062A' */
#define SV_INVALID_FILE_OPTION        PLFLIPI(0x0000062B) /* X'0000062B' */
#define SV_TRACE_BUFFER_EMPTY         PLFLIPI(0X0000062C) /* X'0000062C' */
```

Header File: acssvcc.h

B. Creating a Message File

Provided as part of the Common Service Verb API package is a utility (*commkmsgf*) with which you can produce your own message files to be used with the LOG_MESSAGE verb.

To use this utility, you must first create a text me containing the message numbers and text. This must be in the following format:

- ◆ Lines beginning with a semicolon are comment lines and are ignored.
- ◆ The three-character name which you intend to use as the file name must appear as the first line of the file, before the first message number. (It may be preceded by comment lines.)
- ◆ Each message number must consist of four decimal digits. It must be prefixed with the same three-character name which is in the first line of the me, and it must end with a colon. Message numbers must be unique (but do not have to be consecutive or in order). The message number 0000 is not allowed.
- ◆ One of the characters E, W, I, H or P may be included between the four-digit message number and the colon. This indicates the severity of the message:

E Error message

W Warning message

I Information message

H Help message

P Prompt message

This character is optional and is for information only; it is not included as part of the message file. Only uppercase letters are accepted.

- ◆ All text following the message number and colon is used as the message text.

Creating a Log Message File

! The positions within the message (if any) where the data strings supplied by the LOG_MESSAGE verb are to be inserted are indicated by %1, %2 and so on. %1 represents the first string supplied to the message, %2 to the second and so on (regardless of the order in which %1 and %2 appear in the message text).

A sample input text file is shown below.

```
; This is some header information which will be
; ignored by commkmsgf.
;
; This is the source text for the message file NEW.MSG
;
NEW
NEW0001: This is the first message, no parameters
NEW0002: This is another message
;
; The following message has some parameters
;
NEW0003: Parameter 1 = %1, parameter 2 = %2
; End of file
```

To convert this text file into a message file, you use the *commkmsgf* utility:

```
commkmsgf infile outfile
```

where infile and outfile are the input text filename and the output message filename. The output file must have the three-letter name specified at the start of the input text, with the extension msg. For example:

```
commkmsgf myfile. text /usr/lib/linkix/new.msg
```

commkmsgf will write error messages to standard output if it detects errors in the input file format.

Note that the message file created must be in the directory */usr/lib/linkix* in order for the LOG_MESSAGE verb to find it..

Index

A

- ACSSVC (C function), 1-4
- ACSSVC_C (C function), 1-4
- ACSSVC_P (C function), 1-4
- acssvcc.h*, 1-4
- ASCII to EBCDIC conversion, 2-2
- Audit/error log, 2-18, 2-22

C

- C functions
 - ACSSVC, 1-4
 - ACSSVC_C, 1-4
 - ACSSVC_P, 1-4
- C programs
 - calling convention, 1-4
 - CSV header file, 2-1
 - function compatibility with other implementations, 1-4
 - portability, 1-8, 2-1
 - reserved parameters, 1-8
 - symbolic constants, 2-1
- Character conversion, 2-2
 - conversion tables, 2-3
 - custom conversion tables, 2-3
 - creating, 2-14
- Compatibility
 - with DCA/Microsoft products, 1-2, 1-8
 - with IBM OS/2 Extended Edition, 1-2, 1-8
- Compiling and linking, 1-7
- CONVERT, 1-3, 2-2
- COPY_TRACE_TO_FILE, 1-3, 2-6
- Custom conversion tables
 - sample, 2-4

D

- DEFINE_TRACE, 1-3, 2-9
- Documentation conventions, 1-2

E

- EBCDIC to ASCII conversion, 2-2
- Environment variables
 - SNACTL, 2-13
 - SNATRC, 2-7, 2-9
- Error/audit log, 2-18
- Errors
 - logging, 2-18

G

- GET_CP_CONVERT_TABLE, 1-3, 2-14

H

- Header file, 2-1

L

- LOG_MESSAGE, 1-3, 2-18
- Logging errors, 2-18

M

- Message file
 - creating, B-1

N

- Network Management Vector
 - Transport (NMVT), 2-22, 2-23

P

PFLIPI macro, A-1
PLFLIPI macro, A-1
Portability to other environments, 1-8
Prerequisite knowledge, 1-2

log_message, 2-18
transfer_ms_data, 2-22

Verbs

calling/issuing, 1-5
summary of, 1-3

S

Signal processing, 2-25
sigpoll, 2-22, 2-25
SNACTL environment variable, 2-13
snamkmsgf, B-1
SNATBLG environment variable, 2-3
SNATRC environment variable, 2-7,
2-9
Symbolic constants, 2-1

T

Trace file(s)
clearing, 2-6
copying, 2-6
Tracing
APPC applications, 2-10
CSV applications, 2-11
disabling, 2-9
enabling, 2-9
HLLAPI applications, 2-11
LUA applications, 2-11
set up, 2-9
SNACTL environment variable, 2-
13
SNATRC environment variable, 2-9
TRANSFER_MS_DATA, 1-3, 2-22

V

VCB (verb control block)
structures, 1-4
convert 2-2
copy_trace_to_file, 2-6
define_trace, 2-9
get_cp_convert_table, 2-14

LINKix

Hardware Installation

(FIFO/SIB Board)

This guide describes the steps involved in installing and configuring Cleo's FIFO/SIB board in an IBM AT-style 386 or higher.

Caution!

Adapter boards must be protected from electrostatic discharge. Before touching the FIFO/SIB board, first touch a metal object to discharge any static electricity that may be built up in your body. Handle the board from the sides to avoid contact with any components on the board. Failure to protect the board in this way can cause degradation or failure of the product.

Technical Support

If, after reading the installation instructions, you have questions about installing and configuring the FIFO/SIB board, contact Interface Systems Technical Support between 8:30 A.M. and 5:30 P.M. (EST/EDT) at: (313) 769-5900



INTERFACE
S Y S T E M S . I N C .

© 1993 Cleo Communications
A Division of Interface Systems, Inc.

June 10, 1994

Features and Requirements

The FIFO/SIB is a synchronous communications board that, when combined with Cleo's LINKix software, provides a complete communications package for connecting to IBM (and other) mainframe systems.

Features

The FIFO/SIB features:

- ◆ Synchronous serial port (for direct attachment to a synchronous modem or modem eliminator)
- ◆ User configurable I/O port address
- ◆ User configurable IRQ (interrupt request line)
- ◆ 512-byte FIFO (first in, first out) buffers for both transmit and receive operations
- ◆ Up to 64K baud rate supported

Requirements

The FIFO/SIB board requires the following for proper installation and operation:

- ◆ An IBM compatible computer with an available 8-bit or 16-bit bus slot (a 16-bit slot is required for using extended IRQs, 9 or greater)
- ◆ An available I/O address space at 0x250, 0x260, 0x2B0, 0x2E0, 0x380, 0x3A0 or 0x3E0
- ◆ An available IRQ: 3,5,9, 10, 11, 12 or 13 (IRQs 9 or greater are only available if the FIFO/SIB is installed in a 16-bit slot). Note that IRQ 2 must also be available on your system if you plan to use IRQ 9.
- ◆ External-clocking from the attached modem or modem eliminator
- ◆ A supported UNIX environment

Installing and Configuring Multiple Boards

If you are installing multiple FIFO/SIB adapter boards, keep the following in mind:

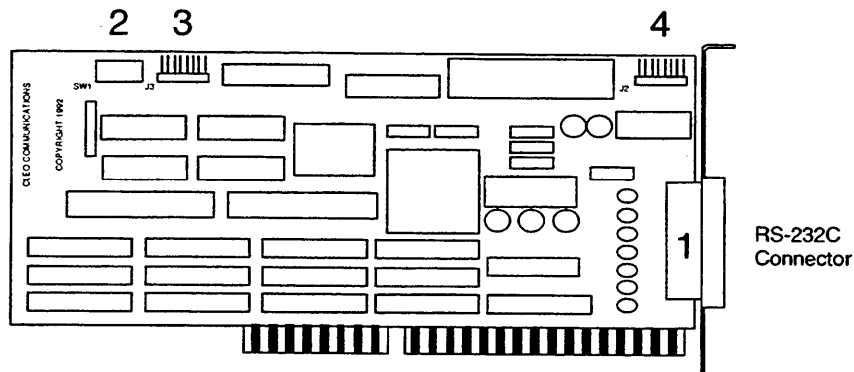
- ◆ All FIFO/SIB boards must be jumpered to use the same IRQ — the hardware configuration script enforces this requirement, automatically setting the IRQ for all boards to that specified for board one.
- ◆ Each FIFO/SIB must be jumpered to use a different I/O address — the hardware configuration script enforces this requirement, allowing you to only specify an I/O address that is not is use by another driver.
- ◆ The board number assigned during hardware configuration must be used when specifying the port number of the link (see **Configuring Links** in chapter 5 of the *Administration Guide*) — if you're unsure which I/O address is being used by a board, use the *LINKix.cfg* utility to display the current configuration.

TIP As you configure each board in a multiple board installation, make a note of the board number and configured I/O address. Then attach a label which contains this information to the face of the adapter bracket that is visible from the rear panel of the computer's chassis.

Pre-installation Configuration

TIP Before installing the hardware, we recommend that you install the LINKix software. If LINKix is already installed, run the *LINKix.cfg* utility before installing the board. The installation script, as well as *LINKix.cfg*, attempts to resolve any potential conflicts with hardware settings (IRQ, I/O address). This allows you to resolve any potential conflicts before you physically configure and install the FIFO/SIB board.

The FIFO/SIB is a synchronous communications board that can be installed in a 16-bit or 8-bit bus slot.



FIFO/SIB Board

The FIFO/SIB board is shipped with a default configuration that will work in many installations. In the figure above, the following areas (some of which can be modified to match your specific environment) have been identified:

1. The RS-232 connector. Your external synchronous modem (or modem eliminator) is connected to this male DB25 connector. At a minimum, pins 1 through 8, 15, 17 and 20 should be connected in a straight-through fashion (no crossovers). The pin assignments for this connector are documented at the end of this hardware guide.
2. The I/O switch block (SW1). The I/O address is set using switches 1 through 3 of this switch block (switch 4 is not connected). Each I/O address specifies the starting address for a 16 byte I/O buffer (e.g., 250-25F, 260-26F).

The table that follows shows how the switch settings should be set for the various I/O addresses.

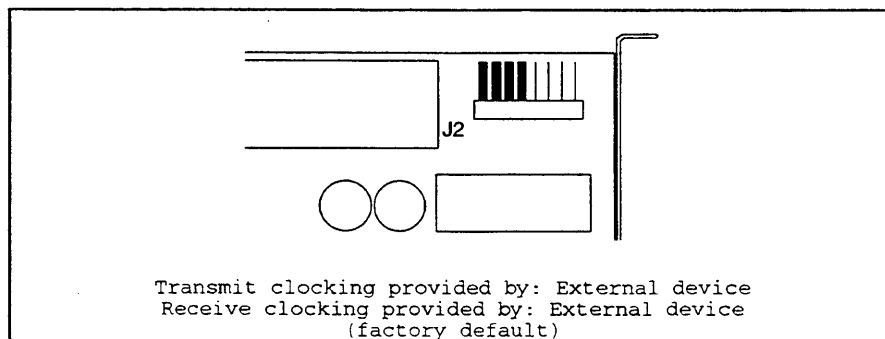
I/O address	Switch			
	1	2	3	4
250	On	On	off	X
260	off	off	On	X
2B0	On	off	On	X
2E0	off	On	On	X
(default) 380	off	off	off	X
3A0	On	off	off	X
3E0	off	On	off	X
Disabled	On	On	On	X

No other board in the PC should use the same address as the FIFO/SIB board. Any conflict will prevent the FIFO/SIB and LINKix from operating properly.

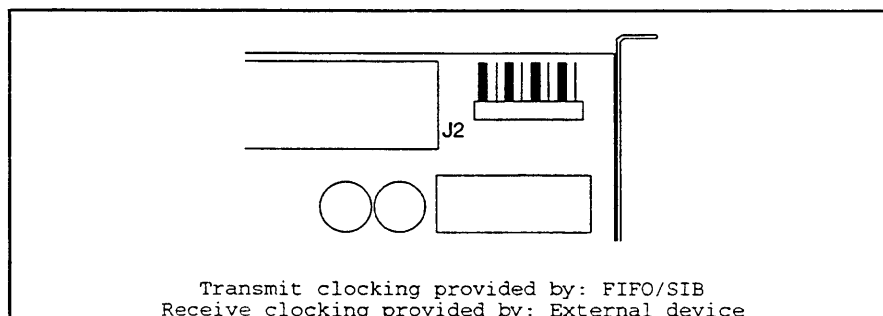
3. The IRQ jumper block (J3). The interrupt request line (IRQ) setting is determined by the placement of the jumper within this group of pins. The IRQ can be set to a value of 3, 5, 9, 10, 11, 12 or 13. The default is IRQ 3. The possible IRQ settings are shown in the illustration that follows.

In order to use the extended IRQs (greater than 9) the FIFO/SIB must be installed in a 16-bit bus slot. Also, if you wish to use IRQ 9, make sure that IRQ 2 is also unused on your system.

4. The clocking jumper (J2). This jumper, which contains 8 pairs of jumper pins numbered 1 through 16, determines which device will provide the clocking for the transmit and receive lines (lines 15 and 17) of the FIFO/SIB RS-232 interface. By placing the jumpers over specific groups of pins, the clocking can be specified as follows:



This is the factory default setting. This setting will work in almost all situations. The maximum cable length that's supported with this setting, when operating at 64K baud, is 15 feet.



To successfully use longer cable (up to 50 feet) at 64K baud, the transmit clocking (pin 15) must be supplied by the FIFO/SIB and the receive clocking (pin 17) must be supplied by the external modem or modem eliminator. You also need to edit a line in `/ect/conf/pack.d/hmod/space. c`, changing it from:

```
int hmod_linespeed = 0;
```

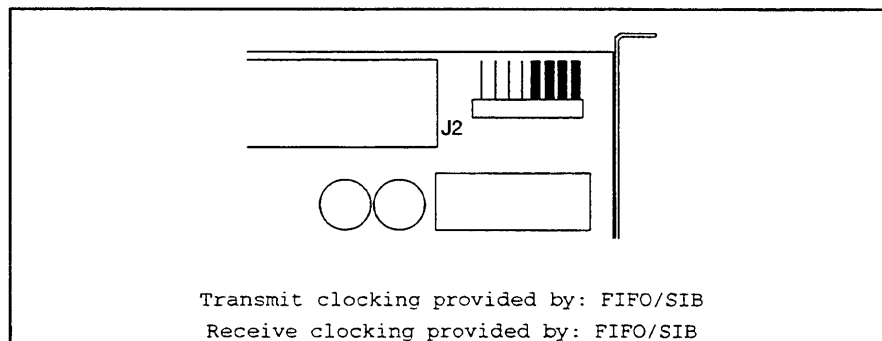
to

```
int hmod_linespeed = <baudrate/100>;
```

For example, 9600 baud would be:

```
int hmod_linespeed = 96;
```

If you think you need to use this setting, call Cleo Technical Support for complete details on this setup.



This setting is only needed if the external modem or modem eliminator cannot provide clocking. In order to use this setting, you also need to edit a line in */ect/conf/pack.d/hmod/space.c*, changing it from:

```
int hmod_linespeed = 0;
```

to

```
int hmod_linespeed = <baudrate/100>;
```

For example, 9600 baud would be:

```
int hmod_linespeed = 96;
```

If you think you need to use this setting, call Cleo Technical Support for complete details on this setup.

Installing the FIFO/SIB Board

Caution: Before you remove the cover of your PC's system unit, be sure to turn the power off for all external accessories and the PC itself. Disconnect all AC power sources by unplugging the computer and any attached accessories. Failure to follow these procedures can cause dangerous voltages to be present in the PC chassis that could cause personal injury and/or damage to your computer or FIFO/SIB board and void your warranty. Even when the ON/OFF switch of your computer is set OFF, dangerous voltages can be present unless all power cables are disconnected.

After removing all power connectors from your PC, you're ready to install the FIFO/SIB board.

1. Following the instructions in your PC's manual, remove the system unit cover.
2. Locate an open full-or half-length expansion slot. Remove the screw that holds the expansion slot cover in place and take out the expansion slot cover. Retain the screw for later use.
3. Align the board in the card guide and carefully insert the FIFO/SIB board into the expansion slot.
4. Secure the board in place by replacing the retaining screw that was removed in step 2.
5. Replace the system unit cover and reattach all the power cables.
6. Attach an external modem (or modem eliminator) to the DB25 connector on the FIFO/SIB board. The cable must have (at a minimum) pins 1 through 8, 15, 17 and 20 connected in a straight-through fashion (no crossovers).

The hardware installation is now complete.

Pin Assignments

The pin assignments for the RS-232 interface on the FIFO/SIB areas follows:

Pin	Signal Direction	Description	
1	SGND	Shield Ground	
2	TXD	Transmit data	output
3	RXD	Receive data	input
4	RTS	Request to send	output
5	CTS	Clear to send	input
6	DSR	Data set ready	input
7	GND	Ground	
8	DCD	Data carrier detect	input
9	+12V	+12 volt power source	
10	-12V	-12 volt power source	
15	TXC	Transmit clocking	input output
17	RXC	Receive clocking	input output
20	DTR	Data terminal ready	output
22	RI	Ring indicator	input
26	SHELL	Shield ground to DB-25 shell	

LINKix

Hardware Diagnostics

(FIFO/SIB Board)

This guide describes the hardware diagnostics utility that's provided with CLEO's FIFO/SIB board and explains how to use the program.

Technical Support

If, after reading these usage instructions, you have questions about using the FIFO/SIB diagnostics utility, contact Interface Systems Technical Support between 8:30 A.M. and 5:30 P.M. (EST/EDT) at: (313) 769-5900

© 1994 CLEO Communications
A Division of Interface Systems, Inc.



June 10, 1994

Usage Instructions

The FIFO/SIB hardware diagnostics utility operates as a separate, standalone program. You should not run these diagnostics on a FIFO/SIB board that is currently being used by LINKix.

The FIFO/SIB hardware diagnostic utility is executed with the command:

```
fifodiag [-c] [-h] [-p ports] [-t tests] [-v]
```

where one or more of the following options can be specified.

-c	This option causes the specified tests to be performed continuously. This option can be useful for diagnosing intermittent problems.
-h	This option prints a usage message.
-p <i>ports</i>	This option allows you to specify which port the tests should be carried out on. By default only port 1 is tested. Multiple ports can be specified by separating port numbers with a comma (e.g., 1,2,3,4). The order in which the ports are listed determines the order in which the ports are tested.
-t <i>tests</i>	This option specifies which of the four available diagnostic tests to perform. By default, tests one through three are run. Multiple tests can be specified by separating test numbers with a comma (e.g., 1,2). Up to eight tests can be specified at one time. The order in which the tests are listed determines the order in which the tests are performed.
- v	This option turns on verbose reporting, which prints out useful information. By default, <i>fifodiag</i> performs its tests and exits with an appropriate UNIX exit code. When using this option, however, informational messages are directed to the standard output device (stdout) and all diagnostic error messages are written to standard error (stderr). This allows the two types of output to be redirected as needed. This option can be particularly useful when running a test in continuous mode.

Although you can use verbose reporting to view return codes on screen, we recommend that you use the UNIX command:

```
echo $ ?
```

immediately after executing *fifodiag* to display the return code.

- e This advanced option causes *fifodiag* to perform its tests using an external clocking source.

Testing

This diagnostic utility tests all major components of the FIFO/SIB hardware. It tests areas that may be in conflict with other installed hardware adapters and tests for proper operation of the I/O address and IRQ.

Tests

The FIFO/SIB diagnostic utility, *fifodiag*, offers four levels of testing:

1. I/O Test

The purpose of this test is to determine if all I/O operations supported by the FIFO/SIB are working properly. The first portion of this test simply determines if the board is present and accessible. The second portion of this test determines if the IRQ block is strapped correctly. The third and final portion of this test exercises the FIFO transmit buffer.

2. Bus test

This test verifies that the FIFO/SIB can operate in this specific system. Using worst case back-to-back I/O instructions, this test ensures that bus timings for the tested PC are compatible with the FIFO/SIB board. If the FIFO/SIB fails this test, the bus timings in the PC are inappropriate (too fast) for the FIFO/SIB. On the other hand, when the FIFO/SIB passes this test, you can be relatively secure in assuming that the board will operate in all computers of the same brand and configuration.

3. Internal loopback test

This test verifies the FIFO/SIB's ability to read and write data to and from the FIFO buffers using internal loopback options. This test can be performed with an internal or external (-e) clocking source.

4. External loopback test

This test verifies the operation of the various interface signals (RTS, CTS, DCD, DTR, DSR), the interface itself, various components that drive the interface, the modem cable and possibly the modem. This test requires an external diagnostic loopback connector (see Diagnostic Loopback Connector) or a modem with digital loopback capability. This test can be performed with an internal or external (-e) clocking source.

Testing Scenarios

When attempting to diagnose problems with LINKix on the FIFO/SIB board, we recommend that you proceed with the following test scenarios.

A. Initial Test

- ◆ Disconnect the FIFO/SIB from the modem or modem eliminator (especially on a multi-drop line).
- ◆ Run *fifodiag* with default settings: Tests 1 through 3, internal loopback with internal clocking.
- ◆ Refer to the next section, Diagnosing Problems, for information on interpreting return codes for any failures. If the FIFO/SIB passes this test, continue with the next test scenario.

B. Advanced Test #1 (Internal clocking)

- ◆ Attach the loopback plug or a modem that has been placed in loopback mode.
- ◆ Run tests 1 through 4 (-t 1,2,3,4).
- ◆ Refer to the next section, Diagnosing Problems, for information on interpreting return codes for any failures. If the FIFO/SIB passed this test using a loopback plug...
- ◆ Place the loopback plug at the modem end of the cable and rerun the tests. A failure at this point indicates a bad cable. If the FIFO/SIB passed this test, continue with the final test scenario.

C. Advanced Test #2 (External clocking)

- ◆ Attach the FIFO/SIB to a modem that has been placed in loopback mode or to a modem eliminator and make sure that the external device has been configured to provide clocking.
- ◆ Run tests 1 through 3, specifying that an external clock source should be used. *If and only if you are using a modem placed in loopback mode should you run test 4 with the -e option.* Running test 4 with a device that's attached to an active multi-drop line will cause test frames to be transmitted, intruding on the normal operation of the line.

- ◆ Refer to the next section, Diagnosing Problems, for information on interpreting return codes for any failures. If the FIFO/SIB passed this test, your problem resides elsewhere.

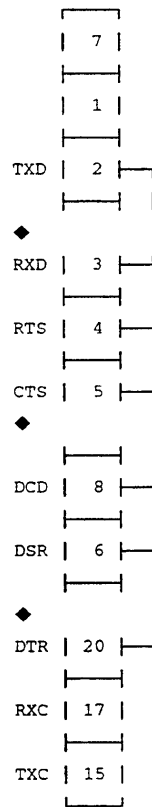
Diagnosing Problems

When testing the FIFO/SIB hardware with *fifodiag*, keep in mind that failure of one or more tests indicates a hardware conflict or a hardware failure. Where possible, we indicate when a hardware failure is the most likely reason for the failure. Please note, however, that a final determination of hardware failure must be ascertained by a CLEO support technician, using a variety of problem solving techniques.

Test	Return Code	Diagnosis/Action
	16	1. invalid command line option specified. Check the syntax and reenter the command.
	17	1. The specified port is not closed. Check to make sure that another copy of <i>fifodiag</i> , LINKix, or some other application is not using the board.
	18	1. Program execution was prematurely interrupted.
1	1	1. I/O configuration mismatch. The I/O setting on the board does not match the address configured in LINKix. Make sure the board setting and the configuration match. 2. I/O address conflict. The I/O address used by the FIFO/SIB is being used by another device in the system. Try a different I/O address with the FIFO/SIB. 3. Hardware failure.
1	2	1. IRQ strapped incorrectly. The IRQ jumper block does not match the configured IRQ. Make sure the board setting and the configuration match. 2. I/O address conflict. The I/O address used by the FIFO/SIB is being used by another device in the system. Try a different I/O address with the FIFO/SIB. 3. Hardware failure.
1	3 4 5 6	1. I/O address conflict. The I/O address used by the FIFO/SIB is being used by another device in the system. Try a different I/O address with the FIFO/SIB. 2. Hardware failure.
2	7	1. System bus speed incompatible with FIFO/SIB. 2. Hardware failure.

Loopback Connector

The following diagram shows the necessary cabling and pin connections that are required for an external loopback plug. This plug, or a modem that can be placed in loopback mode, is required for running *fifodiag* test 4. The following pins should be connected on a DB25S (female) connector.



Test	Return Code	Diagnosis/Action (Internal Clocking)	Diagnosis/Action (External Clocking, -e)
3	11	1. Specified too high baud rate with -b option. 2. Hardware failure.	1.
3	12	1. Hardware failure.	1.
3	13	1. Hardware failure.	1.
3	14	1. Hardware failure.	1.
3	15	1. IRQ conflict. The IRQ setting on the FIFO/SIB is being used by another device in the system. Try a different IRQ with the FIFO/SIB.	1.
4	8	1. Loopback plug not installed or defective. 2. Modem not in loopback mode. 3. Bad cable to modem.	1. Modem not in loopback mode. 2.
4	9	1. Loopback plug not installed or defective. 2. Modem not in loopback mode. 3. Bad cable to modem.	1. Modem not in loopback mode. 2.
4	10	1. Loopback plug not installed or defective. 2. Modem not in loopback mode. 3. Bad cable to modem.	1. Modem not in loopback mode. 2.
4	11	1. Loopback plug defective. 2. Modem not in loopback mode. 3. Bad cable to modem.	1.
4	12	1. Loopback plug defective. 2. Modem not in loopback mode. 3. Bad cable to modem.	1.
4	13	1. Loopback plug defective. 2. Modem not in loopback mode. 3. Bad cable to modem.	1.
4	14	1. Hardware failure.	1.
4	15	1. IRQ conflict. The IRQ setting on the FIFO/SIB is being used by another device in the system. Try a different IRQ with the FIFO/SIB.	1.

LINKix 3270

32.70 User's Guide

June 1994

RESTRICTED RIGHTS

Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (C)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013. CLEO Communications.

LIMITED RIGHTS

Limited rights shall be effective indefinitely and are not subject to expiration as set forth in paragraph (3) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013.

Interface Systems, Inc., 5855 Interface Drive, Ann Arbor MI 48103



June 10, 1994

3270 User's Guide

June 1994

CLEO Communications reserves the right to, without notice, modify or revise all or part of this document and/or change product features or specifications and shall not be responsible for any loss, cost or damage, including consequential damage, caused by reliance on these materials.

This document may not be reproduced, stored in a retrieval system or transmitted, in whole or in part, in any form or by any means (electronic, mechanical, photocopied or otherwise) without the prior written permission of CLEO Communications.

© 1994 CLEO Communications – All rights reserved.

Portions:

© 1989–1992 Data Connection Limited

© 1989, 1990 Digital Communications Associates, Inc.

© 1989, 1990 Microsoft Corporation

Table of Contents

Before You Begin

The Documentation	x
Administration Guide	x
3270 User's Guide	x
HLLAPI Programmer's Guide	x
Common Service Verbs Programmer's Guide	x
On-Line Help	xi
Documentation Conventions	xii
What Is LINKix 3270?	xiii
Technologies and Guidelines	xiv
Systems Network Architecture	xiv
Systems Application Architecture	xiv
Common User Access (CUA)	xiv
Common Communications Support (CCS)	x v

1. The User Interface

Terminal Requirement	1-2
Required Functions	1-2
Recommended Functions	1-2
Keyboard Mapping	1-3
Menu Bar	1-5
Choosing a Menu from the Menu Bar	1-5
Menus	1-7
Choosing a Menu Option	1-8
Dialogs	1-9
Pushbuttons	1-10
Choosing a Pushbutton	1-10
Check Boxes	1-12
Choosing a Check Box	1-12

Table of Contents

Radio Groups	1-13
Choosing a Radio Button	1-13
Edit Boxes	1-14
Typing Information in an Edit Box	1-15
List Boxes	1-15
Selecting an Item from the List Box	1-16
Elevator Bar	1-16
List Boxes with Associated Edit Boxes	1-17
Dual List Boxes	1-18
Moving Items Between Dual List Boxes	1-18
Accelerator Keys	1-19
Other Interface Features	1-20
Disabled (Grayed) Items	1-20
On-line Help Screens	1-20
Message Boxes	1-22
Error	1-22
Warning	1-22
Information	1-23
More Info	1-23
Screen Redrawing	1-24
Fast Exit from Menu Interface Programs	1-24
 2. 3270 Emulation	
Overview of Emulation Features	2 - 2
Keyboard Mappings	2 - 2
3270 Sessions	2 - 2
Model 2-5 Display Support	2 - 3
International Language Support	2 - 3
Background Operation	2 - 3
File Transfer	2 - 4
Printer Emulation	2 - 4
Temporary Exit From 3270 Emulation	2 - 4
Type-ahead During 3270 Emulation	2 - 5
Keystroke Recording and Replay	2 - 5

Table of Contents

Menu Interface	2 - 5
The File Menu	2 - 5
The Customize Menu	2 - 6
The Control Menu	2 - 6
HLLAPI Support	2 - 6
Starting the 3270 Emulation Program	2 - 7
Running 3270 Emulation in the Background	2-10
Stopping Background 3270 Emulation	2-11
Session Screen and Main Screen	2-12
Switching Between 3270 Sessions and the Main Screen	2-12
Temporary Exit to the UNIX Command Prompt	2-13
Communicating with the Host	2-13
Ending Emulation	2-14
3270 Operations	2-15
Using the Keyboard	2-15
Controlling a Terminal's Cursor	2-15
Using Other Keys	2-16
3270 Status Line	2-16
Switching Between 3270 Sessions	2-18
3270 Data Fields	2-19
Attribute Characters	2-19
EAB Characters	2-19
Keystroke Recording and Replay	2-20
Valid Keystrokes	2-20
Recording a Keystroke Sequence	2-21
Pausing Within a Sequence	2-22
Replaying a Recorded Sequence	2-23
Clearing a Recorded Sequence	2-24
Saving Keystroke Sequences	2-24
Multiple Sessions	2-25
Running HLLAPI Programs	2-26
Automating 3270 Emulation Startup	2-27
3270 User's Guide	v

Table of Contents

Fast Path to 3270 Terminal Emulation	2 - 28

3. Controlling 3270 Emulation

Main Screen and Menus	3 - 2
Main Screen Menus	3 - 2
The Control Menu	3 - 3
Moving to Terminal Emulation	3 - 3
Controlling Display Sessions	3 - 4
Enabling a Display Session	3 - 6
Making a Session Current	3 - 7
Moving Between Active 3270 Sessions	3 - 7
Disabling a Session	3 - 8
Remapping a Display Session	3 - 8
Controlling Printer Sessions	3-10
Enabling a Printer Session	3-12
Remapping a Printer Session	3-12
Maximum Print Position for LU1 Printer Sessions	3-13
Transferring Files	3-14
Viewing Host Response Times	3-15
Sending NetView User Alerts	3-17
Exiting 3270 Emulation	3-20

4. Transferring Files

Write Structured Fields (WSF) Support	4 - 2
Preparing to Transfer Files	4 - 3
Available Sessions	4 - 3
Unavailable Sessions	4 - 4
Transferring Files Using Menu Options	4 - 5
Setting Host File Transfer Parameters	4 - 5
Transferring Files	4-10
MVS/TSO Host	4-10
VM/CMS Host	4-12

Table of Contents

CICS Host	4-14
Command Line File Transfers	4-16
MVS/TSO	4-17
WCMS	4-17
CICS	4-17
Command Line File Transfer Parameters	4-17
Command Line File Transfer Options	4-19
Sample File Transfer Commands	4-22
Monitoring File Transfer Status	4-23
Stopping a File Transfer	4-26
Finishing File Transfer	4-27
File Transfer Successful	4-27
File Transfer Unsuccessful	4-27
Fast Path to File Transfer	4-28
5. Customizing 3270 Emulation	
The Style File	5 - 2
Search Path for Style Files	5 - 2
Main Screen Menus	5 - 3
File Menu	5 - 4
Customize Menu	5 - 5
Opening, Creating and Saving Style Files	5 - 6
Opening a Style File	5 - 6
Creating a New Style File	5 - 9
Saving a Style File	5 - 9
Saving a Style File Under a Different Name	5-10
Customizing Display Sessions	5-11
Customizing Printer Sessions	5-15
Customizing Display Colors	5-18
Use Default Color Mapping	5-21
Copy Colors From Another Session	5-22

Table of Contents

Customizing Colors	5-23
Choosing a Host Language	5-25
Customizing Key Definitions	5-28
Redefining Keys	5-29
Undefining Keys	5-32
Choosing Default Keyboard Mapping	5-33
Customizing Miscellaneous Options	5-35
Custom Translation Tables	5-37
Custom Translation Table Utilities	5-38
<i>comtrdmp</i>	5-38
<i>comtrcnv</i>	5-40
6. Solving Problems	
A. Default Keyboard Mappings	
B. Status Line Information	
C. File Transfer Messages	
D. HLLAPI Terminal Emulator	
E. Glossary	

Before You Begin

Before you get started, take a moment to read this section. It provides an overview of the following LINKix 3270 topics:

- ◆ The Documentation
- ◆ Documentation Conventions
- ◆ Common Abbreviations
- ◆ What is LINKix 3270?
- ◆ Technologies and Guidelines

This manual assumes that the LINKix 3270 package has already been installed, along with the appropriate hardware. If this is not the case, see your system administrator or refer to the *Quick Start* booklet or the *Administration Guide*.

The LINKix 3270 package that's installed on your machine may not support all options that are discussed in this manual. When the functionality of a feature is either limited to or not supported by a specific product or operating system, the documentation indicates this fact.

This documentation also assumes that the user has some knowledge of the UNIX operating system (login steps, terrninfo, etc.).

The Documentation

Provided with LINKix 3270 is a comprehensive library of documentation that consists of the following manuals and an on-line help system.

Administration Guide

The *Administration Guide* explains the procedures for installing and configuring the software, suggested troubleshooting of installation problems, setting up system files, as well as monitoring and tracing program execution. A section on diagnostics explains audit and error log files and possible solutions to problems. There's also a section on NetView alerts.

3270 User's Guide

The *3270 User's Guide* describes product features, architecture, connectivity, configuration options and the user interface of LINKix 3270. The section on 3270 emulation covers how to start and stop 3270 emulation, transferring files, using customization features such as remapping your keyboard and display colors, interpreting status-line information, sending NetView user alerts and viewing response times. An appendix in this manual contains a glossary of terms used throughout the manuals.

HLLAPI Programmer's Guide

The *HLLAPI Programmer's Guide* contains the information you need to write application programs using the High-Level Language Application Program Interface (HLLAPI). It includes conceptual and detailed reference information, as well as sample application program source code.

Common Service Verbs Programmer's Guide

The *Common Service Verbs Programmer's Guide* contains complete reference information on CLEO's implementation of the Common Service Verbs (CSV) programmer's interface. It includes conceptual and detailed reference information.

On-Line Help

On-line help is available whenever the menu interface is active. The help screens are both descriptive and instructive; instead of merely stating the choices available at a particular point in the program, they guide you in making the correct choice.

Pressing **F1** at any point in the menu interface initiates the help system. A help screen will appear with information about the options that are available at this point. While a help screen is displayed, you can press **F1** again to receive information about the help system. This explains the type of information you will receive and how to move around in the help screen system.

For more information about the operation of the on-line help system, see chapter 1 of this manual.

Documentation Conventions

The following documentation conventions are used in this manual:

<i>italics</i>	Italicized words are used for emphasis, to identify the titles of manuals and to indicate file and directory names.
bold	Bold type is used for cross-references to other chapters within the manual, or to other sections within a chapter.
KEYNAME	A key name in this format, for example ENTER, refers to a specific key on your terminal's keyboard. Some keyboard commands require that you press one key and hold it down while you press one or more other keys. These are also called combination keystrokes. To indicate this, key names are shown together with + signs between them (for example, CTRL + D). While holding down the first key, press the other key(s).
KEYNAME	A key name in uppercase letters, for example ENTER, refers to a 3270 key. Appendix A lists the default UNIX terminal keys used to represent these 3270 keys. If you remap your keyboard, you can always find the UNIX terminal keystroke used to represent a particular 3270 key by using the View and Define Key Definitions dialog.
0xnn	A number preceded by 0x (for example 0x20) indicates a hexadecimal value.

What Is LINKix 3270?

LINKix 3270 is a software product that permits a UNIX computer to communicate with other computers via an SNA network. In addition, it provides complete file transfer capabilities, as well as many other features.

The 3270 emulation program allows your UNIX terminal to emulate an IBM 3270 series terminal to access host 3270 applications. LINKix 3270 offers the following 3270 emulation features:

- ◆ 3278/79 terminal emulation
- ◆ 3287 printer emulation
- ◆ Complete Model 2 through 5 full-screen support
- ◆ Complete extended attribute byte (EAB) support
- ◆ Color remapping
- ◆ Support for international language versions of EBCDIC at the host
- ◆ Keyboard remapping
- ◆ File transfer that's compatible with IBM's IND\$FILE host file transfer program
- ◆ Response Time Monitor (RTM)-a feature that tracks host response times
- ◆ Communication with IBM's NetView through 3270 user alerts

Support for color remapping, EAB, international language operation and screen models other than model 2 is dependent on your UNIX computer and its terminals. See the **Terminal Requirements** section in the *Administration Guide* for more information.

Technologies and Guidelines

LINKix 3270 uses the following technologies and guidelines for IBM-compatible PC communications:

- ◆ Systems Network Architecture (SNA)
- ◆ Systems Application Architecture (SAA)

Systems Network Architecture

LINKix 3270 adheres to IBM's Systems Network Architecture (SNA). SNA is a set of rules that brings uniformity to data communications systems and the ways they interact. These rules define functions that must be performed by communications software for information to be transferred from one computer to another, error-free and in a form that is usable by the receiving computer.

Systems Application Architecture

Systems Application Architecture (SAA) is another set of guidelines created by IBM. SAA helps developers standardize applications. These standard applications can function consistently in different operating environments, with minimal program modification and user retraining.

LINKix 3270 complies with two of the major elements of SAA-Common User Access (CUA) and Common Communications Support (CCS).

Common User Access (CUA)

CUA defines how user interfaces look and feel and how the user can enter and select information. LINKix 3270 complies with CUA in its interface design.

Common Communications Support (CCS)

CCS specifies how to make connections between various programs, systems, communications networks and devices. LINK.ix 3270 supports the following CCS session services:

- ◆ 3270 data stream
- ◆ Data link controls for SDLC
- ◆ Node type 2.1, when APPC support is provided with the product.
- ◆ APPC LU 6.2, when APPC support is provided with the product.
- ◆ Data link controls for X.25 (QLLC), when X.25 support is provided with the product.

Before You Begin

1. The User Interface

The user interface of LINKix 3270 complies with the SAA guidelines of CUA (Common User Access). This chapter describes how to use the interface and its components. Specifically, the following topics are covered:

- ◆ Terminal requirements
- ◆ Menu bars
- ◆ Menus
- ◆ Dialogs
- ◆ Disabled (grayed) items
- ◆ Help screens
- ◆ Screen redrawing

The example screens shown in this chapter are taken from the 3270 emulation program, except where indicated. Note that features such as highlighted characters and reverse video are not always shown in these examples.

Terminal Requirements

In order to use the LINKix 3270 menu interface, your terminal should have certain functions defined in the UNIX computer's terminfo database. These functions fall into two categories:

- ◆ **Required.** Only terminals with these functions can display the menu interfaces. (Terminals without these functions can be used, however, for transaction programs operating under one of the APIs provided with LINKix 3270.)
- ◆ **Recommended.** These functions make the screen interfaces easier to use, but can be used without them.

Required Functions

Screen Size

The size defined in terminfo must be at least 80 columns by 24 rows.

Cursor Addressability

Menus and dialogs cannot be drawn without this capability.

Alternate Character Mode

These characters are used to draw the menus and dialogs. If alternate character mode is not supported, the boxes do not appear correctly on the screen.

F1 Function Key

This key is used to invoke the on-line help facility.

Up and Down Arrow Keys

These keys are used to move within list boxes and to select radio buttons.

Recommended Functions

F2 . . . F12 Function Keys

These function keys are used in the 3270 emulation program to represent the 3270 keys PF2 to PF12; some of them are also used as accelerators in the Configuration program and 3270 Control Interface (an accelerator key provides a short cut to

certain dialogs or functions). If you do not have these keys, you can remap the 3270 keys to different keystrokes; the accelerator keys cannot be remapped, but dialogs or functions can always be reached using the main screen menus.

LEFT and RIGHT Arrow Keys

These keys are used for moving within edit boxes and for selecting radio buttons.

PAGE UP and PAGE DOWN

These keys are used to move up and down within list boxes.

HOME and END

These keys are used to move to the first or last entry in a list box. The **UP** and **Down** arrow keys or **PAGEUP** and **PAGEDOWN** can be used instead, but are slower.

BACKTAB

This key is useful for moving backward through the sequence of items in a dialog. If it is not available, the keystroke **CTRL + B** can be used instead. (Note that these keystrokes are equivalent only in the menu interface. In 3270 emulation, the default key mapping for the 3270 BACKTAB key is **BACKTAB**; you will need to remap this to **CTRL + B**, or any other suitable keystroke, if you do not have the **BACKTAB** key.)

INSERT, DELETE and BACKSPACE

The **INSERT** and **DELETE** keys are used in the 3270 emulation program as defaults for the 3270 INSERT and DELETE keys; if they are not available, the 3270 keys can be remapped to different keystrokes. **DELETE** and **BACKSPACE** are useful for modifying information in an edit box.

Color and Attribute Support

While LINKix 3270 can be run without color or attribute support, it can make use of these features if your terminal supports them. This applies particularly to the 3270 emulation program when communicating with host programs which support color and highlighting attributes.

Keyboard Mapping

The 3270 emulation program's default (the mapping between keystrokes on your terminal and the 3270 keys they represent) assumes that your terminal's keyboard has all the keys in both the Required and Recommended lists. If your terminal does not

The User Interface

have suitably labelled keys, it's still possible to use LINKix 3270 by remapping 3270 keys to a different keystroke.

Extra keys on your terminal must also be included in the terminfo database if you want to use them in the 3270 emulation program to represent 3270 keys. For example, if you have the function keys **F13 - F24**, you'll probably want to include them in terminfo and map the 3270 keys PF13 - PF24 to these keys. For more information on this procedure, see the *Administration Guide*.

Menu Bar

The menu bar is the starting point for LINKix 3270's customization interface. After you start the 3270 emulation program, you display the main customization screen with the menu bar at the top by pressing the ACTIONS key (**CTRL + U**).

The example below shows the 3270 emulation program's customization screen and menu bar.

```
File Customize Control                               Fl=Help
LINKix 3270 Customization 3.0
Style file : /usr/lib/linkix/com3270.stu

Copyright (C) 1993 CLEO Communications

Press the Return key to select a menu, or the Esc key to enter emulation
```

The menu bar contains a number of menu names, each of which has one letter (the accelerator key) highlighted. One of the menu names may appear in reverse video, indicating that it's currently selected.

At the top right of the screen, Fl=Help is displayed as a reminder that help screens are always available by pressing **F1**.

Choosing a Menu from the Menu Bar

To choose a particular menu from the, perform the following steps.

- ◆ Press the highlighted letter (accelerator key) from the name of the menu you want to open.

The User Interface

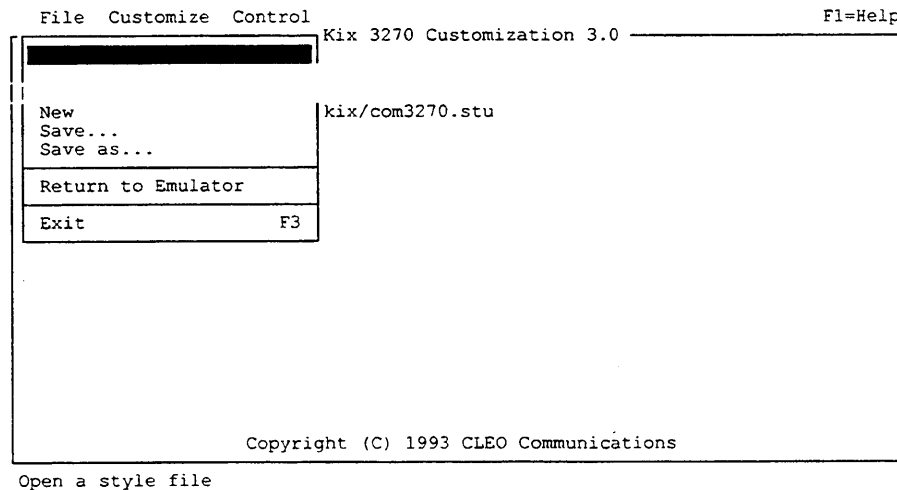
- ◆ Alternatively, press **ENTER** to activate the reverse-video selector, which will highlight the first menu name. Use the **RIGHT** and **LEFT** cursor keys to select the name of the menu you want, then use either the **UP** or **DOWN** cursor keys or **ENTER** to pull down the menu.

The menu appears below the name on the menu bar.

Menus

A menu bar contains a number of menus (also called pulldowns or pulldown menus).

When you select an option from a menu bar, the menu is displayed. The following is an example of a menu.



Each menu contains a number of menu options (also called menu items) which indicate the functions you can access from the menu. One menu option appears in reverse video, the currently selected option. The bottom line of the screen displays information on the currently selected option.

Note that one or more options in the menu maybe disabled (grayed) and cannot currently be selected (for more information on how these items operate, see **Disabled (Grayed) Items** later in this chapter).

The User Interface

Choosing a Menu Option

To choose a particular option from a, do one of the following:

- ◆ Use the **UP** and **DOWN** cursor keys to move to the menu option you want (watch the reverse-video selector), then press **ENTER**.
- ◆ Press the accelerator key (the highlighted letter) of the menu option you want.
- ◆ Press the accelerator key direct accelerator key (this key or key combination is to the right of the menu option). Direct accelerator keys are available for the most commonly used menu options, for example the Exit option in the File menu uses the **F3** key as a direct accelerator key. Direct accelerator keys can be used from the main screen of the program, without first having to select the appropriate menu.

If a menu option is followed by an ellipsis (...), a dialog is displayed when you choose the menu option. For example, choosing Open from the File menu of the 3270 emulation program causes a dialog to appear that prompts you for the name of an existing 3270 style file to open.

If a menu option is not followed by an ellipsis, an action is performed when you choose the menu option. For example, when you choose New from the File menu of the 3270 emulation program, a new 3270 style file is created.

Dialogs

LINKix3270 provides dialogs for entering responses making choices. The example below shows a typical dialog from the 3270 emulation program.

The dialog box is titled "Choose Session to Customize" and "Customize Display Parameters for Session 1". It has a menu bar with "File", "Customize", and "Control". A "F1=Help" label is in the top right corner. The dialog is divided into two main sections. The left section contains fields for "Long name" (value: [SESS01..]), "Session ID" (value: [A]), and "Description" (value: [.....]). The right section contains a "Model Type:" list with options (o) 2 (24x80), () 3 (32x80), () 4 (43x80), and () 5 (27x132). Below this is a "Local copy done to:" section with options (o) Real printer and () Print to File. To the right of this is a "Maximum structured field size (kB)" field with value [2] and an "[] Append (not overwrite) file" checkbox. At the bottom is a "Printer or file name" field with value [.....]. The dialog ends with "< OK > <Cancel>" buttons.

```

File  Customize  Control                                     F1=Help
-----
Choose Session to Customize
Customize Display Parameters for Session 1

Style
[ ] Session enabled at initialization      Model Type:
Long name . . [SESS01..]                  (o) 2 (24x80)
Session ID . . [A]                        ( ) 3 (32x80)
                                           ( ) 4 (43x80)
                                           ( ) 5 (27x132)

Description . [.....]

Local copy done to:      Maximum structured
                          field size (kB) . . . . [2]
(o) Real printer        [ ] Append (not overwrite) file
( ) Print to File

Printer or file name . . [.....]

< OK > <Cancel>

```

Dialogs contain one or more fields (areas where information is entered or choices are made). The fields in dialogs may be pushbuttons, check boxes, radio groups, edit boxes, list boxes and dual list boxes.

The **TAB** key (or **CTRL + I**) moves you from one field in a dialog to the next; **BACKTAB** (or **CTRL + B**) moves you from the current field to the previous field. In some cases you can move to another field by using accelerator keys; the use of this feature depends on the type of field.

The cursor movement keys do not move you through the fields in the dialog, but rather perform specific actions that depend on the type of field the cursor is in. The **UP** and **DOWN** arrow keys allow you to move within list boxes and to select radio buttons. The **LEFT** and **RIGHT** arrow keys are used to move within edit boxes and for selecting radio buttons.

The User Interface

Pushbuttons

Pushbuttons are controls that when chosen cause an action to occur. In the example below, <Zoom> and <Exit> are pushbuttons.

Session Number	Long Name	LU/Pool Name
1		COLPOOL
2	SESS02	COLPOOL

< Zoom > < Exit >

Copyright (C) 1993 CLEO Communications

Pushbutton names are enclosed in angle brackets. They're always displayed at the bottom of dialogs, with one exception: <Move>, which appears with Dual list boxes.

Choosing a Pushbutton

To choose a, do one of the following:

- ◆ Use the **TAB** key to move the cursor to the pushbutton (this highlights the angle brackets of the pushbutton) and press **ENTER**.
- ◆ Use the accelerator key of the pushbutton.

On each dialog, there's always a default pushbutton (indicated with highlighted angle brackets). Pressing the **ENTER** key anywhere in the dialog chooses this pushbutton.

Pushbuttons which appear in the lower right-hand corner of a dialog are used to end the dialog. These pushbuttons do not have an accelerator key shown, although the

ESC key can be used as an accelerator key for two of them. There's always one or more of the following:

<OK>

Choosing <OK> indicates that you've finished making changes to the information in the dialog and you want the program to accept the information. The information you've entered is checked. If it's correct, the dialog is ended. If any of the information is incorrect, LINKix 3270 displays a warning or error message and returns you to the dialog to correct the information or cancel the input.

<Cancel>

Choosing <Cancel> abandons any changes you've made to the information in this dialog and ends the dialog. The **ESC** key can be used as an accelerator key for this pushbutton.

<Exit>

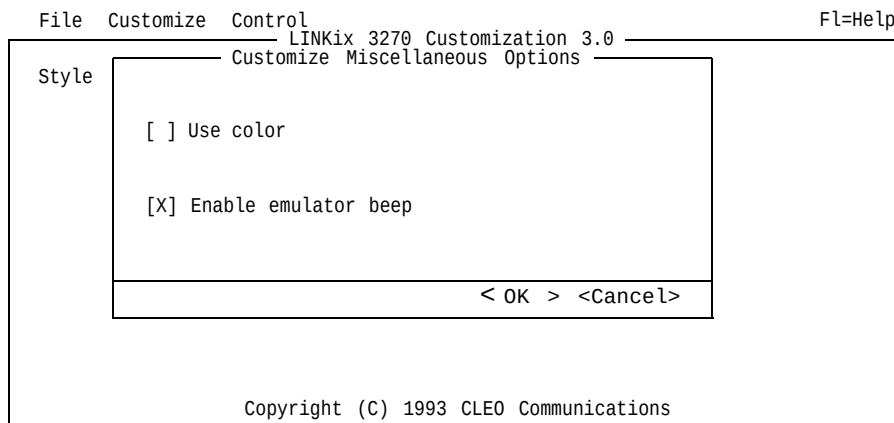
This pushbutton appears in dialogs for which no information needs to be entered; for example, on dialogs which allow you to view information on particular items.

Choosing this pushbutton causes you to exit from the dialog. The **Esc** key can be used as an accelerator key for this pushbutton, since it never appears in combination with the <Cancel> pushbutton.

The User Interface

Check Boxes

Check boxes are used to specify options which are either on or off. They are shown as a pair of square brackets [], with or without an X between the brackets. In the example below, there are two check boxes: one that allows you to specify whether or not to "Use color" (not checked) and another one that allows you to "Enable emulator beep" (checked).



When an X is displayed in the brackets, the item is selected. When the brackets are empty, the item is not selected. In the example above, the "Use color" check box is not selected and "Enable emulator beep" is selected.

Choosing a Check Box

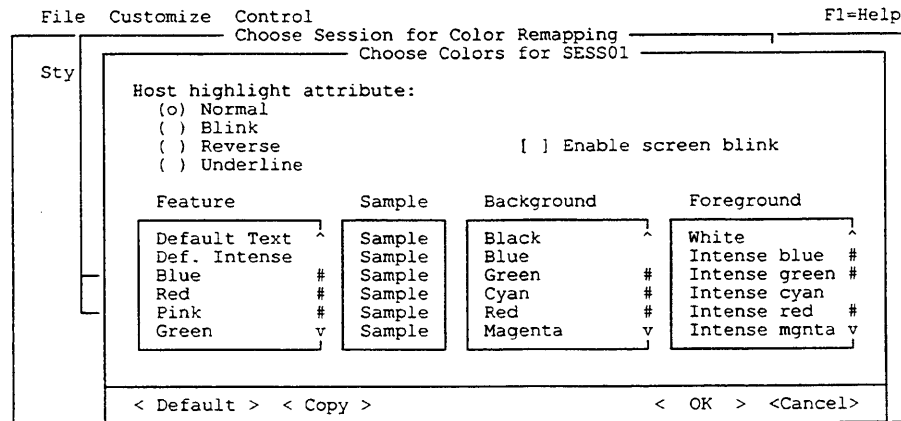
To choose a from the keyboard, do the following:

1. Press **TAB** to move to the check box.
2. Press **SPACE** to toggle the box on or off. Alternately, the **LEFT** and **UP** cursor keys can be used to toggle the box on and the **RIGHT** and **DOWN** cursor keys to toggle the box off.

You can also use the accelerator key to move to a check box. This automatically toggles the state of the check box.

Radio Groups

Radio groups provide a set of two or more options, of which one and only one can be selected at any one time. Each option is indicated by a radio button, which appears on the screen as a pair of parentheses. In the example below, the items under Host highlight attribute (Normal, Blink, Reverse and Underline) are a radio group.



A circle within the parentheses indicates which radio button in the group is selected. Since one and only one radio button can be selected at one time, when one button is selected any other selected button within the group is automatically deselected. In the Host highlight attribute group above, Normal is selected.

Choosing a Radio Button

To choose a from the keyboard, clothe following:

1. Press **TAB** to move to the radio group.

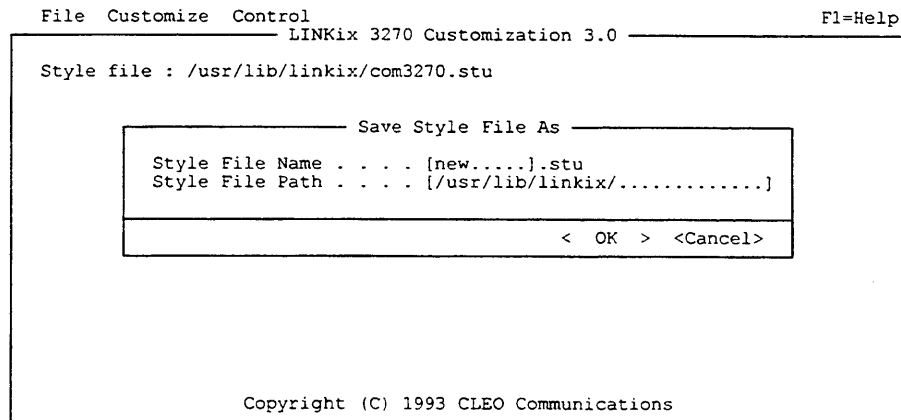
The User Interface

2. Use the cursor keys to select the button you want: **DOWN** or **RIGHT** move forward and **UP** or **LEFT** move backward. If you move forward while on the last button in the group, you move to the first radio button. In similar fashion, moving backward while positioned on the first button moves you to the last radio button.

You can also use the accelerator key to move to a radio button. This automatically selects that radio button.

Edit Boxes

Edit boxes are areas in a dialog where you can type information. They appear as a pair of square brackets separated by some amount of space, in which you enter data. They often contain default information, which you can either accept or replace. In the example below, both fields in the Save Style File As dialog are edit boxes.



In some edit boxes, a limited number of characters are shown in the edit box region, but the typing area scrolls when you type in more characters than the space allotted.

Typing Information in an Edit Box

To add or change information in an, do the following:

1. Press **TAB** or use the accelerator key to move to the edit box. Existing information in the edit box is shown in reverse video.
2. If you begin typing while the existing text is in reverse video, the highlighted text is all replaced with the first keystroke.

If you use the **LEFT** or **RIGHT** cursor keys or the **HOME** or **END** keys to position the cursor within the edit box before you edit the existing text, the highlighting disappears and you manually insert and delete characters as needed.

List Boxes

A list box is a control that contains a list of items, from which you choose one item. When an item is chosen from the list, you might view or specify further information on that particular item by selecting a pushbutton such as <Zoom> or <Edit>. The following example shows a dialog which contains a single control, a list box.

File Customize Control Choose Session to Customize F1=Help

Styl	Session Number	Long Name	Short Name	LU/Pool Name	Session Type
	1			COLPOOL	(Display)
	2	SESS02		COLPOOL	(Display)
	3	SESS03		PRT107	(Printer)
	4	SESS04		PRT308	(Printer)

< Zoom > < Exit >

Copyright (C) 1993 CLEO Communications

Also see **Dual List Boxes** for another way in which list boxes are used.

Selecting an Item from the List Box

To choose an item from the , you first need to move to the list box (unless it is the only control in the dialog or it's selected by default when the dialog appears) by using the **TAB** key or the accelerator key shown in the list box title. One item in the list box (usually the first) is shown in reverse video, indicating that it's selected.

To select a different item within the list box, do one of the following:

- ◆ Use the **UP** and **DOWN** cursor keys to position the cursor on the item you want.
- ◆ Press the key corresponding to the first character in the item name; if more than one item has this character, press the key again until the cursor reaches the one you want.

Once the item is selected, you typically carry out some action by choosing a pushbutton.

A list box may contain more items than the box can show at one time. To view items which are not currently shown, do one of the following:

- ◆ Use the **UP** or **DOWN** cursor keys to move the cursor past the top or bottom of the list box; the items in the box scroll down or up
- ◆ Use the **HOME** or **END** keys to move the cursor to the first or last item in the list box
- ◆ Use the **PAGE UP** or **PAGE DOWN** keys to move up or down through the list box items

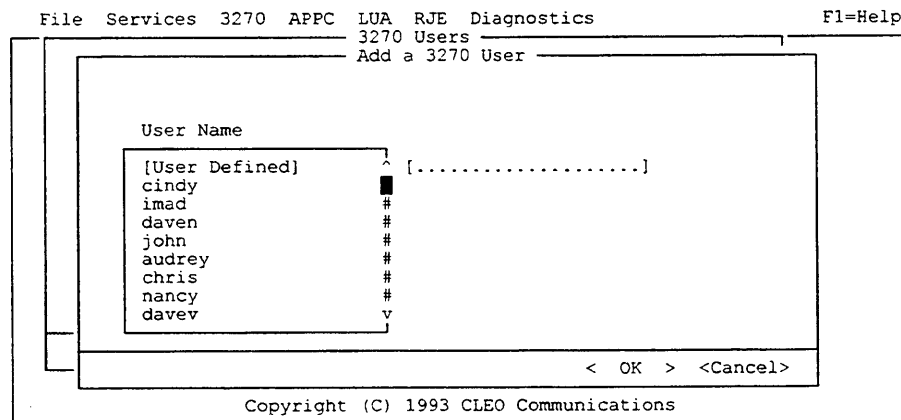
If the item is not currently visible in the list box, you can still select it by using the key corresponding to the first letter of its name. (In the example on the previous page, you could enter the Session Number.)

Elevator Bar

The elevator bar (shown on the right-hand edge of the list box) shows your current position in the list. If the list box is currently displaying the first few items in the list (or all the items, if the list is short enough for all the items to appear at once), the marker (a reverse-video space character) appears at the top of the right-hand edge of the box. As you scroll down the list box, the marker moves downward to indicate your approximate position in the list.

List Boxes with Associated Edit Boxes

In the example below (from the configuration program), the list box öUser namì has an edit box associated with it.



Where a list box has an associated edit box, you can specify an entry either by selecting it from the list box or by typing it into the edit box. In this example, you would normally specify the name of a user by selecting it from the list box. However, you could also select [User Defined] from the list box and type the user name into the edit box.

To select an entry from the list box, follow the same steps as for a normal list box. The data shown in the edit box corresponds to the selected entry in the list box, and is disabled (grayed) to indicate that you cannot alter it.

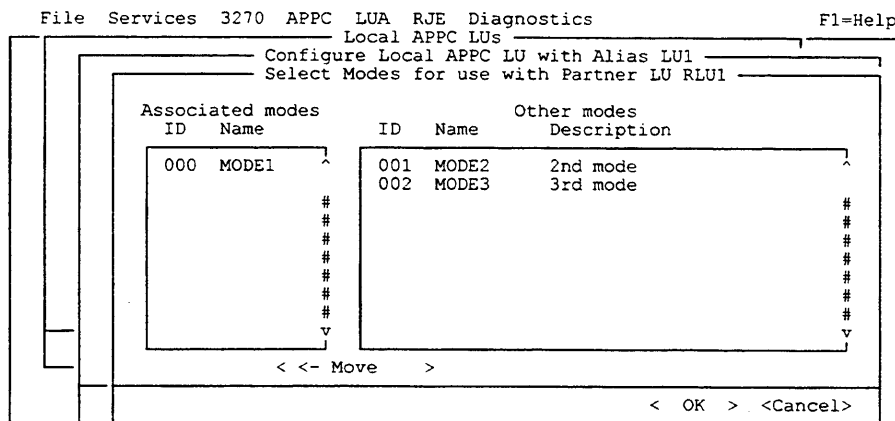
To type an entry in the edit box, follow these steps:

1. Select the entry [User Defined] that appears at the top of the list box. The edit box is cleared, and is enabled (not grayed) to indicate that you can type into it.
2. Use the **TAB** key to move to the edit box.
3. Type the information you want in the edit box.

The User Interface

Dual List Boxes

Dual list boxes allow you to select one or more items from a list. They appear as two list boxes from which you select an item and move it from one list box to the other by choosing the < Move > pushbutton that appears below or between them. In the example below (a dialog from the configuration program), 'Associated modes' and 'Other modes' are a pair of dual list boxes.



The left-hand list box lists all the modes used with this partner LU, and the right-hand list box lists all other modes which could be used by this partner LU.

Moving Items Between Dual List Boxes

To move an item from one list box to the other:

1. Use the **TAB** key (or the accelerator key shown in the list box title) to move to the list box which contains the item you want to move.
2. Select the item you want to move from this list box (see the previous section for details of how to select an item from a list box). The arrow next to the word Move points in the direction in which you want to move the item (away from the list box containing the selected item and toward the list box to which you want to move it).

3. TAB to the < Move > pushbutton. (You cannot use the accelerator key because you're in a list box.)
4. Press **ENTER**, or the accelerator key of the < Move > pushbutton, to move the item. It disappears from one list box and appears in the other.

Accelerator Keys

Each field in a dialog has an accelerator key, which is indicated by a highlighted character in the name of the field. You can move directly to another field in a dialog by pressing the key which corresponds to this character.

However, there are two situations in which accelerator keys cannot be used:

- ◆ When the cursor is in an edit box, since any character key you press is interpreted as text to be entered into the edit box.
- ◆ When the cursor is in a list box, since any character key you press is used to move the cursor to the next item in the list box that begins with that character.

To use accelerator keys in these situations, you must first use the **TAB** or **BACKTAB** keys to move the cursor to a field other than an edit box or list box.

There are also two other accelerator keys which are not indicated by highlighted characters:

- ◆ The **ENTER** key chooses the current default pushbutton (the one with the highlighted angle brackets)
- ◆ The **ESC** key is an accelerator for either of two pushbuttons < Cancel > or < Done >. (These two pushbuttons never appear on the same dialog.)

Other Interface Features

Disabled (Grayed) Items

Sometimes menu options or dialog items are disabled and cannot be selected. An option or item is disabled when it is not applicable for the current situation or when it cannot be changed. For instance, dialog items may not be available because of previous selections you made, such as selecting a particular check box, and some options may not be available in certain product packages.

Disabled options appear grayed (or dimmed), except on a monochrome screen which cannot show grayed or dimmed text. If you are using a monochrome screen, however, items which are disabled (grayed) have no accelerator keys showing.

On-line Help Screens

On-line help screens are available whenever the menu interface is active. A help screen is provided for each of the following:

- ◆ The main screen of each program - describing the function of the program and the menus available from the main screen.
- ◆ Each menu in the main screen's menu bar - describing the overall function of the menu and a summary of the options available in this menu.
- ◆ Each menu option - giving more details of the functions provided by this menu option.
- ◆ Each dialog — explaining the overall purpose of the dialog, each field in the dialog and the valid responses to each field.

To invoke the help screen for a particular dialog or menu item, press the **F1** key. The help screens are context-specific; that is, the information displayed relates to the particular dialog or menu item on which the **F1** key was pressed.

The example below shows the help screen for the 3270 emulation program's File menu.

File Cus	File	elp								
Style fil	Use the File menu to open style files, create new style files, save changes made to files, and to exit the Customization program. The following options are available from the File menu: <table> <tr> <td>Open</td> <td>Open an existing style file.</td> </tr> <tr> <td>New</td> <td>Create a new style file.</td> </tr> <tr> <td>Save</td> <td>Save changes to the current style file.</td> </tr> <tr> <td>Save As</td> <td>Save the current style to a different file.</td> </tr> </table>	Open	Open an existing style file.	New	Create a new style file.	Save	Save changes to the current style file.	Save As	Save the current style to a different file.	
Open	Open an existing style file.									
New	Create a new style file.									
Save	Save changes to the current style file.									
Save As	Save the current style to a different file.									
	< Next > < Previous > < UI Info > < OK >									

Open, create, or save style file, or exit customization program

When there's more help information than can redisplayed on the screen at one time, the < Next > and < Previous > pushbuttons can be used to display the next or previous page of information. To return to the dialog from which the help screen was called, choose < OK > or press the **ESC** key.

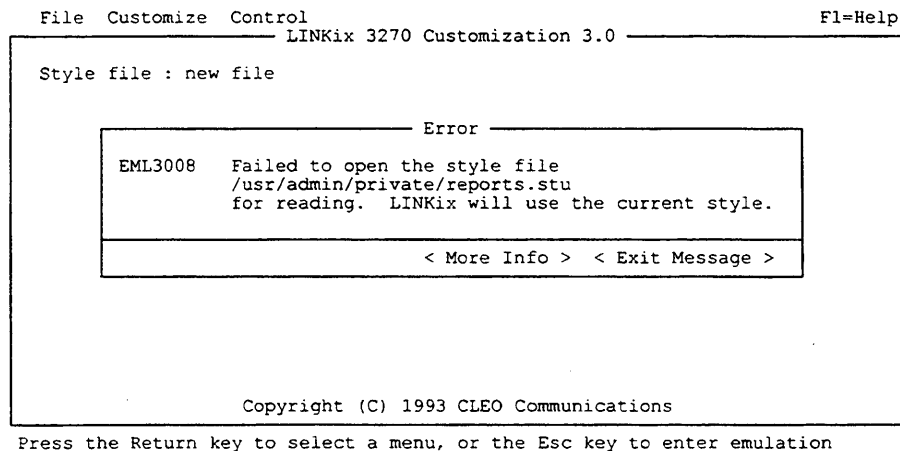
The < UI Info > pushbutton can be used to display help information on the LINKix user interface. As with the other help screens, you can page up and down through this information using the < Next > and < Previous > pushbuttons. To return to the original help screen, press the < OK > pushbutton or the **ESC** key.

While a help screen is displayed, you can press **F1** again to display a screen which gives more information about the help facility. Choosing < OK > or pressing the **ESC** key returns you to the help screen you were previously viewing.

The User Interface

Message Boxes

Message boxes display various messages as you use the menu interface. A typical message box is shown below.



There are three different types of message boxes, as follows:

Error

Error messages indicate a condition that requires action. This may be an error detected by LINKix 3270, for example in a file you were attempting to open. It may also be an error that you made, such as typing invalid data in an edit box and choosing < OK >; you must return to the edit box and correct the entry, or choose < Cancel > to abandon the dialog.

Warning

Warning messages indicate a condition that may require action. For example, if you load a 3270 style file that uses color when your terminal does not support it, a warning message informs you that the color mappings will be reset to their default values; you can either continue with the default mappings, or load another style file.

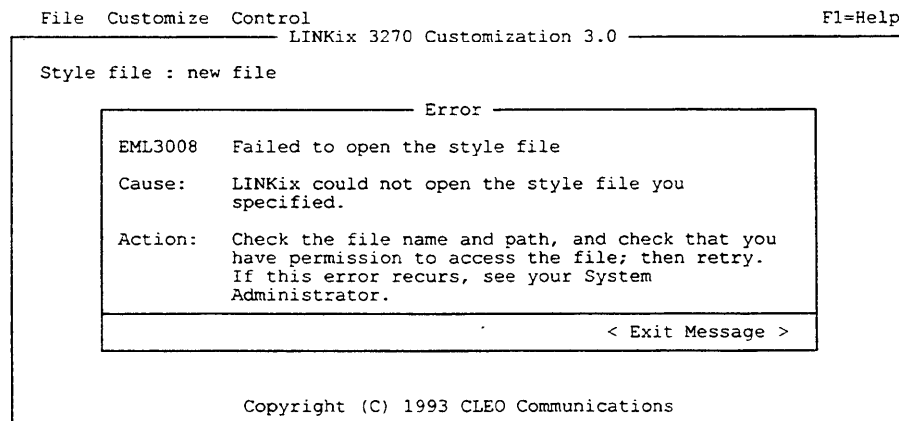
Information

Information messages provide confirmation that an action has been accepted (such as starting a file transfer process), or provide information related to the previous action that may be useful to you. No action is required.

The title of a message box indicates whether it is an Error, Warning or Information message. For all types of message box, choose the < Exit Message > pushbutton to acknowledge the message and continue with the program.

More Info

For Error and Warning message boxes, there is also a < More Info > pushbutton. If you choose this pushbutton, the message box is replaced by a larger box, such as the example below:



This box includes the message text that was shown in the original message box, and also more detailed information on the condition that caused the message and the action you should take. Again, choose the < Exit Message > push button to acknowledge the message and continue with the program.

Screen Redrawing

If your screen becomes corrupted during a 3270 session or if characters are lost due to line transmission problems, you can redraw the by pressing **CTRL + R** or **CTRL + L** from any menu or dialog. This does not affect data you have entered.

Fast Exit from Menu Interface Programs

While using any of the LINKix menu interface programs, you can use the **F3** key to perform a fast exit from any point in the menu interface, without having to close any displayed dialogs and return to the main screen.

You are prompted to save the configuration file or the style file before exiting. You can choose to save and exit, to exit without saving or to abandon the exit.

If you choose to save and exit, the exit process stops if it encounters a condition that requires a pop-up message (Error, Warning or Information); you can then take any action required, before pressing **F3** again to continue the exit.

If you were creating a new configuration file or style file, the Save as dialog appears, to let you specify the name and path for the file. The program then saves the file and exits to the UNIX command prompt.

For the 3270 emulation program, note that the **F3** fast exit applies only within the menu interface, and cannot be remapped. From 3270 emulation, use the 3270 key EXIT (default keystroke **CTRL + X**).

2. 3270 Emulation

This chapter introduces the major features of the 3270 emulation program and explains how to start a 3270 emulation session. The following information is covered:

- ◆ Overview of 3270 emulation features and operations
- ◆ Instructions for starting the 3270 emulation program
- ◆ Instructions on moving between 3270 sessions
- ◆ A description of 3270 data fields
- ◆ A description of the keystroke record and replay facility
- ◆ Instructions on running HLLAPI programs
- ◆ Suggestions for automating startup of 3270 emulation
- ◆ Fast Path to 3270 emulation

You can work with the defaults provided by the program for the appearance and operation of your terminal or you can change them. To customize these parameters before beginning emulation or file transfer, turn to chapter 5, **Customizing 3270 Emulation** or see your system administrator.

If you're an experienced user who wishes to get started quickly, see **Fast Path to 3270 Terminal Emulation** (p. 2-28); otherwise, read the detailed instructions that follow.

Overview of Emulation Features

The 3270 emulation program gives you access to most of the standard features of an IBM 3270 Information Display System, such as extended attribute byte (EAB) support for color and extended highlighting. (Note that some of these features are only available if your UNIX terminal supports them.)

In addition, LINK.ix 3270 provides file transfer and keystroke record/replay capabilities and allows you to develop and run High-Level Language Application programming Interface (HLLAPI) programs.

Keyboard Mappings

The keyboard for a IBM 3278/79 series display station has a group of keys that are not found on the standard UNIX terminal keyboard. LINKix 3270 provides the same functionality by mapping the functions that are needed in 3270 emulation to special keystroke combinations for your terminal. You can review the default keyboard mappings by turning to Appendix A, **Default Keyboard Mappings**.

Special 3270 function keys that are mentioned in this chapter (and their default mappings) include:

ACTIONS (CTRL + U)	SESS <i>n</i> (ESC <i>n</i>)
EXIT (CTRL + X)	STAT TOG (CTRL + W)
NEXTSESS (CTRL + V)	TEMPEXIT (CTRL + Z)
RESET (ESC R)	

3270 Sessions

Depending on the LINKix 3270 package that's installed, you can run one or more 3270 sessions concurrently:

- ◆ IBM 3270 session 3278/79 host display sessions (used for emulation, file transfer and HLLAPI applications)
- ◆ IBM 3270 session 3287 host printer sessions

The emulation program supports up to ten simultaneous host sessions, which maybe any combination of display sessions and printer sessions. If your configuration allows, different sessions can be connected to the same host or to different hosts.

Model 2-5 Display Support

With LINKix 3270 you can emulate the 3270 display models shown below.

3270 Display Model	Screen size

Display devicesModel 2	24x 80

Display devicesModel 3	32x 80

Display devicesModel 4	43 x 80

Display devicesModel 5	27x 132

You can only display screen models other than model 2 if your UNIX computer and your terminal's screen hardware can support them. However, you can use HLLAPI application programs to access sessions which use any screen model (even those which your UNIX terminal cannot access directly). See the *HLLAPI Programmer's Guide* for more information.

International Language Support

LINKix 3270 can communicate with hosts using international language variants of EBCDIC; this allows you to send and receive all the characters associated with a selected host language, including those which are specific to that language. The display of international characters on a terminal depends on the support for these characters provided by the UNIX computer and its terminals.

Background Operation

The 3270 emulation program can be run in the background by using a command-line option. When 3270 sessions are running in the background, you can use host printing applications with 3270 printer sessions, perform command-line file transfers which access 3270 display sessions or run HLLAPI applications. And although you cannot directly view the 3270 display, we've provided a HLLAPI terminal emulation

3270 Emulation

program (described in Appendix D) that allows you to access 3270 sessions that are running in the background.

File Transfer

LINKix 3270 supports IBM IND\$FILE file transfer, allowing you to transfer files between LINKix 3270 and host systems MVS/TSO, host systems VM/CMS and host systems CICS host systems. You can also use the file transfer capabilities to print the contents of a UNIX file on a host-connected printer by transferring the UNIX file to a host file or data set and then printing that file on the host printer.

You can issue the file transfer commands by using LINKix 3270's menus and dialogs, or you can transfer files from the UNIX command prompt. Chapter 4, **Transferring Files**, contains complete, step-by-step instructions.

Printer Emulation

LINKix 3270 lets a printer attached to your UNIX computer emulate an IBM 3287 host printer. It also provides the following print options:

- ◆ User-initiated local copy: Print the current 3270 display at the user's request
- ◆ Host-initiated local copy: Print the current 3270 display at the host's request
- ◆ Direct print output to a UNIX Printing to a file instead of a printer
- ◆ printing LU type 1 printing using a printer session
- ◆ printing LU type 3 printing using a printer session

LU type 1 printing is for host applications that use the SNA Character String (SCS). LU type 3 printing is for host applications that use the 3270 Data Stream. Examples of printers that support both LU type 1 and LU type 3 printing are the IBM 3287, 3289 and 4224.

Temporary Exit From 3270 Emulation

While using the 3270 emulation program, you can perform a temporary exit to another UNIX shell, leaving the 3270 emulation program running. This can be used

to list the contents of a directory, to run a HLLAPI application or to run a command-line file transfer process.

Type-ahead During 3270 Emulation

While waiting for the host to respond during 3270 emulation, you can continue entering up to 16 additional keystrokes which will be processed when the host responds; these can be data characters, cursor control keys or keys such as ENTER. If you type too many keystrokes, the keystroke lost message appears on the status line (x ?+) and the keyboard is locked until you press RESET.

You should not attempt to use this feature during the logon sequence, particularly while waiting for a password prompt from the host, as it can cause problems with some hosts. After you have supplied a password and received an indication from the host that you are successfully logged on, type-ahead can then be used.

Keystroke Recording and Replay

You can record sequences of keystrokes which you need to use regularly during 3270 emulation (for example, the key sequence required to startup a host application, log on and perform any standard initialization), and replay these sequences later instead of retyping all the keystrokes. The keystroke sequences can include data characters, cursor control keys, program function or program access keys, or keys such as ENTER which send information to the host.

Menu Interface

The 3270 emulation program provides pull-down menus with which you can customize and control 3270 emulation. An on-line help system is available by pressing **F1**.

The File Menu

You use the File menu to do the following:

- Open a style file, which contains parameters that control the appearance and behavior of your terminal in 3270 emulation
- Create a new style file
- Save changes to a style file

3270 Emulation

- ◆ Move from the main 3270 emulation menu to 3270 display sessions
- ◆ Exit LINKix 3270

The Customize Menu

Options under the Customize menu allow you to customize the following display and printer session features:

- ◆ Session characteristics Customizing 3270 session
- ◆ Output devices or files
- ◆ Display colors Customizing display colors
- ◆ Host language - the language variant of EBCDIC used at the host (if supported by your version of UNIX)
- ◆ Mapping 3270 functions to keys on your terminal's keyboard mapping 3270 functions

The Control Menu

The Control menu contains options that allow you to control the following 3270 emulation features:

- ◆ Control display and printer sessions (simulating powering the device on or off)
- ◆ Transfer files between the UNIX computer and the host
- ◆ View host response times
- ◆ Send NetView alert messages to the NetView operator

HLLAPI Support

You can write application programs that communicate with the host 3270 application using the LINKix 3270 HLLAPI programming interface. Application programs that use HLLAPI functions require an active 3270 emulation session.

You can use HLLAPI programs to automate your startup operations for communicating with a host application or to access host applications which require larger screen models than your terminal supports.

Starting the 3270 Emulation Program

Before starting the 3270 emulation program, you should make sure that you are configured as a 3270 user and that the LINKix control daemon has been started (as well as all necessary services). If you try to start the emulation program before the control daemon is started, you will get an error message (Could not access daemon's shared memory). See your system administrator for more information, or refer to the chapter on **Managing LINKix Components** in the *Administration Guide*.

You start the emulation program by entering the following command at the UNIX command prompt:

```
c3270 [-b | -e] [-i | -t <convfile>] [-s <stylefile>] [-m3 | -m4 | -m5 | -M] [-l <LU name>]
      [-h <Session IDs>]
```

Where the available options are:

-b

The -b option starts the 3270 emulation program as a background process. When you start the 3270 emulation program in the background, LINKix 3270 automatically enables all suitable sessions. If there is no suitable session, an error message is displayed and LINKix 3270 terminates. This option cannot be used with the -e option (see below).

Example: c3270 -b

See **Running 3270 Emulation in the Background**, for complete details.

-e

Using the -e option disables the menu interface, thus reducing the amount of UNIX memory required. If you specify this option you can use only 3270 emulation sessions, and you must use a style file which has at least one display session that's configured to be active initially.

Example: c3270 -e

-i | -t <convfile>

The -i option should be used when communicating with hosts using foreign language versions of EBCDIC. Specifying this option at startup activates LJINKix 3270's international support, allowing you to specify a host language other than U.S. English

3270 Emulation

(through the Customize menu) and to remap your keyboard to include any characters specific to that language (chapter 5 contains complete information). This option should only be used if your UNIX operating system and terminal hardware are set up to use the ISO8859 character set. If you don't use this option, the host language defaults to U.S. English and it cannot be changed. Note that this option cannot be specified if using the `-t` option (see below).

The `-t` option should be used if you need to specify your own translation table file for converting between ASCII and EBCDIC (this file is created with the `comtrdmp` and `comtrcnv` utilities; see **Custom Translation Tables** in chapter 5 for details). The name of a file containing the translation table (*convfile*) may be specified with or without a path; if the file is not in the current directory, include the full path. Note that this option cannot be specified if using the `-i` option (see above).

Examples:

```
c3270 -i
c3270 -t tablefile
```

`-s <stylefile>`

The `-s` option directs LINKix 3270 to load the specified style file. This option must be followed by a style file name (with or without a directory path). If you specify a path, LINKix 3270 tries to open the file in the directory given; if no path is given, the file is assumed to be in the current directory. The style filename must have the extension `.stu`, but you need not specify this extension (it will be added if you do not specify it). If you do not specify a style file, LINKix 3270 uses the default style file specified for you in the configuration file. See **Search Path for Style Files**, in chapter 5, for more information.

The style file contains parameters that control the appearance and behavior of your UNIX terminal in 3270 emulation:

- ◆ which sessions are initially active,
- ◆ session colors, keyboard mappings and audible signals,
- ◆ terminal type and
- ◆ printer destination.

The name of the current style file is displayed on the main screen of the 3270 emulation program. If you want to use a different style file after the 3270 emulation program has been started, use the File menu to open it.

If the style file you specify cannot be found or the file cannot be opened, the following message appears on the main screen when the program starts:

```

File  Customize  Control                               F1=Help
----- LINKix 3270 Customization 3.0 -----
Style file : new file

      Error
      -----
      EML3008  Failed to open the style file
               /usr/lib/linkix/com3270.stu
               for reading.  LINKix will use the current style.
      -----
               < More Info >  < Exit Message >

Copyright (C) 1993 CLEO Communications

Press the Return key to select a menu, or the Esc key to enter emulation

```

Choose < OK > to acknowledge the message. You can then try to open the correct file using the File menu. Alternately, you can continue with 3270 emulation using the default (current) style.

-m3 | -m4 | -m5 | -M

One of the -m options can optionally be specified to prepare the terminal for a model 3, 4 or 5 session; or the -M option can be specified if the terminal only provides mode1 2 monochrome support. Note that mode1 5 support may not be available on all terminals. If a screen model is not explicitly specified when starting the emulation program, a model 2(24 x 80) color display is assumed by default. If you specify a model type different from that specified in the style file or configuration, you can change that model type (if your system administrator has not disabled this option); see **Customizing Display Sessions** in chapter 5 for details.

Example: c3270 - M

-1 <LU names>

This option allows you to change the long name associated with one or more sessions, provided the configuration allows the LU/session mapping to be modified. Using this option when LU/session mapping cannot be modified causes c3270 to return with an error message.

This option provides a convenient way of using one configuration file and remapping LU names as needed.

3270 Emulation

The LU names specified with this option will override the LU names stored in the 3270 record of the configuration file, starting with session 1. If fewer names are specified with this option than sessions in the configuration file, the extra sessions remain unchanged. If more LU names are specified with this option than exist in the configuration file, the extra sessions are added using default values (note that there is a limit on the maximum number of enabled sessions). If more than 10 names are included with this option, *c3270* returns with an error message.

Example: *c3270 -1 LU1, LU2, LU3, LU4*

-h <Session IDs>

When using the menu interface to customize sessions, you are limited to 26 session IDs (represented by the letters A through Z). If you need to manipulate more than 26 sessions using the HLLAPI interface, you will have to remap some (or all) of the session IDs that are stored in the style file using the *-h* option.

The *-h* option allows you to specify one or more hexadecimal digits which will be used to remap the session IDs stored in the 3270 style file. This option allows a single user to make up to 254 sessions available at one time (by starting multiple copies of *c3270*), each one with a unique HLLAPI session ID.

Example: *c3270 -h 0x01, 0x02, , 0x04, 0x05*

When specifying a list of IDs, omitting an ID by typing two adjacent commas causes the existing session to have no ID. If you specify fewer IDs than are defined in the style file, extra IDs in the style file remain unchanged. If more session IDs are specified than are configured in the style file, the extra IDs are ignored.

Running 3270 Emulation in the Background

LINKix 3270 allows you to run the 3270 emulation program as a background process. When running the 3270 emulation program in the background, you can still access 3270 display sessions using a HLLAPI application or perform host printing with printer sessions. (You can also run a command-line file transfer process, but to do this you will need to use a HLLAPI application to logon to the host, since you cannot do this directly when the program is in the background. See **Command-line File Transfers**, in chapter 4, for more information.) And although you cannot directly view the 3270 display, we've provided a HLLAPI terminal emulation program (described in Appendix D) that allows you to access 3270 sessions that are running in the background.

When you start the 3270 emulation program in the background, the following sessions will automatically be activated:

- ◆ All initially active printer sessions
- ◆ All initially active display sessions which have a session ID assigned

If there are no suitable sessions, LINKix 3270 displays an error message and the program terminates. See **Customizing Display Sessions** or **Customizing Printer Sessions** in chapter 5 for information on session IDs and initially active sessions.

Before starting the 3270 emulation program in the background, you should make sure of the following:

- ◆ You are configured as a LINKix 3270/3270 user
- ◆ The control daemon has been started, as well as all necessary LINKix services
- ◆ Your default style file, or the style file you will specify on the command line, contains at least one initially active display session with a session ID or at least one initially active printer session.
- ◆ See your System Administrator if necessary.

Stopping Background 3270 Emulation

To stop the background 3270 process, use the *stop3270* utility. This program stops all copies of the 3270 emulation program which are running with your UNIX user ID. The *stop3270* utility responds with a message indicating the number of copies of the 3270 emulation program it found and stopped.

Session Screen and Main Screen

After the 3270 emulation program starts, one of the following is displayed:

- ◆ an enabled or active 3270 session, with a status line at the bottom
- ◆ the main screen of the 3270 emulation program

If you see the LINKix 3270 main screen instead of a 3270 display, then you do not have any 3270 sessions set to be enabled at initialization. For instructions on enabling sessions, refer to chapter 3, **Controlling 3270 Emulation**. For instructions on updating your style file to make a session enabled at initialization, refer to **Customizing Display Sessions** in chapter 5.

The following is a sample 3270 display.

```
----- ISPF/PDF PRIMARY OPTION MENU -----
0 ISPF PARMS - Specify terminal and user parameters      USERID - XDCL04
1 BROWSE     - Display source data or output listings   TIME    - 10:41
2 EDIT       - Create or change source data            TERMINAL - 3278
3 UTILITIES  - Perform utility functions                PF KEYS - 24
4 FOREGROUND - Invoke language processors in foreground
5 BATCH      - Submit job for language processing
6 COAND      - Enter TSO command or CLIST
7 DIALOG TEST - Perform dialog testing
8 LM UTILITIES- Perform library administrator utility functions
9 IBM PRODUCTS- Additional IBM program development products
C CHANGES   - Display summary of changes for this release
T TUTORIAL   - Display information about ISPF/PDF
X EXIT       - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF.
```

```
TSO1011      LTTI 00:10.4                      rlp                      LU 06
```

Switching Between 3270 Sessions and the Main Screen

From a 3270 display session you can move to the main screen (to access the File, Customize or Control pull-down menus) at any time to customize and control 3270 emulation. To move from a 3270 display to the main screen, press ACTIONS.

```
File  Customize  Control                               F1=Help
LINKix 3270 Customization 3.0
Style file : /usr/lib/linkix/com3270.stu

Copyright (C) 1993 CLEO Communications

Press the Return key to select a menu, or the Esc key to enter emulation
```

Notice that the name of the style file also appears near the top of the main screen. To resume 3270 emulation from the main screen, press **ESC** or from the File menu choose Return to Emulator. Impressing **ESC** yields the error message "No active display session of suitable model size, no sessions have been enabled (see chapter 3, **Controlling 3270 Emulation**).

Temporary Exit to the UNIX Command Prompt

To move temporarily from a 3270 display session (or the main screen) to a UNIX command prompt, leaving the 3270 emulation program running, press TEMPEXIT (default: **CTRL + Z**). This starts a new UNIX shell of the type specified by the SHELL environment variable (the default is /bin/M). You could use this feature to find the name or directory of a style file, to use the command-line file transfer feature or to run a HLLAPI application, or to run any other UNIX application while maintaining the host sessions.

To resume 3270 emulation from a temporary exit, use the end-of-file (EOF) key sequence for your terminal (typically **CTRL + D**).

Communicating with the Host

To conduct, you logon to a host and communicate with its control and application programs through an active session. You move to the selected session by choosing Return to Emulator under the File menu or by pressing the **ESC** key. Then you

3270 Emulation

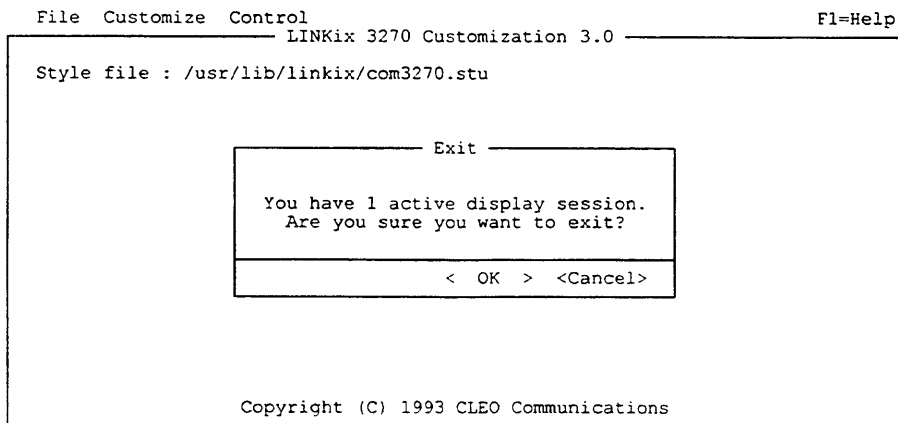
switch to another active session if desired, using the NEXTSESS key and follow the host computer's logon instructions, if any.

Ending Emulation

When you have finished using 3270 emulation and want to exit:

1. Log off the host system (for each active 3270 session).
2. Press ACTIONS to display the main screen.
3. From the File menu, choose Exit or press **F3**. As an alternative to steps 2 and 3, or if you started 3270 emulation using the -e option, you can press EXIT (default: **CTRL + X**) to exit the program.

If you exit while a session is still enabled, a message box is displayed (when you log off the host system, your sessions remain enabled until you explicitly disable them). In the message box, choose < OK > to disable all enabled sessions and end 3270 emulation, or choose < Cancel > to continue with the program. If you wish to exit to UNIX while maintaining your host session(s), use the TEMPEXIT key (**CTRL + Z**).



3270 Operations

With the 3270 emulation program, your UNIX computer emulates a 3270 terminal, which can access host computers. To operate your terminal during 3270 emulation, you should understand how to use the keyboard in a 3270 emulation session. You also should understand information provided by the status line.

Using the Keyboard

During terminal emulation, your keyboard must perform the same 3270 functions that are normally provided by a 3278/79 keyboard. LINKix 3270 lets your terminal perform these functions by assigning them to specific keystrokes.

If you want to use other keystrokes to represent the 3270 keys, make sure you're using the correct entry for your terminal in the terminfo database on your UNIX computer and that this entry contains definitions for all the keys you need to use.

Appendix A describes the default key definitions for emulating the 3270 keyboard, and the section **Customizing Key Definitions** in chapter 5 contains instructions on viewing and changing these defaults. See your system administrator if necessary.

The accelerator keys direct accelerator keystrokes shown in the File, Customize and Control menus (which allow you to jump straight to specific dialogs from the main screen) cannot be remapped to different keys. Key remapping applies only to display emulation.

If you're using a host language which does not match your keyboard type, you may need to use some character keys which do not appear on your keyboard. You can remap these 3270 keys to suitable keystrokes, or you can use the escape sequences provided by your UNIX operating system to generate them (see your operating system documentation).

Controlling a Terminal's Cursor

The default settings for cursor control keys (**UP**, **DOWN**, **LEFT**, **RIGHT**, **TAB** and **HOME**) during 3270 emulation are similar to their functions during normal UNIX terminal operation. There are, however, some important exceptions. Appendix A fully describes the use of cursor control keys.

Using Other Keys

Appendix A also describes the function of the edit keys (those that enter special characters or delete whole fields or displays) as well as program access keys, program function keys, 3270 system keys and printer control keys.

3270 Status Line

When you are in 3270 emulation, the last line of your screen shows status information about the current session. If your terminal’s screen has more lines than it requires for the 3270 emulation display (for example, if you’re using the 24-line screen model 2 on a 25-line screen), the status line is shown below the 3270 emulation display. If not (for example, if you’re using the 43-line screen model 4 on a 43-line screen), the status line shares the last line of the 3270 emulation display; pressing the STAT TOG (Status Line Toggle) key switches the display on the last line between the last line of the 3270 display and the status line information. You can also use this key with any screen model to turn off the status line until the information on it changes, for example to show a ‘Do Not Enter’ message.

The following diagram shows the format of the status line.

SB?	X o-o()	LTTI 00.03.1	NUM ^	/usr/bin/lp	LU001	03-020 B
Session Status	Do Not Enter	Comm. Check or LTTI	Numeric Field Insert Indicator	Printer Identifier or Record/Replay ID	LU Name	Cursor Position Session ID

- Session Status

Session Status messages indicate whether or not the UNIX computer and the host system are ready to communicate on this session or if they’re already communicating.
- Do Not Enter

Do Not Enter messages indicate the current state of the 3270 session. Some of these messages indicate normal conditions; others indicate that your 3270 keyboard is disabled temporarily and specify the reason for this.
- Comm. Check or LTTI

Communications Check codes indicate the current state of the connection to the host. Some of these messages indicate normal conditions which occur during initialization; others indicate problems with either hardware or configuration.

The Last Transaction Time indicator (LTTI) indicates the time taken for the most recent transaction with the host. This information is only displayed if there is no communications check code displayed; it also does not appear if your System Administrator has not given you permission to view response time information.

Numeric Reid

The Numeric field indicator appears if the cursor is in a numeric-only field.

Insert Indicator

The Insert indicator shows when your keyboard is in insert mode (you can insert a character at the cursor location, not typeover). Press the INSERT key (default: **INSERT**) to enter insert mode; press RESET (default: **ESC R**) to exit insert mode.

Printer Identifier

The Printer identifier shows the name of the printer or file to which local copy print requests on the current 3270 session are directed. This is specified in the Customize Display Session dialog (see chapter 5, **Customizing 3270 Emulation**).

Record/Replay ID

The Record/Replay ID message is shown instead of the printer identifier while you are recording a keystroke sequence or replaying a previously recorded sequence. This is displayed as RECORD *nn* or REPLAY *nn*, where *nn* is a number (1Æ 24) that identifies the PF key associated with the keystroke sequence.

The message BUFFER LOW (*mm*) can also appear in this position if you are recording a keystroke sequence and are approaching the limit of 1500 characters: *mm* is the number of remaining keystrokes you can enter before reaching the limit.

LU Name

The LU Name shows the name of the 3270 LU being used for the current 3270 session. This name is specified in the configuration file; you can change it if your System Administrator has given you permission to remap sessions (see chapter 3, **Controlling 3270 Emulation**).

Cursor Position

This information shows the cursor's current position on the screen (row-column).

3270 Emulation

Session ID This is the one-character session ID of the current 3270 session, if one is assigned. You can change the session ID by using the Customize Display Session dialog (see chapter 5, **Customizing 3270 Emulation**).

Status Line Display When Recording/Replaying Keystrokes

When you use the RECORD or REPLAY keys, the status line is temporarily replaced by a list of the identifiers assigned to recorded keystroke sequences. The normal status line display is restored when you select an identifier for the keystroke sequence to record or replay.

Switching Between 3270 Sessions

To switch from one 3270 session to another, use the 3270 system keys. Press NEXTSESS for the Next Session or enter the key combination for the session you want to move to (see **Session Selection Keys** in Appendix A).

3270 Data Fields

This section describes the characteristics of 3270 data fields.

Attribute Characters

On formatted 3270 displays, an invisible attribute character at the beginning of each field determines whether you can enter data in the field and, if so, the type of data you can enter (numeric, alphabetic or both). The attribute character also determines whether information is displayed and, if it is, determines the intensity or color of the display. A 3270 display with no attribute characters (and therefore no fields) is referred to as an unformatted display.

EAB Characters

Some host applications use the feature to improve the overall quality of the display. If your terminal supports EAB it can display Display features reverse video, Display features blinking characters and Display features underlining, and it can show characters in different Display features colors. However, these features will only be available if your UNIX computer and terminal support color.

If the host application does not use this feature, the characters are displayed as one of four base colors, depending on the attribute byte of the field. These colors indicate whether the field is protected (meaning you cannot enter data into it) and whether it is highlighted or normal intensity.

The following table shows how the color option and the field attributes determine the base color. The base color corresponds to one of the four colors listed in the Color Mappings dialog. The actual color displayed on the screen, however, depends on whether you're using a color or mono display and what color mappings you are using for that session (see chapter 5, **Customizing 3270 Emulation**).

Field attributes	Color
Unprotected, normal intensity	Green
Unprotected, intensified	Red
Protected, normal intensity	Blue
Protected, intensified	White

Keystroke Recording and Replay

The 3270 emulation program provides facilities for recording sequences of keystrokes used during 3270 emulation and replaying these sequences. This allows you to save sequences of keystrokes that you use regularly (for example, to start up a host application and log on, or to enter a complex host command) and to use the recorded sequences instead of having to type all the keystrokes again.

Recorded sequences are identified by one of the PF keys PF1 to PF24; to identify a sequence to be recorded or replayed, you press the RECORD or REPLAY key followed by the appropriate PF key. This means that you can have up to 24 recorded sequences at a time. The PF keys retain their standard functions for all other uses; they are only interpreted as identifiers for keystroke sequences when they follow the RECORD or REPLAY keys.

Valid Keystrokes

Any of the keys in the Program Function, Program Access, 3270 System, Edit and Characters key groups can be included in a recorded key sequence. See Appendix A of this manual, or the Customize Key Definitions dialog in the 3270 emulation program's menu interface, for more information on the key groups. Note that it is the host 3270 keys which are recorded, not the local mappings of them on your terminal; your recorded key sequences are not affected if you remap your keyboard.

Other keys, which perform local functions such as TEMPEXIT or REDRAW rather than host functions, are not recorded. In particular, when you switch to another 3270 session, to a TEMPEXIT shell or to the 3270 control interface, recording is suspended; subsequent keystrokes are not recorded until you switch back to the session on which you started recording.

You can record keys which request action from the host, such as ENTER or PA1. However, because these keys transfer control to the host application so that the 3270 emulation program cannot accept further input until the host has responded, keystrokes in the recorded sequence which immediately follow these keys may not be sent to the host when the sequence is replayed. This is the same behavior as if you typed the keystrokes instead of using recording and replay; keystrokes typed immediately after ENTER or PA 1 may not be accepted.

Recording a Keystroke Sequence

To record a sequence of keystrokes:

1. Press the RECORD key (**ESC S**). The status line displays the message RECORD PF KEY, as shown below. It may also display a list of numbers enclosed in parentheses; these are the numbers of the PF keys which already have key sequences associated with them.

```
RECORD PF KEY (1 4 11 12 15 21)
```

2. To record a keystroke sequence, press any PF key (PF1 - PF24). To abandon keystroke recording, either press RECORD or QUIT (**CTRL + T**) and you'll return to normal 3270 emulation.

If you press a PF key, the status line returns to normal. The RECORD *nn* message is displayed (see status line layout, below), where *nn* is the PF key number you selected. If this PF key already has a keystroke sequence assigned to it, a beep sounds to warn you (unless the "Enable emulator beep" option in the Customize Miscellaneous Options dialog is turned off). To avoid overwriting the existing sequence, press QUIT, keystroke recording is abandoned and the existing sequence is unchanged.

3. Type the keystrokes to record. You can use any of the keys in the Program Function, program Access, 3270 System, Edit and Characters key groups. Other keys, which perform local functions such as TEMPEXIT or REDRAW rather than host functions, can be entered but are not recorded. See **Valid Keystrokes** above for more information.

Because the DELETE key (**DEL**) is used to clear a recorded sequence, it cannot be the first keystroke in the sequence. See **Clearing a Recorded Sequence** below for more information. Also, because the RECORD key stops recording a keystroke sequence, it too cannot be the first keystroke; otherwise, the PF key is marked as having a sequence assigned to it, but the sequence does not contain any keystrokes.

You can record up to 1500 keystrokes in one sequence. If you record more than 1475 keystrokes, a warning beep sounds and the RECORD *nn* message on the status line is replaced by the BUFFER LOW (*mm*) message; *mm* is the number

3270 Emulation

of keystrokes which can yet be added to the sequence. This number is decremented as you continue to enter keystrokes.

At any time during this step, you can abandon recording and return to normal 3270 emulation by pressing the QUIT key. The keystrokes you've entered are not recorded; any previous sequence assigned to the PF key is left unchanged.

4. When you have entered all the keystrokes you wish to assign to a PF key, press the RECORD key again and the keystroke sequence is saved. To replay the sequence, press the REPLAY key (**ESC P**) followed by the same PF key (see **Replaying a Recorded Sequence** below).

Pausing Within a Sequence

Sometimes you may need to record a keystroke sequence that allows you to enter a combination of fixed and variable keystrokes. For example, you may want to record a host command in such a way that you can supply different parameters to the same command each time you use it, or you may want to force a recorded logon sequence to stop for you to type the password (instead of recording the password).

You can use the PAUSE key (**ESC W**) while recording a keystroke sequence to indicate where the sequence should stop for additional input data. When you replay the same keystroke sequence, replay stops at this point to allow you to type the new data before continuing.

To include a pause in a recorded key sequence:

1. While recording the sequence, press the PAUSE key.

The message RECORD PAUSED replaces RECORD *nn* on the status line. Any keystrokes that you type at this point are not recorded; however, the keystrokes you have already recorded are retained.

While recording is paused, you can press the RECORD key to stop recording and store the keystrokes you have entered, or you can press the QUIT key to abandon recording. Each of these keys works in the same way as if the PAUSE key had not been pressed.

2. After typing any variable data, press the PAUSE key again. The status line display is restored, and recording continues as usual.

3. Repeat steps 1 and 2 during recording at each place where you need a pause in the sequence.
4. Save the recorded key sequence by pressing the RECORD key.

Replaying a Recorded Sequence

To replay a recorded key sequence:

1. Press the REPLAY key (**ESC P**).

The status line changes to display the message REPLAY PF KEY, as shown below. It also displays a list of numbers enclosed in brackets; these are the numbers of the PF keys which have key sequences associated with them.

REPLAY PF KEY (1 4 11 12 15 21)

2. Press any PF key (PF1 - PF24) whose number is shown in the list.

To abandon keystroke replay and return to normal 3270 emulation, press QUIT.

If you pressed a valid PF key, the status line returns to normal. The REPLAY *nn* message is displayed, where *nn* is the PF key number you selected, and the recorded keystroke sequence is replayed.

If you pressed a PF key for which no sequence was recorded, or you pressed a non-PF key, a warning beep sounds (unless the `Enable emulator beep` option in the Customize Miscellaneous Options dialog is turned off). The REPLAY message remains displayed. Either select a valid PF key or press QUIT to abandon replay.

3. If you replay a key sequence that contains a pause, replay stops at the point where you entered the PAUSE key. The message REPLAY PAUSED replaces REPLAY *nn* on the status line.

Type in the appropriate keystrokes, and press the REPLAY key. The replay operation continues where it left off.

3270 Emulation

- 4 . To stop the replay at any time before it has finished, press QUIT, the remaining keystrokes are not replayed, and you return to normal 3270 emulation.

Note: If you enter a keystroke which takes you out of the current 3270 session, such as ACTIONS or FILE XFR (which move to the 3270 control interface), NEXTSESS or SESS *n* (which move to another 3270 session) or TEMPEXIT (which moves to a new UNIX shell), the replay is suspended while you are not in the 3270 session in which you started replay. When you return to the session, the replay resumes.

Other keystrokes typed during a replay sequence, such as edit keys or character keys, are queued up in the 3270 emulation program's type-ahead buffer and will be output at a pause or after the replay has finished.

Clearing a Recorded Sequence

If you have recorded a key sequence which you no longer want to use, you can clear it so that there is no longer a key sequence assigned to the PF key.

- 1 . Press the RECORD key, then press the PF key which identifies the sequence you want to clear.

A warning beep sounds (unless the "Enable emulator beep" option is off), to remind you that this PF key already has a key sequence assigned.

- 2 . To clear the keystroke sequence, press the DELETE key. The sequence is cleared and this PF key no longer appears in the list of PF keys.

To abandon clearing the sequence, press the QUIT key. The recorded sequence is left unchanged and you return to normal 3270 emulation.

Saving Keystroke Sequences

Chapter 5 of the *3270 User's Guide* describes how to use 3270 style files to define the appearance and behavior of your 3270 emulation. You can customize the 3270 emulation program as you use it, and the changes you make are immediately reflected in the program; however, if you want to keep the changes so that you can use the same customization next time you use the 3270 emulation program, you need to save them to a style file.

The 3270 emulation program stores recorded keystroke sequences in the same way as customization changes. Any keystroke sequences you record during 3270 emulation are available until you exit the 3270 emulation program, or until you open a different 3270 style file. In order to keep the sequences you have recorded, you must save your 3270 customization to a style file. See chapter 5 of the *3270 User's Guide* for information on how to do this.

Warning: If you save the style file while you are recording or replaying a key sequence (even if you have used the PAUSE key to pause recording), the key sequence cannot be saved. The new key sequence (and any previous key sequence assigned to the same PF key) will be lost; the PF key will be left without a key sequence assigned. The same applies if you exit the 3270 emulation program during recording or replay. To avoid losing keystroke sequences, always ensure that you have finished recording or replaying key sequences before saving the style file or exiting the program.

Multiple Sessions

If you are using more than one 3270 display session at a time, the key sequences you define are shared between all your sessions. This means that you cannot record a sequence for a particular PF key on one session, and at the same time record or replay a sequence identified by the same PF key on another session. For example, if you are recording a sequence for PF10 on one session, you cannot change to another session and replay the sequence for PF10 until you have finished recording it. Similarly, if you are replaying a sequence for PF10 on one session, you cannot record a new sequence for PF10 on another session until you have finished replaying it. However, you can replay a sequence identified by the same PF key on more than one session at a time.

If you try to use a keystroke sequence which is currently in use, the message ERROR - IN USE is displayed on the status line. Press the RESET key to continue.

Running HLLAPI Programs

During , you can run programs that use the HLLAPI application program interface provided with LINKix 3270. These programs let you simulate the actions of a 3270 user by entering data and handling data received from the host. For example, you can use a HLLAPI program to log on to a 3270 host automatically.

Before you begin this procedure, refer to **Customizing Sessions** in chapter 5 for details on how to change a session ID. Also, check with the provider of your HLLAPI program to make sure that the correct session is selectable.

To run a HLLAPI program, follow these steps:

1. Start the 3270 emulation program.
2. Check that the session ID of an active 3270 display session matches the session ID used by the HLLAPI program. This information is in the Customize Display Session dialog (see chapter 5 for more information).
3. Switch to the 3270 session and logon to the host, unless the HLLAPI program performs these actions automatically.
4. Move to another terminal, or use the TEMPEXIT keystroke to exit to another UNIX shell, and start the HLLAPI program. If you logon at another terminal, make sure you use the same logon ID as you used to run the 3270 emulation program; the two programs must run with the same effective user ID. Take note that by moving to another terminal to run the HLLAPI program you can watch the program manipulate the 3270 display.

Alternatively, if you are using a HLLAPI application which performs all necessary interaction with the host and does not require any operator input, you can start the 3270 emulation program in the background (see **Running 3270 Emulation in the Background**, earlier in this chapter) and then run the HLLAPI application. Note that you still require a 3270 session with a session ID which matches each session ID used by the HLLAPI application.

For details on writing HLLAPI programs, see the *HLLAPI Programmer's Guide*.

Automating 3270 Emulation Startup

You can automate 3270 emulation automating startup procedures to minimize your keying sequence by taking the following steps:

1. Create a style file that's customized to make at least one session active when LINKix 3270 is started.
2. Use a UNIX shell script to start the program specifying this style file as a parameter.

You can also automate startup to a particular host application by writing a HLLAPI application program to log on to the host and perform whatever initial steps are required. For details on writing HLLAPI programs, see the *HLLAPI Programmer's Guide*.

Fast Path to 3270 Terminal Emulation

This is designed to help experienced users begin terminal emulation after the basic configuration settings have been set. It provides a brief outline of 3270 display and 3287 printer emulation. For more detailed instructions, read the appropriate section(s) earlier in this chapter.

1. Prepare for 3270 emulation.

Make sure that you are configured as a 3270 user and, if required, that the LINKix 3270 control daemon has been started (as well as all necessary LINKix services). See your system administrator or refer to the chapter on **Managing LINKix Components** in the *Administration Guide*.

2. Start the program.

You start the emulation program by entering the following command at the UNIX command prompt:

```
c3270 [<options>]
```

See p. 2-7 for details on all available options.

3. Enable one or more 3270 displays.

Before using a session for 3270 emulation, you must enable it (communicate with the host). The session you want to use may already be enabled. If you move right into a 3270 session when you start the program (if you see a host screen with a status line at the bottom) you have at least one enabled session. If the session you want to use is enabled, skip this step and continue with step 4 to enable a printer session or step 5 to begin emulation.

- a. From the main screen of the program, pull down the Control menu and choose Display Sessions. (From an active 3270 session, press ACTIONS to access the main menu.)
- b. In the Control Display Sessions dialog, highlight the session to enable.

- c. Choose < Change status >. The status changes from Disabled to Enabled and then to SSCP (system services control point), if your system is configured correctly and the data link is active.
- d. Repeat the enabling process for as many sessions as you need.
- e. Highlight the session you wish to use first and choose < Make Current >.
- f. Choose < Done > to return to the 3270 emulation main menu.

4. Enable one or more 3287 printer sessions (optional).

Before you can use a printer session, it must be enabled (the default style file, *com3270.stu*, provided with the default configuration file, *com.cfg*, may have the configured printer sessions enabled at startup). To see if the printer session you want to use is enabled and/or change its status:

- a. From the main menu, pull down the Control menu and choose Printer Sessions.
- b. In the Control Printer Sessions dialog, highlight the session you want to enable from the list of sessions.
- c. Choose < Change status >. The status changes from Disabled to Enabled.
- d. Enable as many printer sessions as you need.
- e. Choose < Done > to return to the main menu.

Printer sessions do not have separate screens as displays do. Data sent to printer sessions is directed to the configured device or file.

5. Begin emulation.

- a. From the main menu, press **ESC**. If you see the No Active Display Session error message, choose the < Exit Message > pushbutton and enable a session by following the instructions in step 3.
- b. Using your 3270 session, follow logon or application instructions to use your host system.

3270 Emulation

6. Switch between active s.

Press the SESS *n* session selection key which corresponds to the number of the session you want to move to, or press NEXTSESS (which moves to the next session) until you reach the session you want. See Appendix A for the default mappings of these keys on your terminal.

7. Exit emulation. Exiting

- a. Log off all hosts. (This is recommended but not always necessary. LINKix 3270 can terminate active sessions if you exit emulation without logging off. However, with some host applications this may cause problems when you next try to log on.)
- b. Press EXIT to exit emulation and return to the UNIX command prompt.
- c. If any 3270 sessions are active, a message will be displayed asking you to confirm the exit. Choose < OK > to exit or < Cancel > to return to 3270 emulation. To the host it will appear as if the devices are being powered off when you exit emulation.

If you wish to exit to UNIX while keeping your host sessions active in the background, use the TEMPEXIT key.

3. Controlling 3270 Emulation

This chapter explains how to control 3270 emulation using the File and Control menus. It contains the following information:

- ◆ Descriptions of the pull-down menus used to control 3270 emulation
- ◆ Step-by-step instructions for controlling display sessions
- ◆ Step-by-step instructions for controlling printer sessions
- ◆ A brief description of how to transfer files between your UNIX computer and a host (complete file transfer instructions are in chapter 4, **Transferring Files**)
- ◆ Step-by-step instructions for viewing host response times
- ◆ Step-by-step instructions for sending NetView alert messages
- ◆ Instructions for resuming or beginning 3270 emulation
- ◆ Instructions for exiting the 3270 emulation program or continuing emulation

You can work with the defaults provided by the program for the appearance and operation of your terminal or you can change them. For instructions on customizing these parameters turn to chapter 5, **Customizing 3270 Emulation**.

Main Screen and Menus

After starting the 3270 emulation program, you can access the main menu screen at any time you are using a. To do so, press ACTIONS.

```
File Customize Control                               Fl=Help
LINKix 3270 Customization 3.0
Style file : /usr/lib/linkix/com3270.stu

Copyright (C) 1993 CLEO Communications

Press the Return key to select a menu, or the Esc key to enter emulation
```

Main Screen Menus

The main screen provides three menus for controlling 3270 emulation:

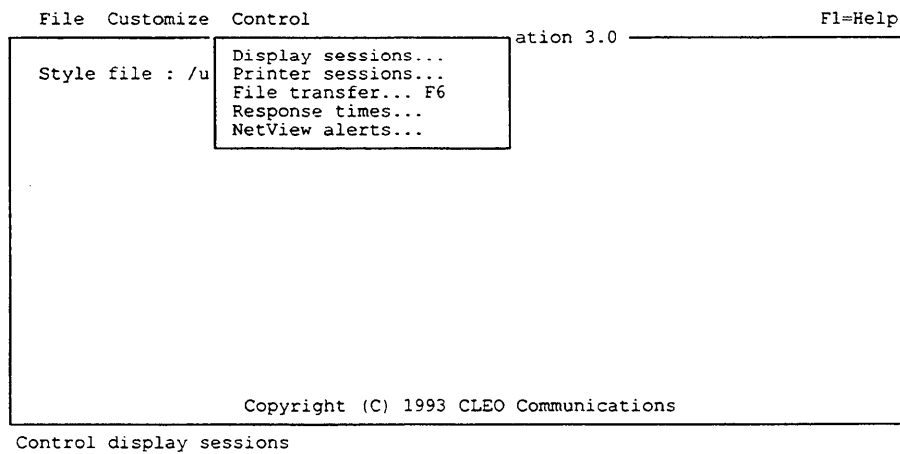
- ◆ File
- ◆ Customize
- ◆ Control

Chapter 5, Customizing 3270 Emulation, explains how to use the Customize menu to change your terminal's appearance and performance during 3270 emulation. Chapter 5 also explains how to use the File menu to save the changes in style files for future use. This chapter covers the options in the Control menu.

The Control Menu

Use the Control menu to do the following:

- ◆ Control display and printer sessions
- ◆ Transfer files between your UNIX computer and a host
- ◆ View host response times
- ◆ Send NetView user alert messages to host operators



Moving to Terminal Emulation

To move from the main screen to emulation, press **ESC** or select the File menu and choose Return to Emulator. Impressing **ESC** yields the error message "No active display session of suitable size", no sessions have been enabled. The Control menu, described in this chapter, allows you to enable sessions for use.

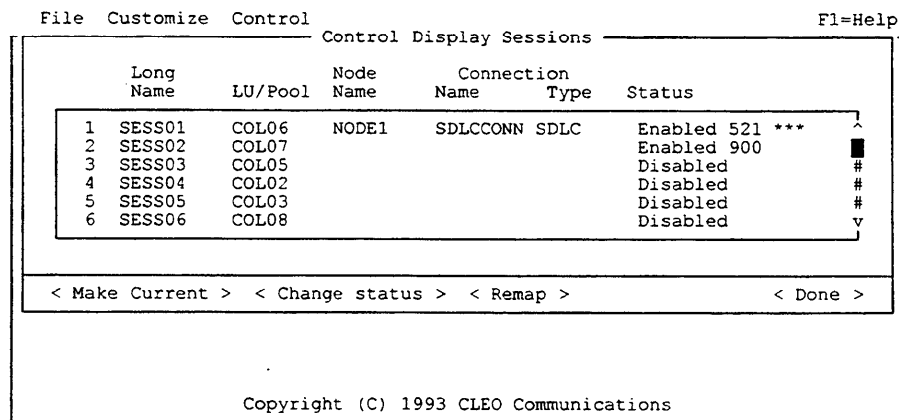
Controlling Display Sessions

The Control menu allows you to control display sessions in several different ways, the two most common uses being:

- ◆ to enable/disable sessions
- ◆ to make a session current

To control display sessions, from the Control menu select Display Sessions.

The dialog Control Display Sessions dialog is displayed.



The list box contains all the display sessions configured for you, showing whether each session is enabled or disabled. One or more display sessions can be enabled at startup, see **Customizing Display Sessions** in chapter 5 for details.

Wondering why all sessions aren't always enabled? Sessions in an LU pool, shared with others, become unavailable to anyone else when they're enabled. If all sessions are enabled, none are available to others who use the same LU pool.

The information contained in the Control Display Sessions dialog consists of

display sessionLong Name

The unique identifier for each session. See **Customizing Display Sessions** in chapter 5 for instructions on changing the Long Name.

display sessionLU / Pool

Identifies the local LU or LU pool that's used by the session. To use a different LU or LU pool, see **Remapping a Display Session**, later in this section.

display sessionNode Name

This is the name of the local node on which the LU is configured. This field is blank if the session is disabled or has not yet found an LU in the pool to which it is assigned.

display sessionConnection Name

This is the name of the connection from LINKix 3270 to the remote system.

display sessionConnection Type

Identifies the type of connection. If the session is disabled, or has not yet found an LU in the pool to which it is assigned, this field is blank.

display sessionStatus

Indicates whether the session is ready to communicate with the host or is already communicating. The following table lists and explains the possible Session Status messages. To conduct terminal emulation or to transfer files using the session, you must enable it. To enable a session, highlight it and choose < Change status >. See **Enabling a Display Session**, later in this chapter.

Status	Description
display sessionDisabled	Session is not communicating with the local node. To conduct terminal emulation or to transfer files, you must enable this session by highlighting it and choosing < Change status >.
display sessionEnabled <i>nnn</i>	Session is communicating with the local node but not with a host system. The variable <i>nnn</i> is the communications check code, displayed on the status line of the 3270 session; see

Controlling 3270 Emulation

appendix B for explanations of these codes. The code indicates why LINKix 3270 has not been able to contact the host system.

The session status may also appear as “Enabled” without a communications check code for a few seconds before it changes to SSCP (see below).

Session is activated and communicating with a host system.

Host App Name Session is communicating with the named host application.

Three asterisks (***) to the right of this column mark the display session current session. To make a session current, you must first enable it, then highlight it and choose < Make Current >. See **Making a Session Current**, later in this chapter.

< Make Current >

Choose this pushbutton to make the highlighted display sessionmaking current. This makes it the session you go to first when you begin or resume 3270 emulation. You can make a session current only if it is enabled, and if its screen model is one that your terminal can display. If you try to make a disabled session current, a message box will warn you that you cannot. In that case, choose< OK > to acknowledge the message, then enable the session before making it current. See Enabling a Display Session and Making a Session Current, later in this chapter.

< Change status >

Choose this pushbutton to enable a disabled session or to disable an enabled or active . An active session is one that is communicating with the host system or with a host application. In the Control Display Sessions dialog, an active session’s status appears as SSCP or as the name of a host application. See Enabling a Display Session, later in this chapter.

< Remap >

Choose this pushbutton to change the LU that is mapped to the highlighted . See Remapping a Session, later in this chapter. This option is disabled (grayed) if the session is active or if your system administrator has not given you permission to remap sessions.

< Done >

Choose this pushbutton when you have finished controlling display sessions.

Enabling a Display Session

You can conduct 3270 emulation only on an active session, that is, one you have enabled and that is communicating with the host system or a host application. To toggle a display session's status from disabled to enabled, take the following steps:

1. Select a display session in the list box.
2. Choose < Change status >. If LINKix 3270 is configured correctly, enabling a session will cause the Status message to change from Disabled to Enabled, then to SSCP and, when you have logged on to the host, to the name of the host application.

Making a Session Current

The current is the one you move to when you press **ESC** at the main menu (or choose Return to Emulator from the File menu). You must enable a session before you can make it current.

If a session uses a screen model which your terminal cannot display (for example, if your terminal has a 24 x 80 screen and the session uses Model 4 which is 43 x 80), you can enable the session, but you cannot make it the current session because it cannot be displayed on the screen when you return to emulation. (You can still access the session using a HLLAPI application, even though you cannot display it on the screen.)

To make a session current:

1. Select an enabled session in the list box.
2. Choose < Make Current >. Three asterisks (***) appear to the right of the Status column to mark the session as current.

If you disable the current display session, the next enabled display session (with a screen model which can be displayed) becomes the current session automatically. If no display sessions with suitable screen models are enabled, no session is current.

Moving Between Active 3270 Sessions

To move to another active 3270 when in emulation mode, press the SESS n session selection key which corresponds to the session you want, or continue pressing

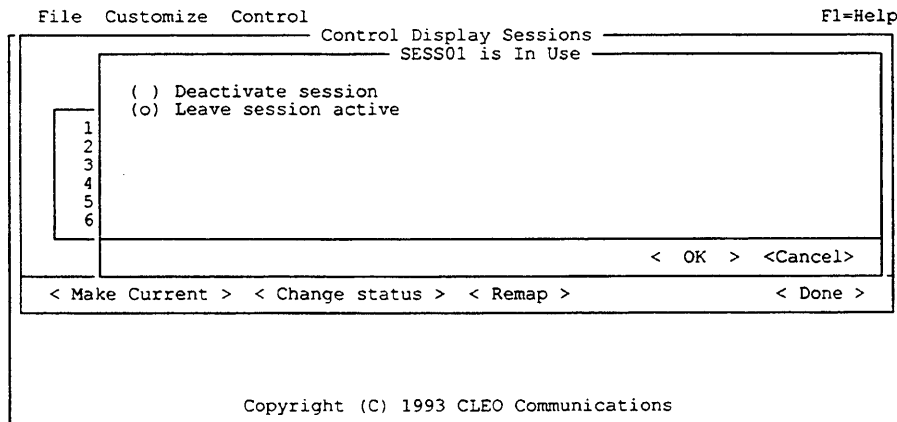
Controlling 3270 Emulation

NEXTSESS (which moves to the next active session) until you reach the session you want. See Appendix A for the default mapping of these keys on your terminal.

Disabling a Session

To disable an enabled , highlight the session in the Control Display Sessions dialog and choose < Change status >. Status changes to Disabled.

If you try to disable an active display session, the dialogsSession Is In Use dialog is displayed.



To respond to this dialog:

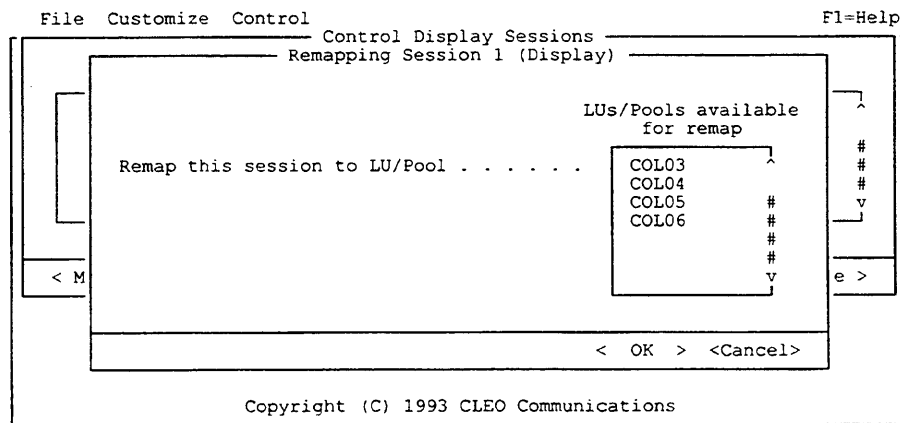
1. Select the appropriate radio button: Deactivate Session or Leave Session Active.
2. Choose < OK >. The Control Display Sessions dialog is displayed.
3. Select < Done > to return to the main menu.

Remapping a Display Session

If your package supports this option and your system administrator has configured it for your use, you can remap a display session to use a different LU or LU pool. The display session must be disabled before you can remap it — if the session is disabled but you can't remap it, your package does not support this option or your system administrator has not given you permission to remap sessions.

To remap a display session:

1. In the Control Display Sessions dialog, select a disabled display session from the list box.
2. Choose < Remap >. The dialogRemap Display Session dialog is displayed.



The list box shows all the display LUs and LU pools which you can remap sessions to (this remap list is configured for you in the configuration file).

3. Select the new LU or LU pool for this session from the list box.
4. Choose < OK > to remap the session or < Cancel > to abandon.

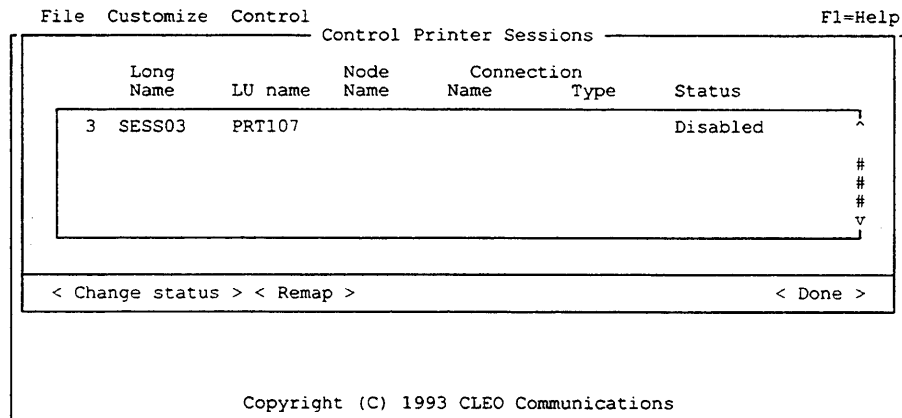
Controlling 3270 Emulation

Note: If you remap the session to an LU or pool which uses a screen model that your terminal cannot display (for example, if the LU is model 3 (32 x 80) and your terminal has a 24 x 80 screen), then you will not be able to use the session for 3270 emulation. However, you can still use it for file transfer or for running HLLAPI applications.

Controlling Printer Sessions

Printer sessions are used by host applications which can print to 3287 printers using LU types 1 or 3. You controls for much the same reason and in much the same way as you control display sessions.

To control printer sessions, from the Control menu choose Printer Sessions. The dialogControl Printer Sessions dialog is displayed.



The information and options in the Control Printer Sessions dialog consist of:

Long Name

The unique identifier for each session. See Customizing Printer Sessions in chapter 5 for instructions on changing this option.

LU

Identifies the local LU assigned to the session.

Controlling 3270 Emulation

Node Name

This is the name of the local node on which the LU is configured. This field is blank if the session is disabled.

Connection Name

Name of the connection from LINKix 3270 to the remote system.

Connection Type

Identifies the type of connection. If the session is disabled, this field is blank.

Status

Indicates whether the session is communicating with the local node. The following table lists and explains the printer session status messages. To use the session for printer emulation, you must enable it. See the next section, Enabling a Printer Session.

Status	Description
Disabled	Session is not communicating with the local node. To enable this session, you must choose Change Status.
Enabled <i>nnn</i>	Session is communicating with the local node but not with a host control program. The variable <i>nnn</i> is the communications check code, displayed as “-+z <i>nnn</i>” on the status line of the 3270 session; see appendix B for explanations of these codes. The code indicates why LINKix 3270 has not been able to contact the host control program. The session status may also appear as “Enabled” without a communications check code for a few seconds before it changes to SSCP (see below).
SSCP	The session is activated and communicating with a host control program.
Host App Name	The 3270 emulation program is communicating with the named host application, but printing has not yet started.
Printing	The 3270 emulation program is communicating with a host application and printing is in progress.

< Change status >

Choosing this pushbutton enables a disabled or disables an enabled session. See the next section, **Enabling a Printer Session**.

< Remap >

Choosing this pushbutton changes the LU that is used by the highlighted. See **Remapping a Printer Session**, later in this chapter.

< Done >

Choosing this pushbutton exits you from the Control Printer Sessions dialog.

Enabling a Printer Session

To conduct printer emulation, you must enable a printer session. To toggle a session's status disabled to enabled:

1. Select a printer session in the Control Printer Sessions dialog.
2. Choose < Change status >.

Once a printer session is enabled, you can direct 3270 output to it. This output may be sent to a printer attached to your UNIX computer or to a UNIX file. No display function is associated with printer sessions.

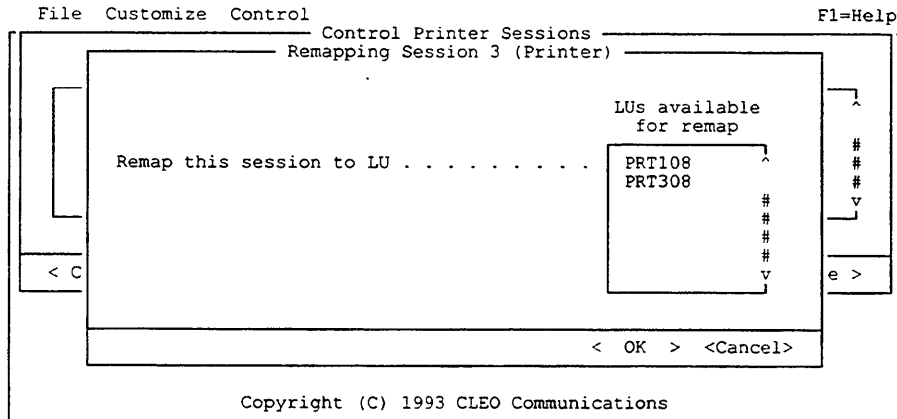
Remapping a Printer Session

If your package supports this option and your system administrator has configured it for your use, you can remap a printer session to use a different LU. The session must be disabled before you can remap it — if the session is disabled but you can't remap it, your package does not support this option or your system administrator has not given you permission to remap printer sessions.

To remap a printer session:

1. In the Control Printer Sessions dialog, select a disabled printer session from the list box.
2. Choose < Remap >. The dialog Remap Printer Session dialog is displayed.

Controlling 3270 Emulation



The list box shows all the printer LUs which you can remap sessions to (this remap list is configured for you in the configuration file).

3. Select the new LU for this session from the list box.
4. Choose < OK > to remap the session or < Cancel > to abandon remapping.

Maximum Print Position for LU1 Printer Sessions

When the host sends data for a printer session, it can specify the maximum number of characters to be printed on a line — the “maximum print position.” The default value is 132; LINKix 3270 will accept any value from the host up to 132, but will reject larger values.

If you need to accept larger values specified by the host, or to restrict the host to values below 132, you need to set the environment variable `SNALUIMPP` before starting the 3270 emulation program. This should be set to the maximum number of characters acceptable per line (the limit is 158). LINKix 3270 will then accept any value from the host which is less than or equal to the value of this environment variable, but will reject larger values. Refer to your UNIX system documentation for details on setting environment variables, if you’re not sure how to set `SNALUIMPP`.

Transferring Files

LINKix 3270 provides two ways to transfer files between a host and the UNIX computer:

- ◆ Using the 3270 emulation program's dialogs, under the Control menu
- ◆ Entering file transfer commands at the command prompt of a separate terminal (or a shell accessed on the same terminal by using the TEMPEXIT function) while the 3270 emulation program is running

Chapter 4, **Transferring Files**, explains how to use each method.

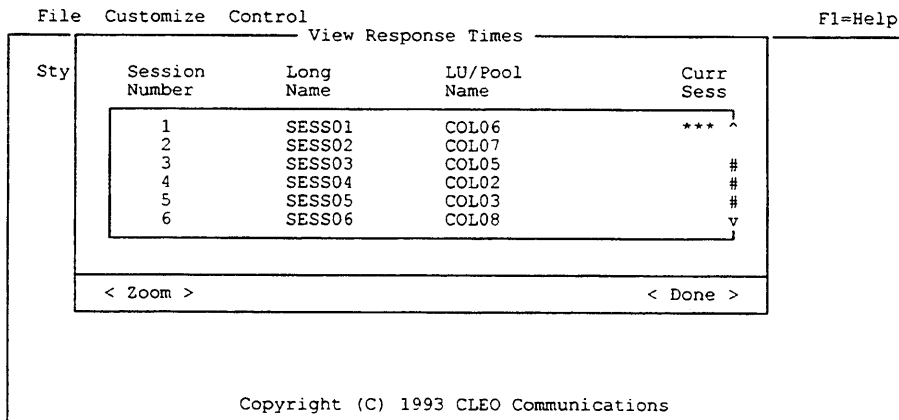
Viewing Host Response Times

This feature is available in LINKix 3270's remote SNA product only. If you cannot access this feature, either you're using a non-SNA package or your system administrator has not configured LINKix 3270 to provide it.

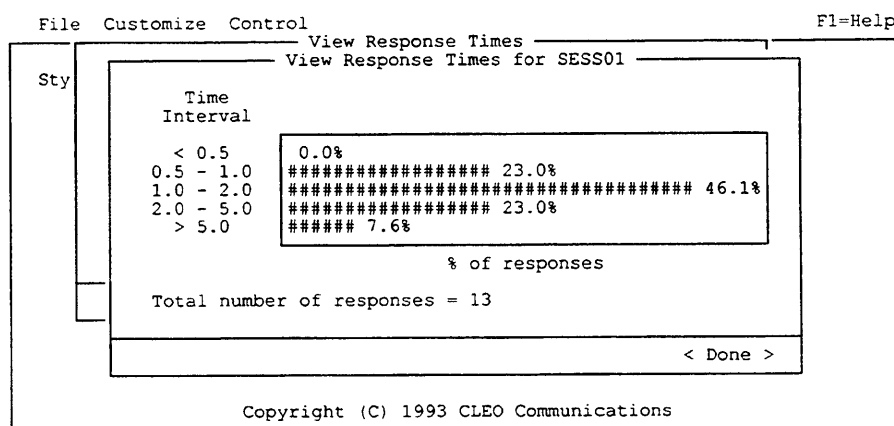
LINKix 3270 tracks how long the host computer takes to respond to each request from you — the time that elapses after you send a request to the host until the host responds. These host response times are recorded for each display session.

To view a chart showing the distribution of response times for your display sessions:

1. From the Control menu, choose Response Times. If this option is disabled (grayed), your system administrator has not configured LINKix 3270 to provide it, or you are using one of LINKix 3270's non-SNA packages. The dialogsView Response Times dialog is displayed.



2. Highlight a session and choose < Zoom >. The dialogsView Response Times for Session dialog is displayed.



This dialog depicts a series of time intervals, measured in seconds, and the percentage of host responses that occurred within each interval.

3. Choose < Done > to return to the View Response Times dialog.
4. Choose < Done > to return to the main menu.

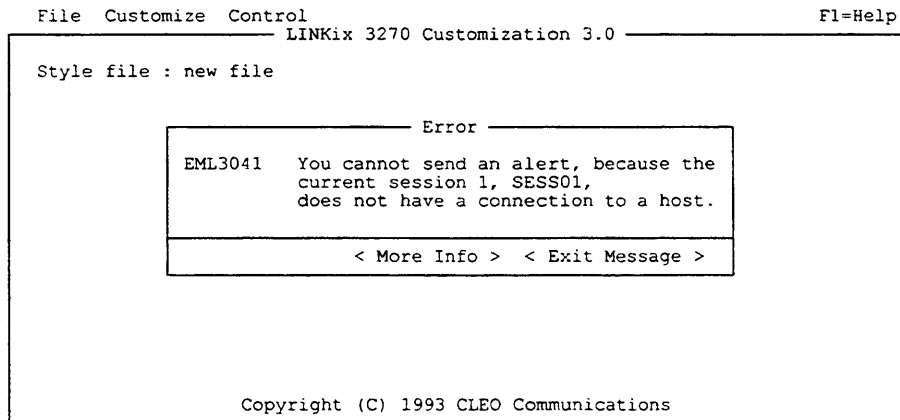
For more information on host response times, see the *Administration Guide*.

Sending NetView User Alerts

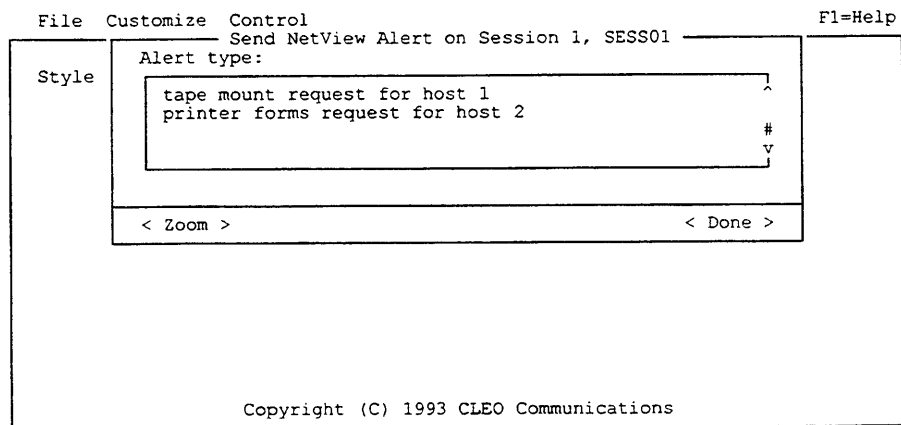
This feature is available in LINKix 3270's remote SNA product only. If you cannot access this feature, either you're using a non-SNA package or your system administrator has not configured LINKix 3270 to provide it.

To send to operators at the host computer:

1. Make sure the session over which you want to send the alert is the current session. See **Making a Session Current**, earlier in this chapter.
2. From the Control menu, choose NetView Alerts. If this menu option is disabled (grayed), it's not available because you're using a non-SNA package or your system administrator has not configured LINKix 3270 to provide this capability.
3. If you see an error message indicating that you do not have an enabled session, a valid SSCP-PU session or an appropriate connection, make sure that the appropriate session is enabled and communicating with the host.



If you have a valid SSCP session, the dialogSend NetView Alert dialog is displayed.



4. Highlight the alert you want to send from Alert Type list box and choose < Zoom >.

Controlling 3270 Emulation

5. A Send NetView Alert dialog (similar to the one shown below) is displayed. The text which indicates the meaning of each parameter is set up by your UNIX system administrator. For details, consult the *Administration Guide* or contact your UNIX system administrator.

File Customize Control F1=Help

Send NetView Alert on Session 1, SESS01

Send NetView Alert: tape mount request for host 1

tape number [.....]

date [.....]

Not used [.....]

Message Text:
[.....]

< Send > <Cancel>

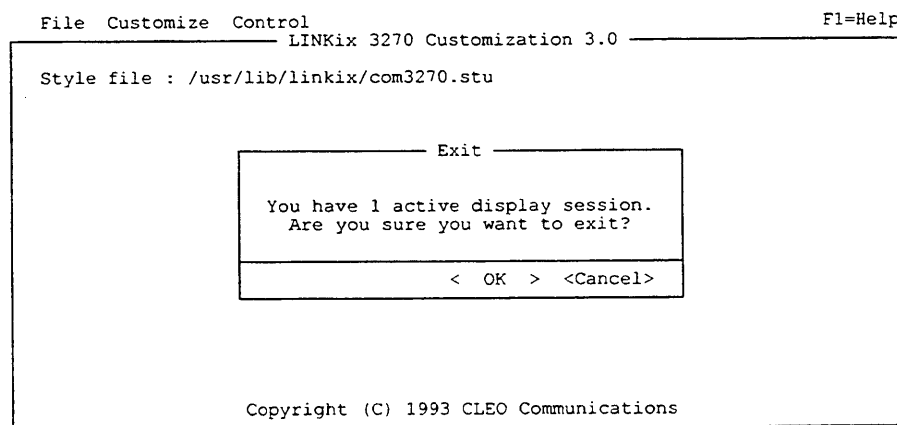
Copyright (C) 1993 CLEO Communications

6. Enter the information requested, including a free-form message if you wish.
7. Choose < Send > to send the alert or < Cancel > to abandon sending the alert.
8. A message box appears giving details of the alert sent. Choose < OK > to acknowledge the message.

Exiting 3270 Emulation

To exit the 3270 emulation program and return to the UNIX command prompt, follow these steps:

1. Log off any active sessions. (This is recommended but not always necessary. LINKix 3270 can terminate active sessions if you exit emulation without logging off. However, with some host applications this may cause problems when you next try to log on.)
2. From the File menu, choose Exit. If any sessions are active when you choose this option, the Exit message box is displayed:



3. Choose < OK > to exit and deactivate any active sessions, or choose < Cancel > to resume emulation.

If you've made any changes to your 3270 emulation style parameters and have not saved them, LINKix 3270 prompts you to save them before exiting. See **Opening, Creating and Saving Style Files**, in chapter 5, for more information.

4. Transferring Files

LINKix 3270 provides two methods for transferring files between your UNIX computer and a host computer: using the menus and dialogs of the 3270 emulation program or by entering commands at the UNIX command prompt. This chapter explains how to transfer files using the 3270 emulation program in, and host environments. It provides the following:

- ◆ General instructions on preparing to transfer files
- ◆ Instructions for using the menu options to transfer files
- ◆ Instructions for command line file transfer
- ◆ Fast Path to file transfer

IND\$FILE support required: When you transfer files between a host and your UNIX computer, the host runs the IBM\$FILE host file transfer program (IBM product numbers: #5665-31 1 for TSO; #5663-281 for CMS; #5789-DQH for CICS). If the name of this host program is not IND\$FILE, you must set the IND_FILE to the correct program name before running the emulation program (the host program must still be IND\$FILE-compatible). Refer to your UNIX documentation or contact your system administrator.

If you're an experienced user and wish to start immediately, see **Fast Path to File Transfer** at the end of this chapter; otherwise, read the detailed instructions that follow.

Write Structured Fields (WSF) Support

The file transfer programs, *cmnsend* and *comreceive*, use the faster style of file transfer, sometimes referred to as DFT-style file transfer. In order to transfer files with these programs, the host's definition for your terminal (i.e., the logmode table entry) must indicate support for WSF. If the host does not support WSF file transfers, using *comsend* or *comreceive* from the command line (or using the menu system which calls *comsend* or *comreceive*) will cause your 3270 session to enter a constant WAIT state. To clear this condition, you have to disable and enable the 3270 session using the Control menu.

If you encounter a problem with WSF file transfer support, your logmode entry may not be properly set. It may be possible for you to dynamically redefine your default logmode table entry at logon time. Contact your host administrator for details.

Preparing to Transfer Files

Before starting to transfer files, you need to start the 3270 emulation program, enable at least one session and logon to the host. You may also need to set the file transfer maximum structured field size for the session you are using for the file transfer (see **Customizing Display Sessions** in chapter 5). You can then check to see whether or not you're ready to begin transferring files by accessing the dialogsFile Transfer dialog in one of two ways:

- ◆ From the Control menu choose File Transfer
- ◆ From an active 3270 session press FILE XFR.

File	Customize	Control	File Transfer			F1=Help
Session Number	Long Name	LU/Pool	Session Status	Transfer Status	Bytes Transferred	
1	SESS01	COL06	TS01011	In progress		^
2	SESS02	COL07	Disabled	Unavailable		
3	SESS03	COL05	Disabled	Unavailable		#
4	SESS04	COL02	Disabled	Unavailable		v
Messages for file transfer with SESS01						
File transfer requested						^
						v
< Host Parameters > < Start transfer > < Abort transfer > < Done >						
Copyright (C) 1993 CLEO Communications						

Information in this dialog, indicates the status of your file transfer activity.

Available Sessions

If the file Transfer Status for the session you want to use is available, you are ready to set host parameters and begin transferring files. Proceed with the section **Transferring Files Using the Menu Options**.

Transferring Files

Unavailable Sessions

If the session you want is Unavailable, check its session status in the File Transfer dialog. Then do one of the following:

- ◆ If the session is disabled, use the Control Menu to enable the session, then logon to the host.
- ◆ If Session Status is Enabled *nnn*, correct the communications problem (see the 3-digit communications check code *nnn*), then logon to the host. See Appendix B, **Status Line Information**, for details of the check codes.
- ◆ If session Status is SSCP, logon to the host.

To logon to the host, take the following steps (steps 1-4 are not required if the session you want to use is already the current session):

1. From the File Transfer dialog, press **ESC** to move to the main screen.
2. From the Control menu, choose Display Sessions.
3. Make the session you want to use for file transfer the current session by highlighting the session and choosing < Make Current >.
4. Choose < Done > to return to the main screen.
5. Press **ESC** to move to the current session.

As an alternative to steps 1-5 above, press **ESC** twice to move to the current session. Then move to the session you want to use for file transfer by pressing the key combination that represents the session, or by using the NEXTSESS key combination until you get to the session you want. See Appendix A, **Default Keyboard Mappings**, for a list of Session Selection key combinations.

6. Follow the logon instructions from the host.
7. Press FILE XFR to return to the File Transfer dialog and confirm that the desired session is now available.

When communications are established (you are logged into the host and at its command prompt), you can begin transferring files. Follow the instructions in the next section, **Transferring Files Using Menu Options**.

Transferring Files Using Menu Options

This section describes how to files using the menu options. The first part provides step-by-step instructions for setting host file transfer parameters, with notes on the differences for different host types; there is then a section on starting the file transfer for each host type. For instructions on transferring files using the command line option, see the next section **Command Line File Transfer**.

Setting Host File Transfer Parameters

Before starting a file transfer, you must set host file transfer parameters, which identify your host's operating system and specify the size and format of files to be transferred. You can set these parameters independently for each 3270 session. File transfer parameters are stored in your 3270 style file; if you wish to keep the changed parameters for future use, use the File menu to save the 3270 style (see chapter 5 for more information).

To set host file transfer parameters:

1. From the File Transfer dialog, highlight the session you want to use.
2. Choose < Host Parameters >. The dialogOptions for session dialog is displayed.

File Customize Control File Transfer Options for SESS01 F1=Help

Host : (o) TSO () VM/CMS () CICS	Host record format:
[] EBCDIC <-> ASCII conversion	(o) Default
Host code page . . . [...]	() Fixed
Local code page . . . [...]	() Variable
[] Strip/Add LF from each line	() Undefined
[] Add to end of existing file	
Host record length [.....]	
Host block size [.....]	
Units of space: () Tracks () Cylinders (o) Blocks	
Blocks: Average block size for file (bytes) . . . [.....]	
Initial units of space [.....]	
Incremental units of space [.....]	
< OK > <Cancel>	

Transferring Files

3. Modify the file transfer parameters as appropriate.

Host

This radio group specifies your host type: TSO, VM/CMS or CICS. Depending on the host type you choose, some of the remaining parameters may not be required. Any parameters not applicable are disabled (grayed). In addition, some parameters are used only when sending files or only when receiving files and are ignored otherwise.

EBCDIC <-> ASCII conversion

This option should be used only for text files. Mark the check box to convert files between EBCDIC format at the host and ASCII format at the UNIX computer. When choosing this option, default values appear for the code pages used for the translation (öHost code page is the EBCDIC code page used by the host and öLocal code page is the ASCII code page used by your UNIX computer). If you need to use a different host code page or local code page, specify the correct ones here.

The code pages supported by LINKix 3270 and the variants of ASCII or EBCDIC which use each code page are listed below. The ASCII code page 8859 is valid only if your UNIX system supports the ISO 8859 character set, and you started the 3270 emulation program with the -i (international) option. The other ASCII code pages and host code pages are always valid.

ASCII code pages

8859	Generalized ASCII code page defined by ISO 8859, used to support all language variants (only available on versions of UNIX which include ISO 8859 internationalization support)
437	US English
850	International code page (contains characters for US English, UK English, French, German, Italian, Spanish, Finnish, Netherlands, Swedish, Swiss, Belgian, Latin American)
865	Danish, Norwegian
860	Portuguese
863	Canadian French

EBCDIC code pages

037	US English, Canadian Bilingual, Netherlands, Portuguese
273	German
277	Danish, Norwegian
278	Finnish, Swedish
280	Italian
284	Spanish, Latin American
285	UK English
297	French
500	Belgian (New), Swiss French, Swiss German

The default host code page is 037; the default local code page is 8859 if your version of the 3270 emulation program supports host language selection and 437 if it does not.

Strip/Add LF from each line

This check box specifies whether or not linefeed (LF) characters should be removed from a text file sent to the host or added to a text file received from the host. This is recommended for text files which need to be readable on the host. Do not select this option for binary files.

Add to end of existing file

This check box indicates whether or not the transferred data is to be added to the end of an existing target file. For a CICS host, this option is valid only for Receive. If you do not select this option, an existing target file is overwritten.

Host record format

This radio group specifies how individual records are structured: Default, Fixed, Variable or Undefined. Choose Default unless you are sending data to an application which requires a specific record format. The value Undefined is valid only for TSO.

Transferring Files

Host record length

If you specified Host record format as Fixed, this field specifies the length (in bytes) of the records to be created. If you specified Host record format as Variable, this field specifies the maximum length (in bytes) of the records to be created. The default, in both cases, is 80 bytes.

Host **block size**

A TSO host groups records into blocks; the default block size is the same as the record length (one record per block). If you need to block records into larger groups, specify the block size you want to use.

Units of space

This radio group specifies the unit of measure used to define the size of the file to be created:

Tracks	Specifies tracks as the unit of measure for record size.
Cylinders	Specifies cylinders as the unit of measure for record size.
Blocks	Specifies blocks as the unit of measure for record size. The size of the blocks is given by the Average block size for file parameter.

Average block size for file (bytes)

If you specified Blocks as the Units of space parameter, this field specifies the size of the blocks in bytes.

Initial units of space

This field specifies the amount of space (in blocks, cylinders or tracks, as per the Units of space parameter) that you want to allocate for the host data set.

Incremental units of space

This field specifies the number of additional units of space to allocate if the amount specified in Initial units of space is insufficient to hold the file.

4. When the parameters are set as needed, choose < OK > to record them, or choose < Cancel > or **ESC** to abandon any changes. You return to the File Transfer dialog, where you are ready to send or receive files. Follow the appropriate section in **Transferring Files** for your host type.

A Note on Text versus Binary Transfer

If the file you are sending/receiving contains only readable text and you want it readable on the destination system (either the host or UNIX computer), then select both the EBCDIC<E>ASCII conversion and Strip/Add LF options. If, however, you want to transfer files that contain unprintable characters (such as programs, database files, spreadsheets, etc.), *do not select the EBCDIC<E>ASCII or Strip/Add LF options*, as they alter the contents of a file and will make one of this type unusable.

The following table provides a quick reference on when and for which host types you need to set parameters for a particular file transfer.

Parameter	TSO	Host W M/CMS	CICS
Host	X	X	X
EBCDIC<->ASCII conversion	X	X	X
Strip/Add LF from each line	X	X	X
Add to end of existing file	X	X	R
Host record format	S	S	—
Host record length	S	S	—
Host block size	S		—
Units of space	S		—
Average block size	S		
Initial units of space	S		—
Incremental units of space	S		—

Key:

- X This parameter is used for both Send and Receive.
- This parameter is not used for this host type (the field is disabled - grayed).
- R This parameter is used for Receive only (ignored for Send).
- S This parameter is used for Send only, when creating a new host file. It is ignored for Receive or when replacing or appending to an existing host file.

Transferring Files

Transferring Files

To file transfer files, highlight an active session in the File Transfer dialog and choose < Start Transfer >. (An error message is displayed if you try to start transfer on an inactive session.)

You cannot communicate with the host on this session, or transfer another file, while the file transfer is in progress; the WAIT message is displayed on the status line until the transfer is ended. Appendix B has details on status line messages.

Depending on the Host type that you've specified, one of three different dialogs will be displayed.

MVS/TSO Host

If you selected the MVS/TSO Host type radio button, the dialog Transfer TSO File on session is displayed.

File Customize Control F1=Help

Transfer TSO file on SESS01

(o) Send () Receive

Local file name . . . [.....]

Host dataset . . . [.....]

Member name . . . [.....]

Password . . . [.....]

< OK > <Cancel>

To complete the file transfer:

1. Choose the appropriate radio button: Send, to transfer a UNIX file to the host; Receive, to transfer a host data set to the UNIX computer.

2. Make whatever changes are necessary to the following fields:

Local file name

This field should contain the full name, including the full directory path (not a relative path) if necessary, of the UNIX file sent to the host or the UNIX file receiving data from the host.

Host dataset

This field should contain the name of the data set to which a UNIX file is sent, or the data set from which data is received.

Member name

This field is optional. If the host data set is a partitioned data set, this is the member within that data set to which the file is sent or from which data is received.

Password

This field is optional. If the host data set is protected by a password, specify the password here.

3. Choose < OK > to transfer the file. An information message box appears which indicates the file transfer has started.

File Customize Control
File Transfer
F1=Help

Session Number	Long Name	LU/Pool	Session Status	Transfer Status	Bytes Transferred	
1	SESS01	COL06	TS01011	Available	^	
2			Information			#
3			EML3057 TSO file transfer started on SESS01. Sending /usr/lib/linkix/new.stu.			v
4						v

Messages f
< Exit Message >

< Host Parameters >
< Start transfer >
< Abort transfer >
< Done >

Copyright (C) 1993 CLEO Communications

Transferring Files

- 4 . Choose < Exit Message > to acknowledge this message.
- 5 . To monitor progress of the, refer to the File Transfer dialog and read the information under Session Status, Transfer Status and Messages for File Transfer with *Session*.

VM/CMS Host

If you selected the VM/CMS Host type radio button, the dialog Transfer VM/CMS File on *session* dialog is displayed.

File Customize Control F1=Help

File Transfer

Transfer VM/CMS file on SESS01

(o) Send () Receive

Local file name . . . [.....]

Host filename . . . [.....]

Host filetype . . . [.....]

Host filemode . . . [.....]

< OK > <Cancel>

To complete the file transfer:

- 1 . Choose the appropriate radio button: Send, to transfer a UNIX file to the host; Receive, to transfer a host data set to the UNIX computer.
- 2 . Make whatever changes are necessary to the following fields:

Local filename

This field should contain the full name, including the full directory path (not a relative path) if necessary, of the UNIX file sent to the host or the UNIX file receiving data from the host.

Host filename

This field specifies the host file to which data is sent or the file received from the host.

Host filetype

This field specifies the VM/CMS filetype of the host file.

Host filemode

This field specifies the VM/CMS filemode of the host file. This parameter is optional; if you do not specify it, the default A1 is used.

3. Choose < OK > to transfer the file. An information message box appears which indicates the file transfer has started.

File Customize Control
File Transfer
F1=Help

Session Number	Long Name	LU/Pool	Session Status	Transfer Status	Bytes Transferred
1	SESS01	COL06	VCNA	Available	~
2	Information				
3					#
4	EML3057	VM/CMS file transfer started on SN1. Sending /usr/lib/linkix/new.stu.			v

Messages f

File trans
< Exit Message >

File trans

File transfer finished
v

< Host Parameters >
< Start transfer >
< Abort transfer >
< Done >

Copyright (C) 1993 CLEO Communications

4. Choose < Exit Message > to acknowledge the message.
5. To monitor progress of the file transfer, refer to the File Transfer dialog and read the information under Session Status, Transfer Status and Messages for File Transfer with *Session*.

Transferring Files

CICS Host

If you selected the CICS Host type radio button, the dialog Transfer CICS File on session is displayed.

File Customize Control File Transfer F1=Help

Transfer CICS file on SESS01

() Send (o) Receive

Local filename [.....]

CICS filename [.....]

Comments [.....]

< OK > <Cancel>

To complete the file transfer:

1. Choose the appropriate radio button: Send, to transfer a UNIX file to the host; Receive, to transfer a host data set to the UNIX computer.
2. Make whatever changes are necessary to the following fields:

Local filename

This field should contain the full name, including the full directory path (not a relative path) if necessary, of the UNIX file sent to the host or the UNIX file receiving data from the host.

CICS filename

This field specifies the host file to which data is sent or the file received from the host.

Comments

In this field you can add a comment that will be sent when sending a file; it is written to the first record of the host file. Comments are not valid when receiving a file. The comment may be up to 80 characters, although some hosts may impose a lower limit.

3. Choose < OK > to transfer the file. An information message box appears which indicates the file transfer has started.

File Customize Control
File Transfer
Fl=Help

Session Number	Long Name	LU/Pool	Session Status	Transfer Status	Bytes Transferred
1	SESS01	COL06	TSO1063	Available	^
2			Information		#
3					v
4	EML3057		CICS file transfer started on SESS1. Sending /usr/lib/linkix/new.stu.		

Messages f

TRANS34 CM
< Exit Message >

File trans
 File transfer finished

< Host Parameters >
< Start transfer >
< Abort transfer >
< Done >

Copyright (C) 1993 CLEO Communications

4. Choose < Exit Message > to acknowledge this message.
5. To monitor progress of the file transfer, refer to the section **Monitoring File Transfer Status** (p. 4-23).

Command Line File Transfers

You can also transfer files between LINKix 3270 and the host by issuing commands from the UNIX command prompt during 3270 emulation. You do this from another shell on the same terminal from which you started the 3270 emulation program (using the TEMPEXIT keystroke) or from another terminal (in which case you must log in with the same user ID as you used for the 3270 emulation program). For command line file transfers, the session you are using must have a session ID.

Most of the parameters you enter with the commands are common to all host environments, while others apply to a particular host environment. The format of the command for each host type is given below, and the following paragraphs explain the features of each command.

You can also issue commands with the 3270 emulation program running in the background (see **Running 3270 Emulation in the Background**, in chapter 2, for more information). However, you will not be able to perform the logon sequence to the host (since you cannot access the 3270 display directly when the 3270 emulation program is running in the background). You need a HLLAPI application to perform the logon. This application can be the HLLAPI terminal emulation program that we provide with this package (see Appendix D for details) or one that you create which logs onto the host and then disconnects to leave the session available for other applications. You run this application after starting the 3270 emulation program and before issuing the file transfer command. See the sample program in the *HLLAPI Programmer's Guide* for an example of this type of application; note, however, that our sample will almost surely require modification to suit your particular host's logon sequence.

Note that characters such as parentheses and quotation marks must be preceded by a backslash (\) to prevent interpretation by the UNIX shell. This may also apply to some special characters in host file names (for example \$). The backslash is shown in the following syntax where it is required.

For information on the valid parameters and options for a particular host, you can type either `comsend` or `comreceive` followed by the host type (TSO, VM or CICS). This will generate help; messages which explain the syntax for the particular host type. Messages are also generated if LINKix 3270 detects syntax errors in the file transfer command (i.e., not enough parameters supplied or incorrect parentheses).

MVS/TSO

```
comsend [sessionID: | -h hexID] UNIXfile  
[\ ' ]datasetname[\ (membername\)][\ ' ] [/password] [options. ..]  
comreceive [sessionID: | -h hexID] UNIXfile  
[\ ' ]datasetname[\ (membername\)][\ ' ] [/password] [options. ..]
```

VM/CMS

```
comsend [sessionID: | -h hexID] UNIXfile hostfilename filetype  
[filemode] [\ (options. ..]  
comreceive [sessionID: | -h hexID] UNIXfile hostfilename filetype  
[filemode] [\ (options. ..]
```

CICS

```
comsend [sessionID: | -h hexID] UNIXfile hostfilename \ (options. . .  
[\ )comments\  
comreceive [sessionID: | -h hexID] UNIXfile hostfilename \ (options. . .  
[\ )coments\  
[
```

Command Line File Transfer Parameters

The consists of a number of parameters, with optional items enclosed in brackets ([...]) and parentheses preceded by a backslash (\). When entering a file transfer command, do not include the brackets when including an optional item.

Command or Parameter	Description
comsend or comreceive	These are the programs that send a file to the host and receive a file from the host. This portion of the command must be typed in lowercase letters.
UNIXfiie	This is the name of the UNIX file sent to the host or the UNIX file to which data received from the host is written. Include the directory path if necessary; you can use a path relative to the current directory.

Transferring Files

sessionID: -h hexID	The sessionID: is the long or short name of an active 3270 display session. It must be in uppercase characters and followed by a colon. The session must have a session ID, even if you use the long name here. If you have used the -h option when starting c3270 to remap session IDs outside the range of AÆZ, you must use the -h option hereto designate the session (e.g., -h 0x01). If a sessionID is not specified, LINKix 3270 will find the configured session that's first in alphabetical order (by session ID), and try to use that session.
datasetname	MVS/TSO only. The host data set to which the UNIX file is sent, or the host data set from which data is received. If you use a fully qualified data set name (the first element of the name is a user ID), enclose the data set name in single quotation marks; otherwise, TSO automatically adds the logon id. If the member name is used, it too must be inside the quotation marks.
\(membername\)	MVS/TSO only. This is the member within the data set to which the UNIX file is sent, or from which data is received by the UNIX file. Enclose the membername in parentheses. Required if the host data set is a partitioned data set.
/password	MVS/TSO only. A password is required only if the data set is password-protected. It must be preceded by a slash.
hostfilename	VM/CMS or CICS. The name of the host file to which the data is sent, or the name of the host file from which data is received.
filetype	VM/CMS only. This is the VM/CMS filetype of the host file.
filemode	VM/CMS only. This is the VM/CMS filemode of the host file. If filemode is not specified, the default A1 is used.
\(options	The section, Command Line file transfer options , explains the valid options. VM/CMS and CICS require the left parenthesis before the options; MVS/TSO does not require any parentheses and, if one is included, an error results.

\)comments CICS only. Any comment information which you want to send to the host; this is written into the first record of the host file. The comment may be up to 80 characters, although some hosts may impose a lower limit. Must be preceded by a right parenthesis.

Command Line File Transfer Options

This section explains the file transfercommand line options you can use on the command line when transferring files. Italics in the option names indicate variable parameters. The options are not case-sensitive (you may use either uppercase or lowercase letters). Valid options differ according to the host type and whether you are sending or receiving files.

Option	Description
APPEND	The file being sent is added to the end of the target file or data set. If the file does not exist, it is created.
ASCII	The file is translated between ASCII (on the UNIX computer) and EBCDIC (on the host). This option should be used only for text files, not for binary files. For CICS, you must specify either this option or the BINARY option.
BINARY	CICS only; you must specify either this option or the ASCII option. This option causes the file to be transferred without translation (the opposite of the ASCII option). It should be used only for binary files, not for text files.
<i>H(nnn)</i>	Specifies the host (EBCDIC) code page for translation. If the host page is not specified, the host will use a default code page (this is normally 037 but may vary). This option is valid only if the ASCII option is used.
<i>P(nnn)</i>	Specifies the local (ASCII) code page the host should use to translate the file received from the UNIX computer. If a local page is not specified, the host uses a default code page (this is normally 437 or 850 but may vary). This option is valid only if the ASCII option is used.

Transferring Files

CRLF	Deletes linefeed characters before sending the file to the host or adds linefeed characters when receiving from the host. For CICS, you must specify either this option or the NOCRLF option.
NOCRLF	CICS only; you must specify either this option or CRLF. NOCRLF is the converse of CRLF and specifies that linefeed characters are not to be added or deleted.
T (nn)	Specifies the timeout period. File transfer is canceled if the host does not respond within this time. The variable <i>nn</i> represents the number of 30-second intervals (00-99) to wait. For example, T(02) equals 1 minute. If this option is not specified, or if a value of zero is specified, LINKix 3270 waits indefinitely for the host to respond.
BLKSIZE n	Specifies the block size of the data set (in bytes) when creating an MVS/TSO host data set. If this option is omitted, the default is LRECL.
LRECL (n)	Specifies the logical record length (if the host record format is Fixed), or the maximum length (if the host record format is Variable) when creating a new host data set or host file. If LRECL (n) is not specified, the default is 80.
RECFM (FIVIU)	Specifies the file transfer record format of the host file. (F) specifies a file with fixed-length records. (V) specifies a file with variable-length records. (U) specifies a data set with records of undefined length (this is valid only for MVS/TSO, not for VM/CMS). If this option is not specified, the default is F, unless CRLF is specified in which case it is V.
SPACE q,i unit	Specifies the space to allocate for a new data set. unit is the unit of measurement: TRACKS, CYLINDERS or AVBLK <i>n</i> (blocks of size <i>n</i> bytes). <i>q</i> is the amount of space initially allocated, in the units specified by <i>unit</i> , and <i>i</i> is the increment to add each time the space previously added is filled, in the units specified by <i>unit</i> . The increment <i>i</i> is optional. If it is not specified, and the initial space <i>q</i> is filled before the transfer is completed, the file transfer will fail. If <i>i</i> is supplied, the comma before it is required.

Q Quiet mode. This option suppresses the display of all file transfer messages. Note that /Q is an alternative format for this option.

The table below provides a quick reference to the command line parameters that can be used for each host type when transferring files.

Option	Host		
	TSO	VM/CMS	CICS
APPEND	X	X	R
ASCII	X	X	X
BINARY	—	—	X
<i>H(nnn)</i>	X	X	X
<i>P(nnn)</i>	X	X	X
CRLF	X	X	X
NOCRLF	—	—	X
<i>T(nn)</i>	X	X	X
LRECL (n)	S	S	—
RECFM FIVIU	S	S	—
BLKSIZE n	S	—	—
SPACE q,i unit	S	—	—
Q	X	X	X

Key:

- X** This parameter is used for both Send and Receive.
- This parameter is not used for this host type.
- R** This parameter is used for Receive only (ignored for Send).
- S** This parameter is used for Send only, when creating a new host file. It is ignored for Receive or when replacing or appending to an existing host file.

Note: When transferring files under CICS, you must specify either ASCII or BINARY and either CRLF or NOCRLF.

Transferring Files

Sample File Transfer Commands

The following is an example of an MVS/TSO send command:

```
comsend files.txt A: \ 'id. text.mytext\ (files\)' ASCII P\ (437\)' CRLF
```

This command sends the *filea.txt* (from the current UNIX directory) to the member *filea* of the MVS/TSO partitioned data set *id.text.mytext*, using the 3270 session with a session ID of A. The host translates the file from ASCII to EBCDIC, using the PC (ASCII) code page 437 and removes linefeed characters.

The following is an example of a VM/CMS receive command:

```
comreceive /usr/jim/file2.txt SESS2: mytext script \(\Crlf ascii t\ (03\)
```

This command receives the VM/CMS file *mytext* with filetype *script* (the filemode is not specified, so it is assumed to be *Al*) and stores it in the directory */usr/jim* as a file named *file2.txt* the host translates the file from EBCDIC to ASCII and adds linefeed characters. The file transfer uses the display session with long name *SESS2*. The transfer will be canceled if the host fails to respond within 90 seconds (three 30-second timeout periods).

The following is an example of a CICS send command:

```
comsend FILE3 mytext \(\ASCII NOCRLF\ ) Sales Meeting Report
```

This command sends the file named *FILE3* in the current directory to the CICS file *mytext*. No session name is given, so the 3270 session whose session ID is first alphabetically is used. The host translates the file from ASCII to EBCDIC, but does not remove linefeed characters. The comment *öSales Meeting Report* is written to the first record of the host file.

Monitoring File Transfer Status

You can monitor s by viewing the dialogsFile Transfer dialog.

File Customize Control F1=Help

File Transfer

Session Number	Long Name	LU/Pool	Session Status	Transfer Status	Bytes Transferred
1	SESS01	COL06	TS01011	In progress	^
2	SESS02	COL07	Disabled	Unavailable	
3	SESS03	COL05	Disabled	Unavailable	#
4	SESS04	COL02	Disabled	Unavailable	v

Messages for file transfer with SESS01

File transfer requested

< Host Parameters > < Start transfer > < Abort transfer > < Done >

Copyright (C) 1993 CLEO Communications

The fields in this dialog contain the following information.

Sess Number

The Session Number. All your display sessions are listed in this box.

Long Name

The unique identifier for each session.

LU/Pool

The identifier of the local LU or LU pool assigned to the session.

Session status

Indicates whether the session is ready to communicate with the host or is already communicating. The following table explains the possible session status messages. To transfer files on a session, you must be logged onto the host.

Transferring Files

Status	Description
Disabled	Session is not communicating with the local node. You must enable a session before performing a file transfer.
Enabled <i>nnn</i>	Session is communicating with a local node. The variable <i>nnn</i> is the communications check code, as displayed on the status line of the 3270 session. Appendix B, Status Line Messages, explains these check codes. The session status may also appear as öEnabled without a communications check code for a few seconds before it changes to SSCP (see below).
SSCP	Session is activated and communicating with a host control program.
Host App Name	Session is communicating with the named host application.

Transfer status

Indicates availability of the session for file transfer and the progress of any transfer already started. The following table lists and explains the possible Transfer Status messages.

Status	Description
Available	No file transfer is in progress. The session is active and the host is ready to receive file transfer requests.
Unavailable	The session is not enabled; or communication with the host is established, but the user is not yet logged on to the host.
Receiving	Receive operation is in progress on this session. The number of bytes already transferred is shown.
Sending	Send operation is in progress on this session. The number of bytes already transferred is shown.

Busy	File transfer operation is ending. A list box displays any file transfer messages generated by the transfer. See Appendix C, File Transfer Messages.
Complete	A file transfer has been completed. Total number of bytes transferred is shown.

Bytes Transferred

This field shows the number of characters transferred.

Messages for file transfer with Session

Messages related to the current (or most recent) file transfer are shown in the Messages list box. If there are more messages than can be displayed at one time, you can scroll to display them.

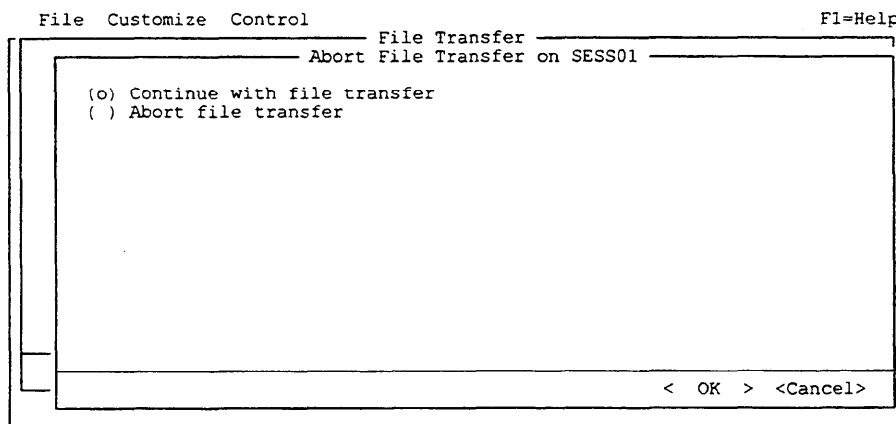
Note that some host error messages (relating to starting the IND\$FILE program on the host) are displayed on the 3270 emulation screen, and not in this dialog box. If a file transfer process appears to hang with no error message displayed, return to the 3270 emulation screen for the appropriate display session to check for any host messages.

See Appendix C for explanations of the file transfer and TRANSnnn host messages which may appear in this list box. For host systems which use other messages (such as INWnnnn), you will need to refer to your host documentation for explanations.

Stopping a File Transfer

To stop a that's in progress:

1. Select the session on which the file transfer is in progress.
2. Choose < Abort Transfer >. The dialogAbort File Transfer dialog is displayed.



3. Choose the appropriate radio button:
 - ◆ Continue with file transfer: cancels the termination and continues with the file transfer.
 - ◆ Abort file transfer terminates the transfer.
4. Choose <OK>.

Finishing File Transfer

File Transfer Successful

When a file has been transferred successfully, the File Transfer dialog indicates so. The transfer status changes to Complete and the size of the transferred file is listed under Bytes Transferred.

When you are finished ring files, choose < Done > or press **ESC** to leave the File Transfer dialog. You return to the main screen of the control interface if you moved to the File Transfer dialog from the control interface, or to the current 3270 emulation session if you moved to the File Transfer dialog from 3270 emulation.

File Transfer Unsuccessful

If the file transfer fails, the 3270 session may enter a constant WAIT state (keyboard locked). To clear this condition, you have to disable and enable the 3270 session using the Control menu. See Controlling Display Sessions in chapter 3, **Controlling 3270 Emulation**, for details.

Fast Path to File Transfer

This fast path is designed for experienced users and provides a brief outline of how to transfer files. For more detailed instructions, read the appropriate section(s) earlier in this chapter.

To begin file transfer between your UNIX computer and host computers, follow these steps:

1. Prepare for 3270 emulation.
 - a. Make sure that you are configured as a 3270 user, and that the LINKix 3270 software is started. See your System Administrator if necessary.
 - b. Also, make sure that the host and your UNIX computer agree on the name of the file transfer program, normally `IND$FILE`, to be executed by the host. LINKix 3270 assumes the name is `IND$FILE` by default. Use the `IND_FILE` environment variable to change it if necessary (and remember to export the variable).
2. Start the program.

You start the emulation program by entering the following command at the UNIX command prompt:

```
c3270 [<options>]
```

See chapter 2 for details on all available options.

3. Enable one or more 3270 display sessions (if no sessions are automatically enabled at startup).
 - a. From the Control menu, choose Display Sessions.
 - b. Enable a session. To do so, highlight it in the Control Display Sessions dialog and choose `< Change Status >`. Status will change from Disabled to Enabled and then to SSCP if the system is configured correctly.
- Repeat this step until you have enabled all the sessions you need.

- c. Make the session you want to use current by highlighting it and choosing < Make Current>.
 - d. Choose < Done> to return to the main menu.
 - e. Press **ESC** to move to the 3270 session.
 - f. Log onto the host.
 - g. Press ACTIONS to return to the control interface.
4. Set parameters for file transfer.
- a. From the Control menu choose File Transfer.
 - b. In the File Transfer dialog, highlight the session you want to use. (The session must be Available.)
 - c. Choose < Host Parameters >.
 - d. Specify the type of host environment you are communicating with (TSO, VNUCMS, or CICS) and set the appropriate parameters. (See the section Host File Transfer Parameters earlier in this chapter for definitions of the parameters.)
 - e. Choose < OK > to return to the File Transfer dialog.
5. Begin transferring files.
- a. Choose< Start Transfer>. The Transfer File dialog for your host is displayed.
 - b. Choose the appropriate radio button: Send (to send a file to the host) or Receive (to receive a file from the host).
 - c. Enter file names and any other parameters specific to the host type.
 - d. Choose < OK > to start the transfer.
 - e. Choose < OK > to acknowledge the start of file transfer.

If you want to stop the file transfer, choose< Abort Transfer>, select the

Transferring Files

Abort File Transfer radio button to confirm terminating the file transfer process, then choose < OK >.

- f. To return to the main screen when you finish transferring files, from the File Transfer dialog choose < Done > or press **ESC**. From a 3270 session, press ACTIONS to return to the main menu.

5. Customizing 3270 Emulation

This chapter explains how to customize 3270 emulation for LINKix 3270. It contains the following information:

- ◆ Descriptions of the menus used for customization
- ◆ Step-by-step instructions for opening, creating and saving style files
- ◆ Step-by-step instructions for customizing 3270 display sessions
- ◆ Step-by-step instructions for customizing 3270 printer sessions
- ◆ Step-by-step instructions for customizing 3270 display colors
- ◆ Step-by-step instructions for customizing 3270 key definitions
- ◆ Step-by-step instructions for customizing 3270 miscellaneous options

The system administrator may have configured LINKix 3270 so that you can only use the default customization settings. In this case, all the menu options described in this chapter (except Key Definitions and Exit) are disabled (grayed) and you cannot use them.

The Style File

A style file contains parameters that control the appearance and behavior of your UNIX terminal in 3270 emulation. These parameters determine session characteristics, display colors, key definitions, audible signals and terminal type. The style file also contains any keystroke sequences you have recorded (see **Keystroke Recording and Replay** in chapter 2).

The name of the open style file is displayed on the main screen of the 3270 emulation program. If the style file that you want to change is not open, use the File menu to open it. Then use one or more of the options on the Customize menu to make changes.

Search Path for Style Files

When you start the 3270 emulation program, you have the option of specifying a style file. LINKix 3270 looks for this file as follows:

- 1 . If you specify a style file, LINKix 3270 tries to load this style file. Style files have an extension of .stu, although the extension should not be specified as it is automatically added. If you specify a fully-qualified or relative directory path, LINKix 3270 looks for a style file in the given directory. If you do not give a path, LINKix 3270 looks for the file in your current directory.

If the file is found, the emulation program is started using this style file. If LINKix 3270 cannot find the file, it uses default settings as in option 3.
2. If you do not specify a style file, LINKix 3270 uses the entry in the configuration file to find the default style file specified for you. It then tries to find this file:
 - (a) In your home directory (specified in the HOME environment variable)
 - (b) If the file does not exist in your home directory, or if the HOME environment variable is not set, LINKix 3270 looks in the */usr/lib/linkix* directory.
- 3 . If LINKix 3270 cannot find the style file you specify or the default style file, it implements built-in default settings. An error message is displayed when the 3270 emulation program is started, to warn you that the style file specified could not be found; choose < Exit Message > to acknowledge this message.

Main Screen Menus

After starting the 3270 emulation program as described in chapter 2, you can access the main screen at any time. If you are logged on to a 3270 session, press ACTIONS to display the main screen.

The main screen provides the following menus:

- ◆ File
- ◆ Customize
- ◆ Control

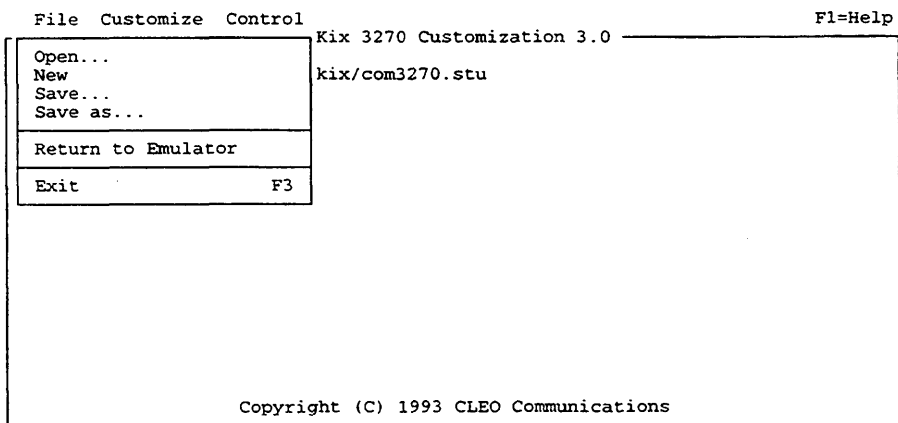
The File and Customize menus are used for customizing 3270 emulation. (The Control menu is used to control 3270 emulation ò see chapter 3 for more details.)

Customizing 3270 Emulation

File Menu

The File menu contains options that allow you to:

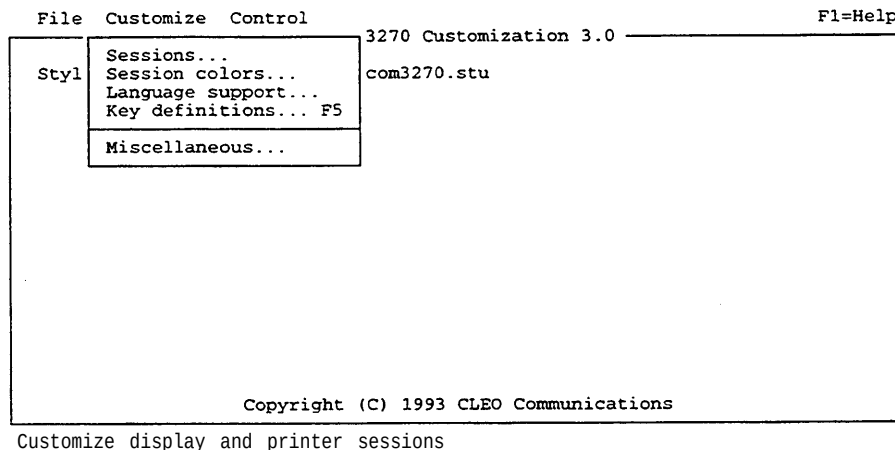
- ◆ open a new or existing style file,
- ◆ save anew or existing style file,
- ◆ move to a 3270 emulation session or
- ◆ exit from the program.



Open a style file

Customize Menu

The Customize menu contains options that allow you to change your 3270 emulation features. These include session characteristics, color mapping, the host language (if your UNIX computer includes ISO 8859 international language support), keyboard mapping, use of an audible signal and choice of mono or color display for 3270 emulation (if your UNIX computer and your terminal support color). The Language support option is disabled (grayed) if you started the 3270 emulation program without using the -i option. For any customization to become permanent, you must save it in a style file.



Customize display and printer sessions

Most customization changes take effect right away. However, two changes take effect only after you exit and restart the 3270 emulation program:

- ◆ designating a session to be active at initialization
- ◆ assigning a session ID

It's also important to note that changes to one parameter may affect others. To avoid problems, set the options in order as follows:

- ◆ If your UNIX computer/terminal supports color and you want to use it, set the "Use color" option in the Miscellaneous dialog box before customizing colors.
- ◆ Set the host language you are using before customizing key mappings.

Opening, Creating and Saving Style Files

From the main screen, you can use the File menu to open, create and save style files.

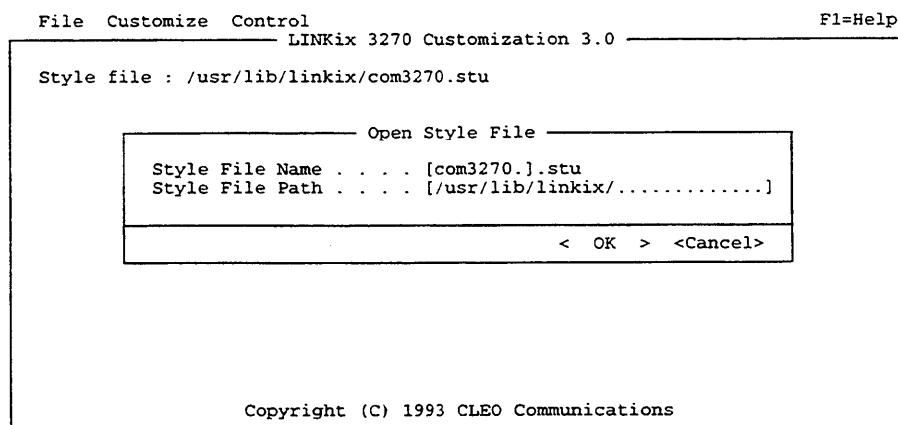
Session IDs specified in a style file only take effect if the style file is opened when the 3270 emulation program is started (either by default or by specifying it on the command line). If you open a style file from the File menu, the session IDs in it will not take effect immediately; you must stop the 3270 emulation program and restart it using this style file to use the session IDs. (See Customizing Display Sessions later in this chapter for more information on session IDs.)

Opening a Style File

You can open any style file and apply its customization parameters to 3270 emulation. If you do not choose a style file, LINKix 3270 implements its default settings.

Follow these steps to open a:

1. Disable any enabled or active display or printer sessions (you may open a style file only when no sessions are enabled). See chapter 3, **Controlling 3270 Emulation**, for instructions on disabling sessions.
2. Save any changes to the current style file, if you have one open and want to keep the changes. See Saving a Style File, later in this chapter, for instructions on saving customization changes to a style file.
3. From the File menu, choose Open. The dialogOpen Style File dialog is displayed.



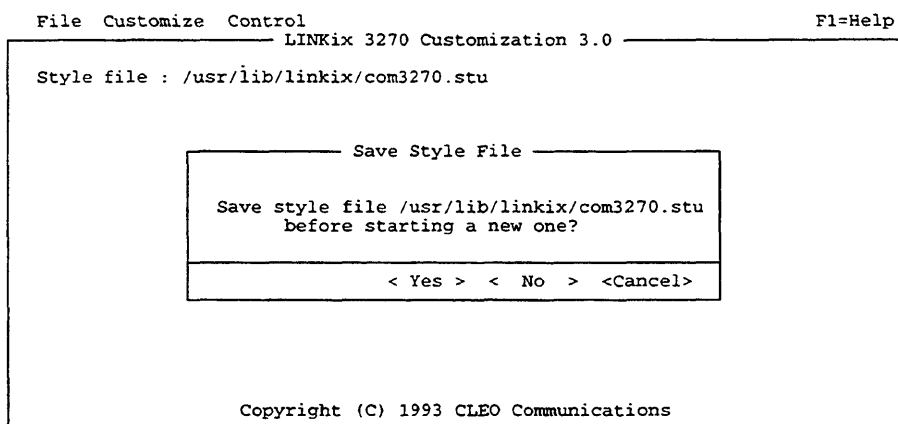
4. In the Open Style File dialog, enter the name of the style file you want to open. Do not include the .stu extension. Enter the connect directory path, if necessary.

If LINKix 3270 cannot find the style file you specify, an error message is displayed. Choose < Exit Message > to acknowledge the message, then verify the name of your style file and re-enter it.

5. Choose < OK > to open the specified file.

If you change the current style file and attempt to open another file before you save the changes, the dialogSave Style File dialog is displayed.

Customizing 3270 Emulation



You have three options:

- ◆ <Yes > saves the changes and opens the new file.
- ◆ < No > opens the new file without saving the changes.
- ◆ <Cancel > continues using the current style file.

Note that the style file in the configuration file contains the name of the default style file and that it is searched for first in your home directory and then in */usr/lib/linkix/*. This means that you can load the default style file created by your system administrator (which will be in */usr/lib/linkix/*), modify this style to create your own version, and then save this version with the same file name in your own home directory, without overwriting the original default file. When you start the 3270 emulation program again, the style file used as the default will be your own version rather than the system default; this saves having to specify the name every time you start the 3270 emulation program. You can, of course, override this default by specifying another file name when you start the program.

Creating a New Style File

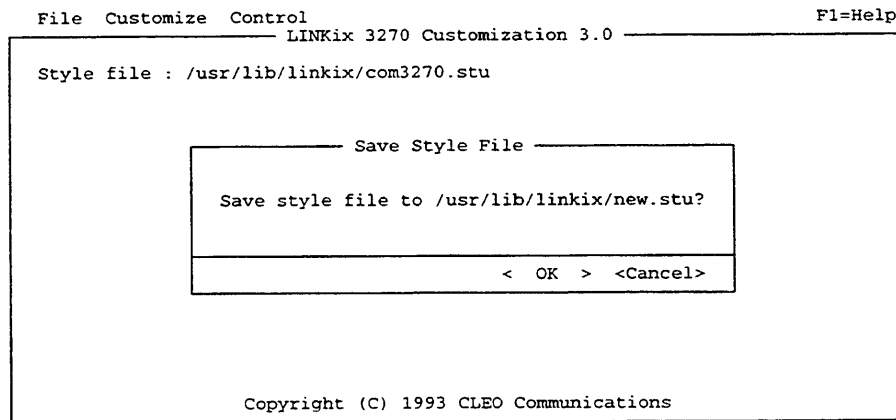
To create a:

1. Choose New from the File menu.
2. Use the Customize menu to customize your 3270 emulation. See **Setting Customization Options**, for instructions on customizing 3270 emulation options.
3. Follow the directions under Saving a Style File to save the style file.

Saving a Style File

To save any changes you make to the current:

1. Choose Save from the File menu. The Save Style File message box is displayed.



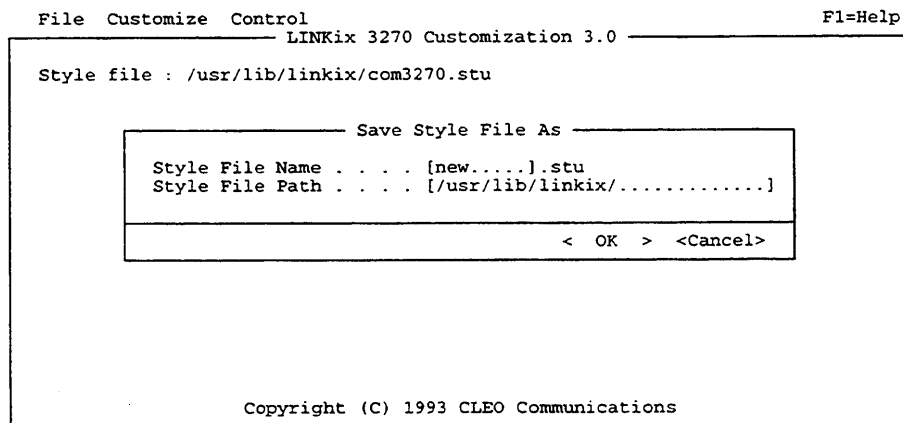
2. Choose < OK > to save the changes or < Cancel > to revoke the command.

If you save changes to a new style file, the Save Style File As dialog is displayed. See the following section, Saving a Style File Under a Different Name.

Saving a Style File Under a Different Name

To save the current style file saving under a different name or to save a new style.

1. Choose Save as from the File menu (if you're saving a new style you can choose Save or Save as). The dialog Save Style File As dialog is displayed.



2. Type a style filename. LINKix 3270 adds the style file extension.stu.
3. If you are saving the file to a directory other than the current directory, type the name of that directory.
4. Choose < OK > to save the file, < Cancel > to revoke the command.

Customizing Display Sessions

You can customize display sessions to do the following:

- ◆ Specify the long name, session ID and description of a session
- ◆ Specify whether a session is enabled or disabled when the 3270 emulation program is started
- ◆ Specify whether the 3270 status line is written to the terminal's status line (if available) or to the last line of the screen
- ◆ Specify the display model for a display session
- ◆ Specify the maximum structured field size used for file transfers on this session
- ◆ Specify a local copy printer or disk file

To customize one or more display sessions, choose Sessions from the Customize menu. The dialogsChoose Session to Customize dialog is displayed.

File Customize Control Fl=Help

Choose Session to Customize

Styl	Session Number	Long Name	Sess ID	LU/Pool Name	Session Type	Curr Sess
	1	SESS01	A	CRTL	(Display)	*** ^
	2	SESS02		CRTL001	(Display)	
	3	SESS03		CRTL002	(Display)	#
	4	SESS04		CRTL003	(Display)	#
	5	SESS05		PRTL001	(Printer)	#
						#
						v
						]

< Zoom > < Done >

Copyright (C) 1993 CLEO Communications

This dialog lists both display sessions and printer sessions. The type of session is displayed in the Session Type column.

Customizing 3270 Emulation

To customize one or more display sessions:

1. Highlight a display session from the list and choose < Zoom >. The dialogCustomize Display Parameters dialog is displayed.

File Customize Control F1=Help

Choose Session to Customize

Customize Display Parameters for Session 1

Styl

Description . . [.....]

Long name . . [SESS01..]

Session ID . . [A]

Model Type:

(o) 2 (24x80)

() 3 (32x80)

() 4 (43x80)

() 5 (27x132)

[] Host screens suppressed at startup

[] Session enabled at initialization

[] 3270 uses terminal status line

Local copy done to:

(o) Real printer

() Print to File

Maximum structured field size (kB) [4.]

[] Append (not overwrite) file

Printer or file name . . [.....]

< OK > <Cancel>

Make whatever changes are necessary to the following fields.

Description

This field contains an optional description of up to 25 characters. This description appears in session list boxes and helps you to identify sessions.

Long name

This field contains a name of up to 8 characters that identifies this session. It must be unique (not match an existing session long name or any other session ID). Lower case characters are converted to uppercase automatically.

Session ID

This field contains a one-character name for the session (A-Z). This letter must not be used by any other session ID or a longname. This field is required if the session is to be used for running HLLAPI applications or command-line file transfer; otherwise it's optional. Lower case is converted to uppercase automatically.

Host screens suppressed at startup

This check box specifies whether or not the host screens for this session will be suppressed when the session is initially started. This option should only be used in conjunction with HLLAPI applications that need to write to the screen in place of, or in addition to, the host screens (see the explanations of DISPLAY ATTRIBUTE, DISPLAY OFF, DISPLAY ON and DISPLAY TEXT in the *HLLAPI Programmer's Guide* for details). When this option is checked, all host screen information sent to this session is suppressed. Use the HLLAPI function DISPLAY ON to display the current host screen (and all subsequent host screens).

If you mark this check box, you must also specify a session ID for the session (see above).

Session enabled at initialization

This check box specifies whether or not the session is enabled each time you start the 3270 emulation program. When running the 3270 emulation program in the background, all sessions that need to be activated must have this parameter checked and must have a session ID assigned (a session can only be activated when running in the background if it has a session ID assigned).

3270 uses terminal status line

This check box indicates whether or not the terminal you are using has the ability to allow programs to write to its status line. If this field is checked, the 3270 emulation program will write its status line to the terminal's status line. Leave this field unchecked if the terminal does not provide a status line or if you do not want to use this feature.

This option is only examined when a session is started; if you change this field while the session is enabled, the change does not take effect until you stop and restart the session.

If your terminal does not support writing to a status line, LINKix ignores this option. See your system administrator for more information on whether or not your terminal supports this option.

Model Type

This radio group specifies the screen configuration for your monitor. The numbers within the parentheses represent the number of rows and columns, respectively. If this radio group is disabled (grayed), your system administrator has not given you permission to change the screen model for this session. Also, the session must be disabled before model type can be changed.

Customizing 3270 Emulation

Maximum structured field size

The size of the data buffer to be used for file transfers on this session. In general, a larger size will speed up the file transfer process. However, some hosts will not accept larger sizes; check this value with your host personnel or system administrator. This value is required. Specify a value (in kB) between 1 and 9. The default is 2. This field is disabled (grayed) if a file transfer is in progress on this session.

Local copydone to

This radio group specifies whether local copies are sent to a Real printer or to a file (Print to file).

Append (not overwrite) file

If you selected Print to file for the Local copy option, this check box specifies whether or not the output is added to the end of the file or if an existing file is overwritten.

Printer or file name

The name of the printer (for example lp) or file (up to 64 characters including the directory path) to which print output is to be directed.

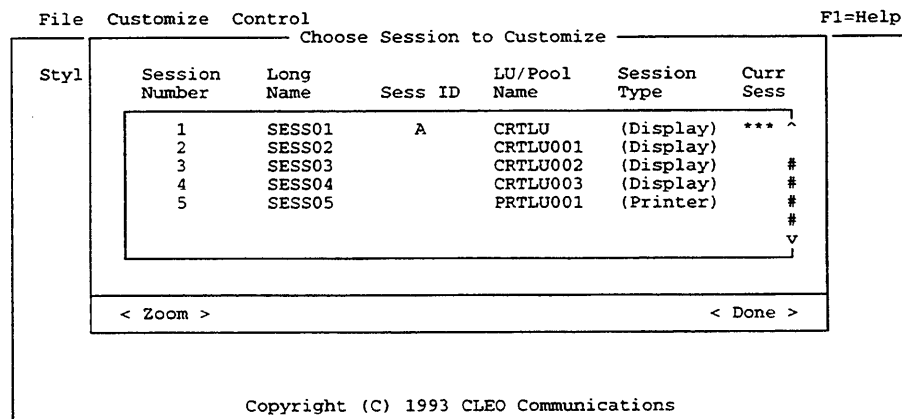
- 3 . Choose < OK > to confirm the changes or < Cancel > to discard them. You return to the Choose Session to Customize dialog.
- 4 . Repeat this process for each display session.

Customizing Printer Sessions

You can customize printer sessions to:

- ◆ Specify whether a session is enabled or disabled when LINKix 3270 is started.
- ◆ Specify the long name or description of a session.
- ◆ Specify a local destination for printer session output.

To customize printer sessions, from the Customize menu choose Sessions. The dialogChoose Session to Customize dialog is displayed.



This dialog lists both display and printer sessions. The type of session is displayed in the Session Type field.

Customizing 3270 Emulation

To customize printer sessions:

1. Use the cursor movement keys to highlight a printer session and choose < Zoom>. The dialogCustomize Printer Parameters dialog is displayed.

File Customize Control F1=Help

Choose Session to Customize

Customize Printer Parameters for Session 3

Styl

Description . {.....}

Long name . . [SESS03..]

Session ID . . [.]

Model Type:

() 2 (24x80)

() 3 (32x80)

() 4 (43x80)

() 5 (27x132)

[] Host screens suppressed at startup

[] Session enabled at initialization

[] 3270 uses terminal status line

Local copy done to:

(o) Real printer

() Print to File

Maximum structured field size (kB) [.]

[] Append (not overwrite) file

Printer or file name . . {.....}

< OK > <Cancel>

2. Make whatever changes are necessary to the following fields.

Description

This field contains an optional description of up to 25 characters. This description appears in session list boxes and helps you to identify sessions.

Long name

This field contains a name of up to 8 characters that identifies this session. It must be unique (not match an existing session long name or any other session ID. Lower case characters are converted to uppercase automatically.

Session ID

This field is disabled (grayed), since it does not apply to printer sessions.

Host screens suppressed at startup

This field is disabled (grayed), since it does not apply to printer sessions.

Session enabled at initialization

This check box specifies whether or not the session is enabled each time you start the 3270 emulation program.

3270 uses terminal status line

This check box is disabled (grayed), since it does not apply to printer sessions.

Model Type

This radio group is disabled (grayed), since it does not apply to printer sessions.

Maximum structured field size

This field is disabled (grayed), since it does not apply to printer sessions.

to

This radio group specifies whether local copies are sent to a Real printer or to a file (Print to file).

Append (not overwrite) file

If you selected Print to file for the Local copy option, this check box specifies whether or not the output is added to the end of the file or if an existing file is overwritten.

Printer or file name

The name of the printer (for example lp) or file (up to 64 characters including the directory path) to which print output is to be directed.

3. Choose < OK > to confirm the changes or < Cancel > to discard them.
4. Repeat this procedure for each printer session that you plan to use-

Customizing Display Colors

You can decide how your terminal shows the colors and highlighting attributes that the host application assigns to the 3270 display's contents, including the status line. You can even use different color combinations, called **color mapping**, for each display session to help differentiate one from another.

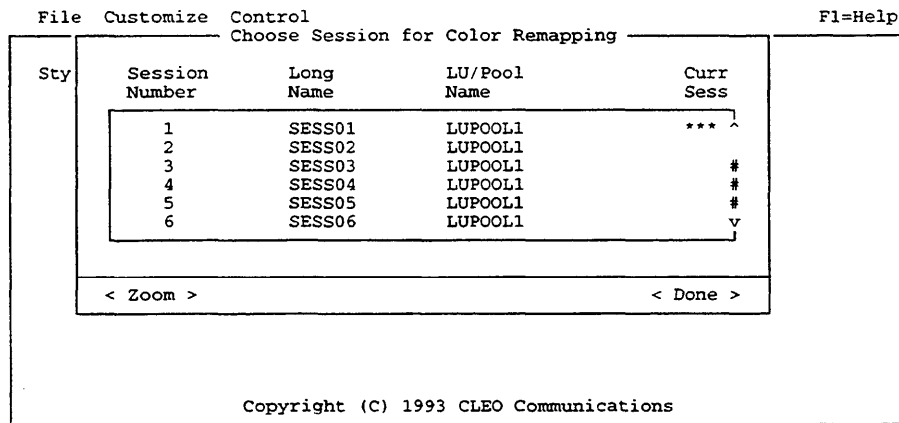
The use of color mappings depends on whether your UNIX computer and your terminal support color. If they do, then you can use the Use color check box in the Miscellaneous Options dialog to specify whether you want to use mono or color display for 3270 emulation. If they do not, you can only use mono display.

When you use a mono display, you can still specify different combinations of reverse video, underline intense and blinking terminal attributes to distinguish between different host colors.

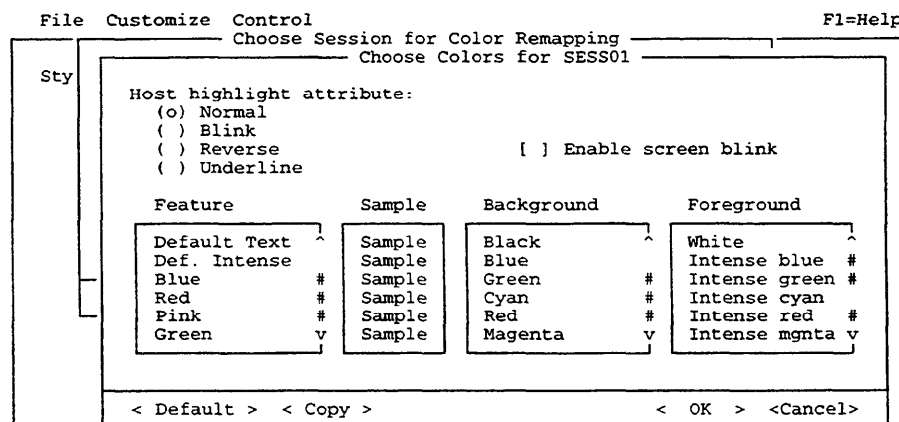
The description of the dialogs used for color mapping, in this section, assumes you have a color monitor. If you have a monochrome monitor, then the procedure for remapping colors is the same, but you will obviously not have the choice of colors, only the choice of terminal attributes.

To customize display colors:

1. From the Customize menu choose Session Colors. The dialogChoose Session for Color Remapping dialog is displayed.



2. Select a specific session and choose < Zoom >. The dialogChoose Colors for session dialog, is displayed.



Customizing 3270 Emulation

In this dialog you assign the color and highlighting characteristics that your terminal uses for each combination of host feature and host highlight attribute.

On the left side of the dialog are the Host Highlight Attribute and the host Feature (a particular color or the status line) to be represented. On the right are the background and foreground colors available to represent the host attributes and features on your terminal. The way your terminal currently represents host attributes and features is shown in the list boxes and in the Sample column.

The information that's displayed consists of the following.

Host highlight attribute

Each attribute (Normal, Blink, Reverse or Underline) corresponds to a highlighting character assigned by the host. Each attribute is represented by a combination of color and highlighting on your terminal.

Feature

These features, which refer to 3270 display contents at the host, include the following: default text, default intense text, four base colors (base blue, base red, base green, and base white), seven sextended colors (blue, red, pink, green, turquoise, yellow, and white), seven intense colors (blue, red, pink, green, turquoise, yellow, and white), and the status line. Each Feature is associated with one of four different Host highlight attributes. All Features, except the status line, can be modified. The status line is always normal and only appears in the Feature list box if Normal is the selected host highlight attribute.

Sample

This field shows the Foreground and Background colors (and the Host highlight attribute) for the host Feature. As you change your terminal's representation of each host Feature/Attribute, this column reflects the change.

Enable screen blink

This check box specifies whether or not the foreground content of the 3270 display will blink for a particular combination of host feature and host highlight attribute.

Background

This box lists all the background colors you can display on your terminal. The color highlighted in this box is the color currently designated to represent the background for the combination of Host Highlight Attribute and host Feature currently selected.

Foreground

This box lists all the foreground colors you can display on your terminal. (If you are using a mono display, the colors listed here depend on the background color selected; for example, if you have selected white background, only black is listed because this is the only valid foreground color.) The highlighted color is the color that's currently designated for the foreground for the combination of Host Highlight Attribute and host Feature selected at left.

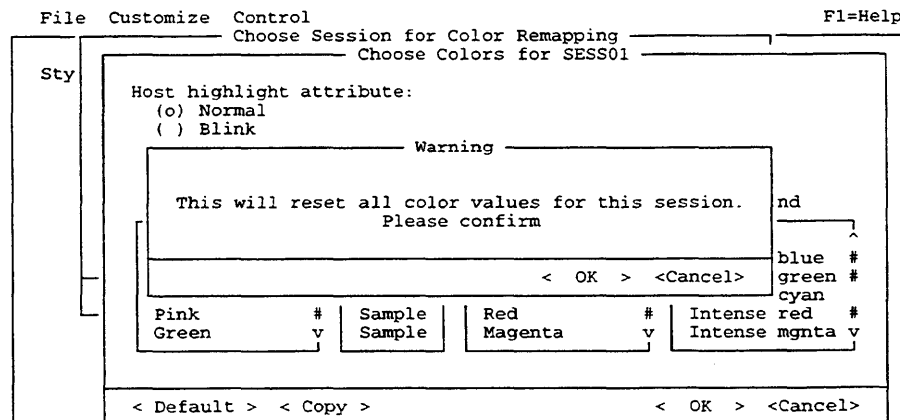
In this dialog you can:

- ◆ Assign the default color mapping to a session
- ◆ Copy colors from one session to another
- ◆ Customize session colors

Use Default Color Mapping

To use the default color mapping that came with LINKix 3270:

1. Choose < Default >. A Warning dialog is displayed.

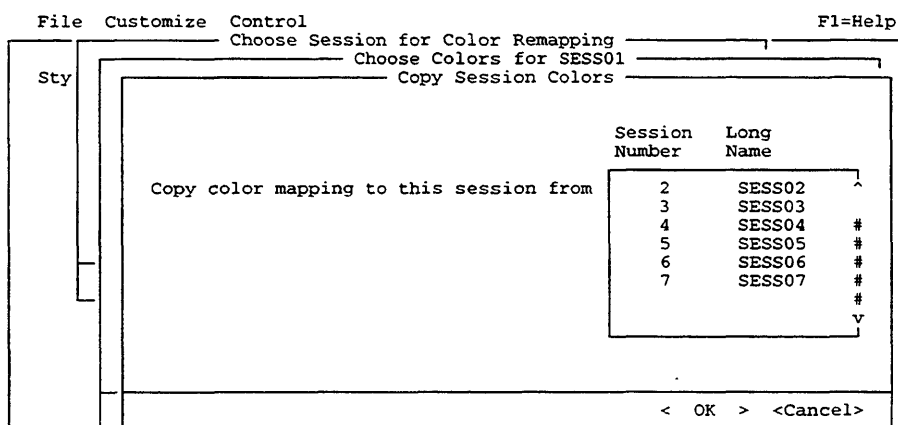


2. Choose < OK > to confirm the change to default colors, or choose < Cancel > to leave the current color and attribute settings in effect.

Copy Colors From Another Session

To copy the color mapping from another session:

1. In the Choose Colors for *Session* dialog, choose < Copy >. The dialogsCopy Session Colors dialog is displayed.



2. Select a session from the list box whose color assignments you want to copy.
3. Choose < OK > or < Cancel >. The Choose Colors for *session* dialog reappears. The color mapping is now the same as for the session you specified in step 2.
4. Make any changes you like to the new color mapping, or leave it as is.
5. Choose < OK > to accept the copied color map, with any subsequent modifications, or choose < Cancel > to return to the color map in effect when you opened the dialog.

Customizing Colors

To customize how your terminal represents 3270 display features and attributes:

1. With the Choose Colors for session dialog displayed, select the appropriate Host highlight attribute radio button.

File Customize Control F1=Help

Choose Session for Color Remapping Choose Colors for SESS01

Host highlight attribute:

☐ Normal ☐ Enable screen blink

☐ Blink

☐ Reverse

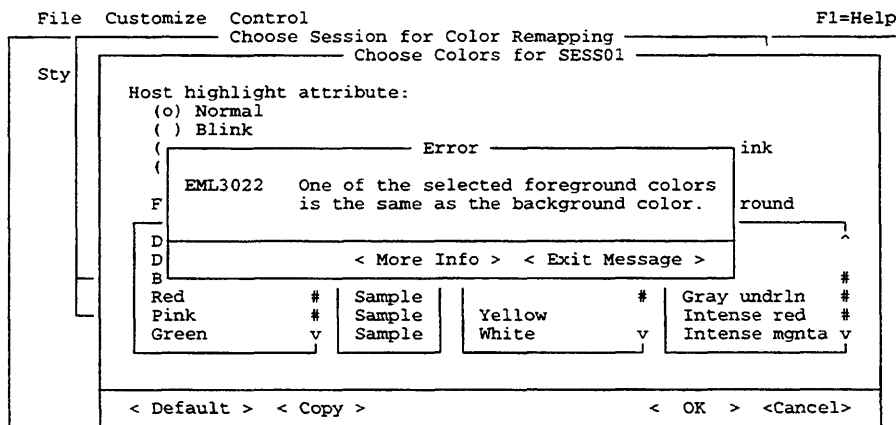
☐ Underline

Feature	Sample	Background	Foreground
Default Text	Sample	Black	White
Def. Intense	Sample	Blue	Intense blue
Blue	Sample	Green	Intense green
Red	Sample	Cyan	Intense cyan
Pink	Sample	Red	Intense red
Green	Sample	Magenta	Intense mgnta

< Default > < Copy > < OK > <Cancel>

2. Choose a host Feature.
3. Choose a Background color and a Foreground color to represent the Feature.
4. Choose whether to use blinking foreground text (Enable screen blink check box).
5. Repeat steps 2 and 4 for each host feature you want to customize.
6. Now repeat steps 1 through 5 for the other Host highlight attributes.
- 7a. To save the customized color settings, choose < OK >. If the foreground and background colors for a feature are the same, an error message is displayed.

Customizing 3270 Emulation



Choose < Exit Message > to acknowledge the error message. The first color mapping that is in error is highlighted so that you can correct it.

- 7b. You can also discard your changes at any time by choosing < Cancel >.
8. You return to the Choose Session for Color Remapping dialog where you can select another session for color remapping, or choose < Done > to return to the main screen.

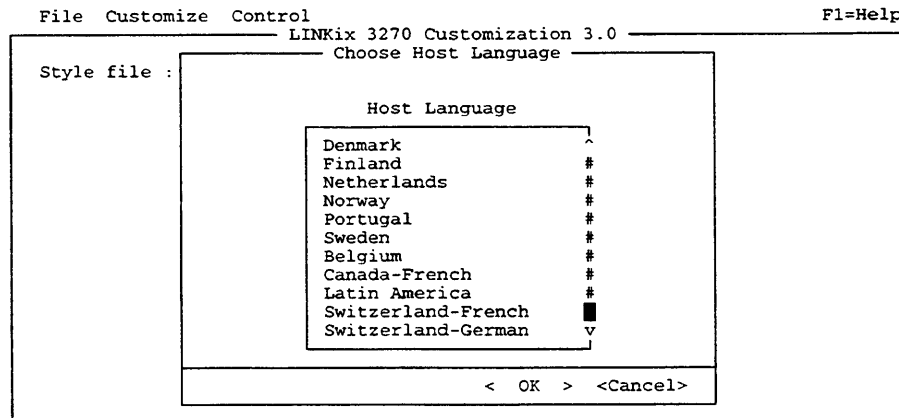
Choosing a Host Language

This option is useful when you are communicating with a host computer in a different country. LINKix 3270 uses the choice you make here to match your local code page with the host code page. Note that the choice of host language affects all your 3270 sessions; you cannot customize host language separately for each session.

This option is only available (on versions of UNIX which include ISO 8859 international language support) if you started the program using the -i option. Otherwise, the Host language menu option is disabled (grayed) and this section does not apply.

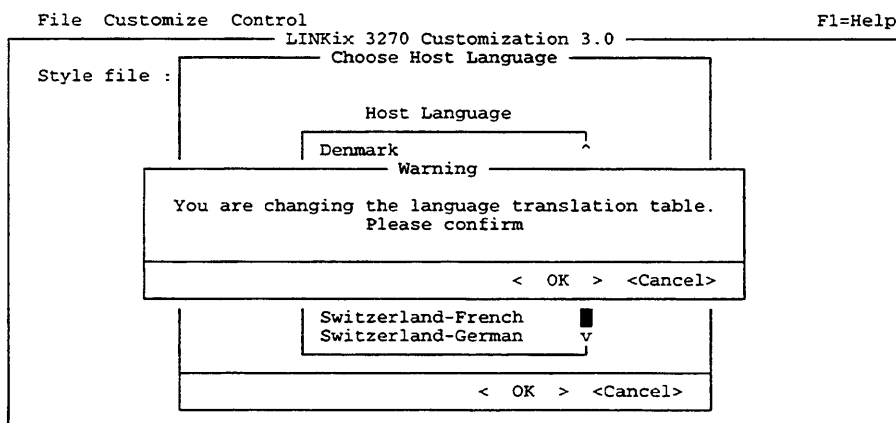
To specify the correct host code page, take the following steps:

1. From the Customize menu, choose Language Support. The dialog Choose Host Language dialog is displayed.



2. Select an available host code page from the Host Language list box.
3. Choose < OK >. If you change code pages, a Warning dialog is displayed that asks you to confirm the change.

Customizing 3270 Emulation



4. Choose < OK > to confirm the change, or choose < Cancel > to cancel the change. You return to the main screen.

The Host Language list box contains the following entries:

United States	Norway
United Kingdom	Portugal
Germany	Sweden
France	Belgium
Italy	Canada-French
Spain	Latin America
Denmark	Switzerland-French
Finland	Switzerland-German
Netherlands	

The host language you select defines the characters that can be sent to the host from your terminal during 3270 emulation sessions. This means that the set of valid 3270 character keys may include some for which there is no corresponding key on your keyboard (for example â or ü). In order to use these keys, you can remap your keyboard as described in the next section, **Customizing Key Definitions**.

If you have a style file which uses a host language other than the default, you need to use the `-i` option when starting the 3270 emulation program if you are going to use this style file. If you use this style file without the `-i` option, the host language is assumed to be U.S. English and all key mappings revert to the 3270 emulation program's default settings.

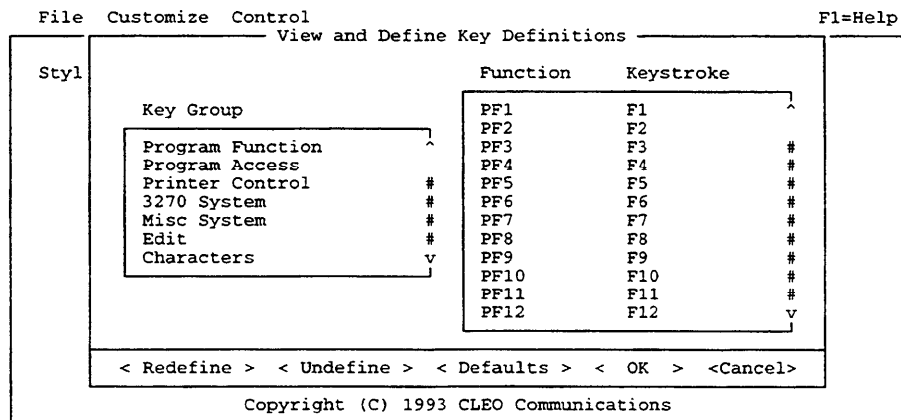
Customizing Key Definitions

LINKix 3270 maps the keys on your terminal's keyboard to functions used in 3270 emulation. You can review those definitions, and change them if you want, by choosing Key Definitions from the Customize menu. If your System Administrator has not given you permission to modify style files, you can view the key mappings as described in this chapter, but you cannot change them.

When you change your keyboard definitions, the change affects all your 3270 sessions- You cannot customize keyboard mapping by session.

The accelerator keydirect accelerator keystrokes shown in the File, Customize and Control menus (which allow you to jump straight to specific dialogs from the main screen) cannot be remapped to different keys. Key remapping applies only to display emulation.

To review current keyboard mapping, start from the Customize menu and choose Key definitions. Or from the main menu or emulation screen, press KEYS. The dialogsView and Define Key Definitions dialog is displayed.



This dialog contains the following items.

Key Group

The 3270 emulation program categorizes 3270 functions into seven groups. These groups are listed in the Key Group box. When a group in the Key Group list box is highlighted, the functions in that group are listed in the Function column on the left side of the list box at right.

Function

The 3270 functions included in the Key Group are listed in this column.

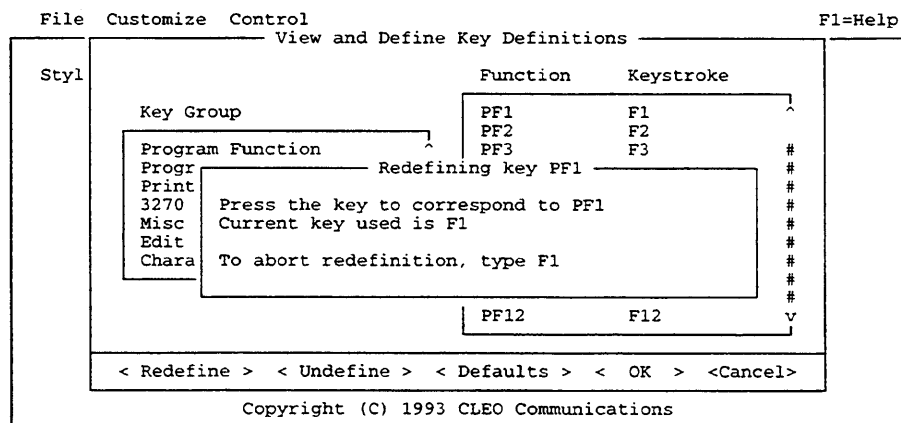
Keystroke

The key or keystroke combination listed in this column is the keystroke on your terminal which performs the associated Function.

Redefining Keys

To redefine the keystrokes:

1. In the Key Group list box, highlight the group that has the function to redefine.
2. In the Function column, highlight the function to be redefined.
3. Choose < Redefine >. The dialog Redefining Key dialog is displayed. This dialog prompts you to press the key combination you want to assign to the function.



Customizing 3270 Emulation

At this point you can not use the **F1** key to invoke the help screens, or the **CTRL + R** keystroke to redraw the screen, since either of these keystrokes will be accepted as valid keystrokes to which a key could be redefined.

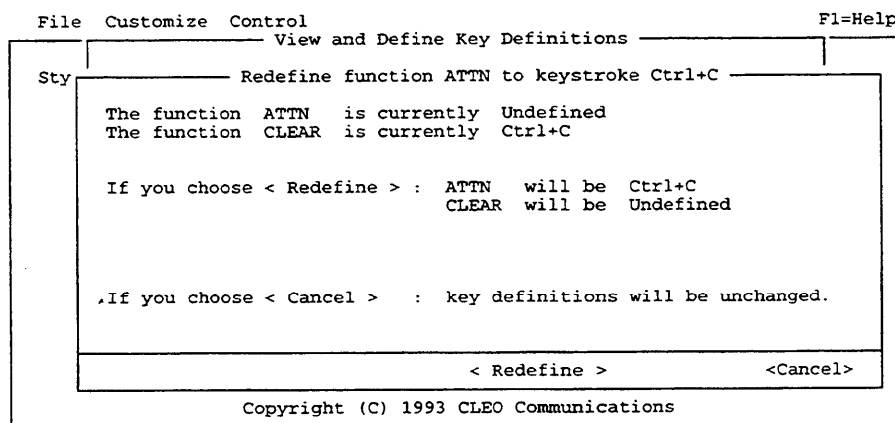
4. Press the key combination on your terminal's keyboard that you want to correspond to the 3270 function. You can use a single key with either **SHIFT** or **CTRL**, or one or two **ESC** keystrokes followed by a single key with either **SHIFT** or **CTRL**. Choose the key combination that's currently mapped to the function if you wish to retain that mapping.

If your terminal uses xon/xoff flow control, you will not be able to use the key combinations **CTRL + Q** and **CTRL + S** for key mappings.

5. Depending on whether or not the key combination is already used and the function is already defined, one of the following occurs:
 - A. If the new key combination is not already used and the function is not defined, the new definition is displayed in the Function/Keystroke list box.
 - B. If the new key combination is not already used but the function is currently defined as a different key combination, the dialog `Redefine function to` dialog is displayed.

Make one of the following choices:

- ◆ Choose < Redefine > to assign the new key combination to the function.
 - ◆ Choose < Cancel > to end the redefinition without making the change.
- C. If the new key combination is already in use and the function is currently undefined, the dialog box shown below is displayed.

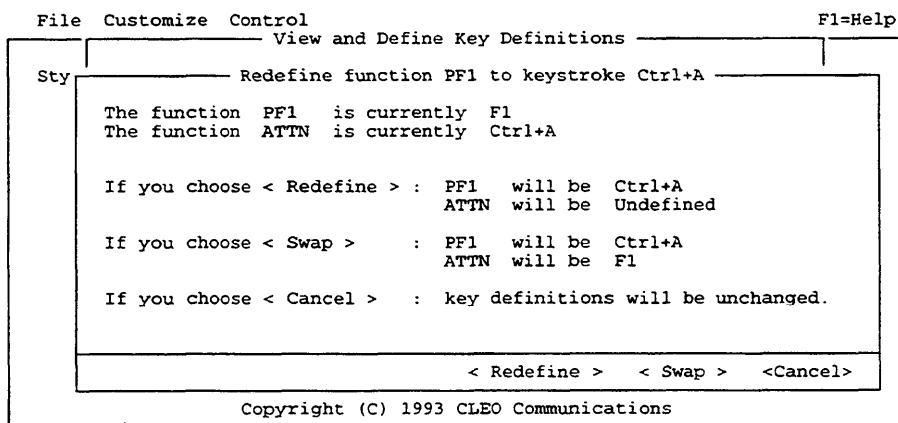


Make one of the following choices:

- ◆ Choose < Redefine > to reassign the new key combination to the function and clear its previous use.
- ◆ Choose < Cancel > to end the redefinition without making the change.

Customizing 3270 Emulation

- D. If the new key combination is already in use and the function is currently defined as a different key combination, the dialog box shown below is displayed.



Make one of the following choices:

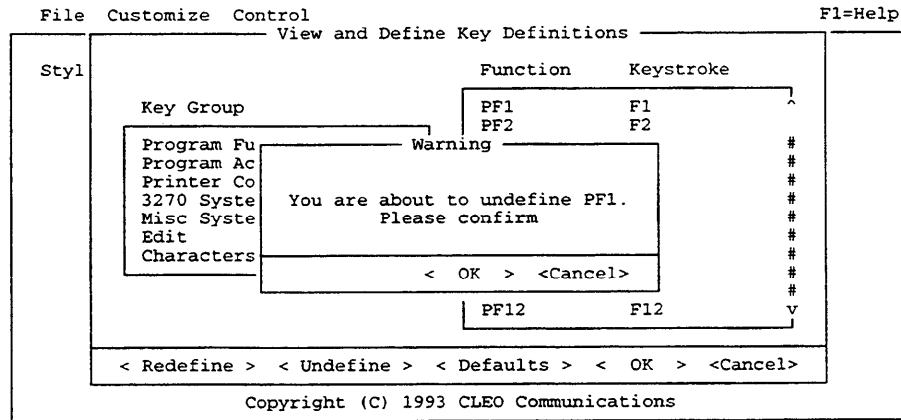
- ◆ Choose < Redefine > to reassign the new key combination to the function and clear its previous use.
- ◆ Choose < Swap > to swap definitions between the two 3270 functions.
- ◆ Choose < Cancel > to end the redefinition without making the change.

6. You return to the dialogsView and Define Key Definitions dialog.

Undefining Keys

To undefine your terminal's representation of a 3270 key, so that no keystroke corresponds to that 3270 key:

1. In the View and Define Key Definitions dialog, select the key you want to undefine from the Function/Keystroke list box.
2. Choose < Undefine >. A warning is displayed.



3. Choose <OK> to undefine the key or < Cancel > to retain the current definition. The View and Define Key Definitions dialog remains. If you undefined the key, "Undefined" appears in the Keystroke column.

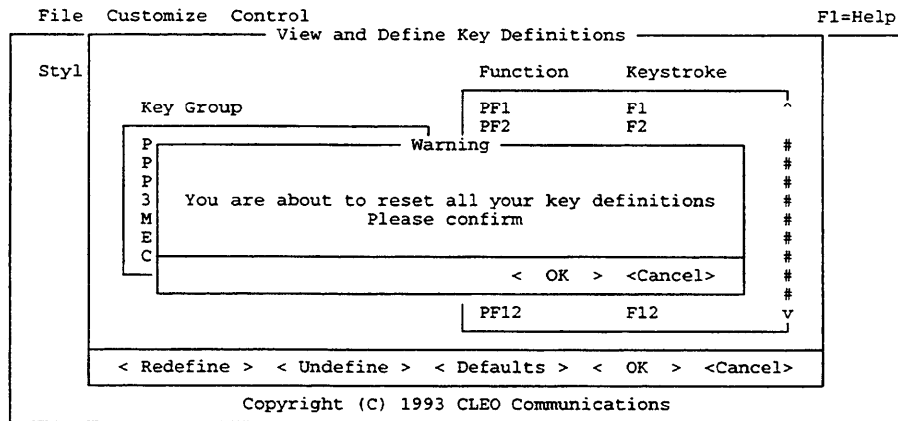
When you finish redefining keys, choose < OK > to confirm your changes or < Cancel > to abandon them. The main menu is displayed.

Choosing Default Keyboard Mapping

To change the keyboard mapping to LINKix 3270's original settings:

1. In the View and Define Key Definitions dialog, choose < Default >. A warning box is displayed.

Customizing 3270 Emulation



2. Choose < OK > to confirm the Default settings or < Cancel > to abandon the change. Either choice causes the View and Define Key Definitions dialog to be displayed.

Customizing Miscellaneous Options

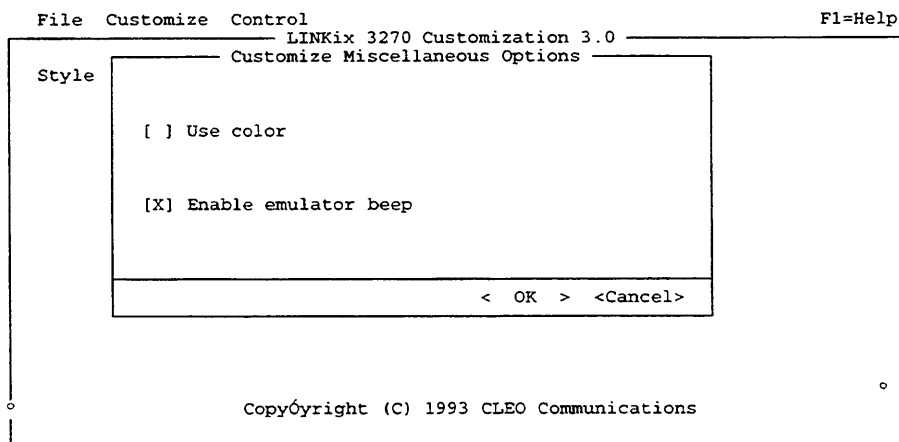
This section allows you to specify whether you want to use mono or color display on your terminal, or whether or not an audible signal (beep) is produced when requested by the host application or to indicate errors during emulation.

The choice of mono or color display applies only if your UNIX computer and your terminal support color if they do not, then you cannot select it. This choice affects only the display of 3270 emulation screens, not LINKix menus and dialogs. If you change the setting, or attempt to use a style file set up for a color terminal on a mono terminal, the colors displayed during a 3270 session will be affected.

The audible signal occurs (for example) when you type too many characters for the keyboard buffer to hold, or when the host sends a signal to the 3270 emulation program to indicate that an error has occurred, or when you use a PF key that already has a key sequence assigned while recording a new sequence. You can specify whether or not you want these audible signals to be produced. (Note that this option does not affect the audible signals which are produced when you make an error in the menus and dialogs of the 3270 control interface ð these cannot be disabled.)

To set any of these options:

1. From the Customize menu choose Miscellaneous. The dialogSpecify Miscellaneous Options dialog is displayed.



Customizing 3270 Emulation

2. Check the Use color check box to use a color display. The check box is disabled (grayed) if your UNIX computer or terminal does not support color.
3. Check the Enable emulator beep check box if you want audible signals to indicate errors.
4. Choose < OK > to confirm any changes or < Cancel > to abandon.
5. If you changed the color option, a message warning that the color mappings for your 3270 display sessions will be reset to the system defaults is displayed. Choose < OK > to confirm the change or < Cancel > to return to the dialog.

Custom Translation Tables

LINKix 3270 supports custom translation tables that can be specified, when starting 3270 emulation, in place of specifying an international host language. The translation table is a text file of the same format as *comtblg.dat* (see the *CSV Programmer's Guide*): 32 lines consisting of 32 hexadecimal digits, representing a total of 512 characters. The first 16 lines (256 characters) specify the ASCII to EBCDIC conversion characters; the other 16 lines specify the EBCDIC to ASCII conversion characters. For example:

```
00010203372D2E2F1605250B0C0D0E0F
101112133C3D322618193F27221D351F
405A7F7B5B6C507D4D5D5C4E6B604B61
F0F1F2F3F4F5F6F7F8F97A5E4C7E6E6F
7CC1C2C3C4C5C6C7C8C9D1D2D3D4D5D6
D7D8D9E2E3E4E5E6E7E8E9ADE0BD5F6D
79818283848586878889919293949596
979899A2A3A4A5A6A7A8A9C06AD0A107
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
000102030009007F0000000B0C0D0E0F
101112130000080018190000001D001F
00001C00000A171B0000000000050607
00001600001E0004000000001415001A
20000000000000000000000002E3C282B00
260000000000000000000000021242A293B5E
2D2F00000000000000000007C2C255F3E3F
0000000000000000000000603A2340273D22
0061626364656667686900000000000000
006A6B6C6D6E6F70717200000000000000
007E737475767778797A00000005B00000
000000000000000000000000000005D00000
7B4142434445464748490000000000000000
7D4A4B4C4D4E4F5051520000000000000000
5C00535455565758595A0000000000000000
303132333435363738390000000000000000
```

Specifying a custom translation table when starting 3270 emulation disables the Host Language dialog, thus disabling the ability to change the host language set.

The internal EBCDIC table is constructed from the ASCII to EBCDIC code page. Only those target characters which are non-zero are valid. LINKix assumes that the local code page represented by the user defined conversion tables can be displayed by the local operating system. **If invalid values are included in the translation tables, the keyboard remapping dialog may be corrupted.**

Customizing 3270 Emulation

File transfer operations will use these two tables as default; however, if either of the code page fields (EBCDIC or ASCII) is filled in, the other must be selected.

- ◆ In the File transfer dialog, if you select the EBCDIC <-> ASCII conversion option, the word USER appears as the default in both the Host code page and Local code page edit boxes. If only one code page is set to USER, an error message is displayed.
- ◆ For command line file transfer, you must enter both H and P options (specifying Host code page and local code page respectively) or neither. Specifying one without the other gives an error message.

Custom Translation Table Utilities

Because the native format of the custom translation tables can be difficult to edit, we've included the following two utilities.

comtrdmp

This utility uses the common service verb GET_CP_CONVERT_TABLE to convert the LINKix translation table into a more readable format:

```
# ASCII -> EBCDIC Translate Table (256 bytes) :
# -----
0x00 0x01 0x02 0x03 0x37 0x2d 0x2e 0x2f
0x16 0x05 0x25 0x0b 0x0c 0x0d 0x0e 0x0f
0x10 0x11 0x12 0x13 0xb6 0xb5 0x32 0x26
0x18 0x19 0x3f 0x27 0x1c 0x1d 0x1e 0x1f

0x40 0x5a 0x7f 0x7b 0x5b 0x6c 0x50 0x7d
0x4d 0x5d 0x5c 0x4e 0x6b 0x60 0x4b 0x61
0xf0 0xf1 0xf2 0xf3 0xf4 0xf5 0xf6 0xf7
0xf8 0xf9 0x7a 0x5e 0x4c 0x7e 0x6e 0x6f

0x7c 0xc1 0xc2 0xc3 0xc4 0xc5 0xc6 0xc7
0xc8 0xc9 0xd1 0xd2 0xd3 0xd4 0xd5 0xd6
0xd7 0xd8 0xd9 0xe2 0xe3 0xe4 0xe5 0xe6
0xe7 0xe8 0xe9 0xba 0xe0 0xbb 0xb0 0x6d

0x79 0x81 0x82 0x83 0x84 0x85 0x86 0x87
0x88 0x89 0x91 0x92 0x93 0x94 0x95 0x96
0x97 0x98 0x99 0xa2 0xa3 0xa4 0xa5 0xa6
0xa7 0xa8 0xa9 0xc0 0x4f 0xd0 0xa1 0x07

0x68 0xdc 0x51 0x42 0x43 0x44 0x47 0x48
0x52 0x53 0x54 0x57 0x56 0x58 0x63 0x67
0x71 0x9c 0x9e 0xcb 0xcc 0xcd 0xdb 0xdd
0xdf 0xec 0xfc 0x4a 0xb1 0xb2 0x00 0x00

0x45 0x55 0xce 0xde 0x49 0x69 0x9a 0x9b
0xab 0x00 0x5f 0xb8 0xb7 0xaa 0x8a 0x8b
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
```

```

0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

0x00 0x59 0x00 0x00 0x00 0x00 0xa0 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x8f 0x00 0x00 0x00 0x00 0xe1 0x00
0x90 0x00 0xb3 0x00 0x00 0xea 0x00 0x41

# EBCDIC -> ASCII Translate Table (256 bytes):
# -----
0x00 0x01 0x02 0x03 0x00 0x09 0x00 0x7f
0x00 0x00 0x00 0x0b 0x0c 0x0d 0x0e 0x0f
0x10 0x11 0x12 0x13 0x00 0x00 0x08 0x00
0x18 0x19 0x00 0x00 0x1c 0x1d 0x1e 0x1f

0x00 0x00 0x00 0x00 0x00 0x0a 0x17 0x1b
0x00 0x00 0x00 0x00 0x00 0x05 0x06 0x07
0x00 0x00 0x16 0x00 0x00 0x00 0x00 0x04
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x1a

0x20 0xff 0x83 0x84 0x85 0xa0 0x00 0x86
0x87 0xa4 0x9b 0x2e 0x3c 0x28 0x2b 0x7c
0x26 0x82 0x88 0x89 0x8a 0xa1 0x8c 0x8b
0x8d 0xe1 0x21 0x24 0x2a 0x29 0x3b 0xaa

0x2d 0x2f 0x00 0x8e 0x00 0x00 0x00 0x8f
0x80 0xa5 0x00 0x2c 0x25 0x5f 0x3e 0x3f
0x00 0x90 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x60 0x3a 0x23 0x40 0x27 0x3d 0x22

0x00 0x61 0x62 0x63 0x64 0x65 0x66 0x67
0x68 0x69 0xae 0xaf 0x00 0x00 0x00 0xf1
0xf8 0x6a 0x6b 0x6c 0x6d 0x6e 0x6f 0x70
0x71 0x72 0xa6 0xa7 0x91 0x00 0x92 0x00

0xe6 0x7e 0x73 0x74 0x75 0x76 0x77 0x78
0x79 0x7a 0xad 0xa8 0x00 0x00 0x00 0x00
0x5e 0x9c 0x9d 0xfa 0x00 0x15 0x14 0xac
0xab 0x00 0x5b 0x5d 0x00 0x00 0x00 0x00

0x7b 0x41 0x42 0x43 0x44 0x45 0x46 0x47
0x48 0x49 0x00 0x93 0x94 0x95 0xa2 0x00
0x7d 0x4a 0x4b 0x4c 0x4d 0x4e 0x4f 0x50
0x51 0x52 0x00 0x96 0x81 0x97 0xa3 0x98

0x5c 0xf6 0x53 0x54 0x55 0x56 0x57 0x58
0x59 0x5a 0xfd 0x00 0x99 0x00 0x00 0x00
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37
0x38 0x39 0x00 0x00 0x9a 0x00 0x00 0x00

# End-of-Translate-Tables

```

This file can more easily be deciphered and can be altered as necessary. After altering this file, use the translation table preprocessor (*comtrcnv*) to convert the file back into LINKix usable format.

Customizing 3270 Emulation

Syntax

The syntax for this utility is as follows:

```
comtrdmp AsciiCodePage EbcdicCodePage OutputFileName
```

AsciiCodePage 3 digit numeric code (see the *CSV Programmer's Guide* for valid codes)

EbcdicCodePage 3 digit numeric code (see the *CSV Programmer's Guide* for valid codes)

OutputFileName File that contains the translation tables in a more readable format

Return Codes

This utility can return one of the following codes on exit:

1	Invocation error
2	CSV error
3	Unable to open output file

comtrcnv

This utility reads an ASCII text translate table created by *comtrdmp* (or of the same format) and outputs a LINKix translate table file of 32 lines. Each line consists of 32 hexadecimal digits, representing 16 characters. The first 16 lines specify the ASCII to EBCDIC table, and the remaining 16 lines specify the EBCDIC to ASCII table. This output file can be used by LINKix in place of a standard host language table.

Syntax

The syntax for this utility is as follows:

```
comtrcnv [ -q] SourceFileName OutputFileName
```

-q Quite mode; suppress display of program identification at start

SourceFileName Text file of hex codes and comments (in *comtrdmp* format)

OutputFileName Translate table in LINKix format

Return Codes

This utility can return one of the following codes on exit:

1	Wrong number of arguments supplied on invocation
2	Invalid arrangement of arguments
3	Unable to open input file
4	More than 512 Translate Table entries supplied
5	Invalid Table Entry
6	Invalid character encountered
7	Not enough table entries supplied
8	Unable to create output file

6. Solving Problems

This chapter provides information to help you solve encountered while using the 3270 emulation program. Specifically, it includes the following:

- ◆ Alist of problems you might encounter
- ◆ Suggested solutions for each problem

If you encounter a problem that persists or is not listed, contact your system administrator for assistance.

When you have a question about a menu or dialog, display it and press **F1** for on-line help.

Common Problems and Solutions

What follows are the most common problems you may encounter when using the 3270 emulation program and suggested solutions to resolve these problems. If you find you need more help, ask your system administrator.

3270 Emulation Program not Found

If you receive a message that the 3270 emulation program is not found:

The program is not installed on your system

Your PATH environment variable does not include the directory that contains the program (*/usr/bin*)

3270 Emulation Program does not start, with error message:
"3270 could not initialize due to failure of file locking operation"
"3270 could not initialize due to failure to fork a process"
"3270 could not initialize due to failure to open file"
"3270 could not initialize due to failure to open pipe"
"3270 could not initialize due to failure to read from pipe"

If the 3270 emulation program does not start and you receive one of the above messages, an operating system command failed during initialization of the program. This is generally because too many other processes are running at the same time. Retry when other processes have finished or when there are not so many users on the system. If this problem occurs repeatedly, contact your system administrator.

3270 Emulation Program does not start, with error message:
"3270 failed to initialize"
"Could not access daemon's shared memory"

If the 3270 emulation program does not start and one of the error messages above is displayed, the control daemon probably has not been started. See your UNIX system administrator.

3270 Emulation Program does not start, with error message:
"Unable to obtain system semaphores"

If the 3270 emulation program does not start and this error message is displayed, the UNIX computer does not have enough interprocess communication (IPC) queues

configured. The 3270 emulation program cannot be started at present. (You may be able to start it later when there are fewer users on the system.)

The message is followed by other messages which tell you to clear this condition using *ipcs* and/or *ipcrm*. If you do not know how to do this, see your system administrator.

3270 Emulation Program does not start, with error message:
"Cannot access cfg file"

If the 3270 emulation program does not start and this error message is displayed, the LINKix 3270 configuration file does not have proper access permissions. See your UNIX system administrator.

3270 Emulation Program does not start, with error message:
"Cannot run 3270 emulation in background"

If the 3270 emulation program does not start and this message is displayed, the 3270 style file you're using (either one you specified on the command line or the default style file specified in the confirmation file) does not include any sessions which can be used in the background. For a session to be activated when running in the background, it must be configured to be enabled at initialization; in addition, if it's a display session, it must have a session ID assigned.

Start the 3270 emulation program without the *-b* option to run it in the foreground, load the style file, assign at least one session to be active at initialization (and assign a session ID if it is a display session), and save the new style. You can then restart the program in the background using the new style file.

For more information on the enabled at initialization option and session IDs, see Customizing Display Sessions and Customizing Printer Sessions in chapter 5.

3270 Emulation Program does not start, with error message:
"Could not access daemon's shared memory"

If the 3270 emulation program does not start and this error message is displayed, the control daemon has not been started. See your UNIX system administrator.

Cannot Get From 3270 Session to Main Screen

If pressing the key combination for the ACTIONS key does not move you from a 3270 session to the main screen, it may be because you started emulation using the

Solving Problems

*Æ*e option. If you need to use the menu interface, stop the emulation program and restart it without specifying -e.

This problem may also be because the ACTIONS function is undefined or remapped to another key. To correct this problem, take the following actions:

- 1 . Press KEYS to display the View and Define Key Definitions dialog.
- 2 . From the Key Group list box, choose Mist System.
- 3 . Note the new key or key combination mapped to the ACTIONS function.
- 4 . Either press **ESC** to return to the 3270 emulation session and then use the key or key combination now mapped to ACTIONS to exit emulation, or follow the instructions under **Customizing Key Definitions** in chapter 5 to remap the ACTIONS function.

Cannot Open Style File

You cannot open an existing style file or create a new one while a session is enabled or active. If you pull down the File menu and see that the Open option is grayed, use the Control Display Sessions dialog to disable any enabled or active sessions. Then return to the File menu and open or create a style file.

Cannot run 3270 emulation without a menu interface

If you get this error message when trying to start the 3270 emulation program without the menu interface using the -e option:

This indicates that the 3270 style file you are using (either one you specified on the command line or the default style file specified in the configuration file) does not include any display sessions which can be used without the menu interface. For a display session to be activated when running without the menu interface, it must be marked as “enabled at initialization,” and must use a screen model that can be displayed on your terminal’s screen.

Start the 3270 emulation program without the -e option to run it with the menu interface, load the style file, assign at least one session to be active at initialization and to use an appropriate screen model, and save the new style. You can then restart the program without the menu interface using the new style file.

See Customizing display sessions and **Customizing printer sessions**, in chapter 5, for more information on the “enabled at initialization” option and on screen models.

Cannot Tell Which Terminal Keys Correspond to 3270 Keys

Press KEYS to display the View and Define Key Definitions dialog, or press ACTIONS to return to the main screen and then choose Key definitions from the Customize menu.

Characters on Display do not Match Keys Pressed

When the letters, numbers and punctuation you press on the keyboard show up as other characters on the display, your keys may have been remapped. Verify that your keyboard is mapped correctly. Press KEYS to view your current key definitions.

If you started the program with the -t (user-defined translation table) option, check that you are using the correct translation table file and that your operating system and your terminal hardware are able to support the character set defined by this translation table. See chapter 2 of this manual for more information on user-defined translation tables.

If you started the program with the -i (international) option, check that you are using the correct host language and that your operating system and terminal hardware are set up to use the ISO 8859 character set.

If you started the program without either option, check that you are not using a style file set up with a different host language. If the style file was setup using -i and includes a host language other than the default, you will not be able to use the keyboard mappings from the style file if you start the 3270 emulation program without -i; all keys revert to their default mappings.

See chapter 5, Customizing 3270 Emulation, for more information on key definitions.

Customize Menu Options Disabled (Grayed)

If all options on the Customize menu except Key Definitions are disabled (grayed), this is because your system administrator has not given you permission to customize LINKix 3270's default settings. See your system administrator if necessary.

Solving Problems

Display Colors are Different from Those Selected

If the colors displayed on your monitor differ from those you selected, check your display type (in the Miscellaneous options dialog) and your color settings.

Verify that your terminal supports the required color and highlighting options, that the display type (mono or color) specified for your terminal matches the terminal you are using and that the terminal colors and attributes associated with each host color and attribute are correct. .

Normally, you can remap colors for each 3270 session. See chapter 5, Customizing 3270 Emulation, for more information on specifying your display type and mapping session colors.

Display is Incomplete or Hard to Read

Verify that the display model number (screen size) which the 3270 session uses is the right one for your terminal. See chapter 5, Customizing 3270 Emulation, for more information.

Also check that the TERM environment variable is set to the correct value for the type of terminal you are using.

Note that the last line of your screen maybe shared between the status line and the last line of the 3270 emulation screen (for example, if you are using display model 2, 24 by 80, on a 24-line screen). To switch the display between the status line and the last line of the 3270 emulation screen, press STAT TOG (Status Line Toggle). On any screen size, you can also use this key to turn off display of the status line; it will reappear when information on it changes (e.g., to show a “Do Not Enter” message).

Error message: “No 3270 sessions in configuration file”

If you get this error message when you start the 3270 emulation program, the system administrator has included you in the configuration as a 3270 emulation user but has not configured any sessions for you to use. See your UNIX system administrator.

Error message: “No 3270 users in configuration file”

If you get this error message when you start the 3270 emulation program, the system administrator has not configured any users of the UNIX computer as 3270 emulation users. See your UNIX system administrator.

Error message: "User (UserID) not found in configuration file"

If you get this error message when you start the 3270 emulation program:

- ◆ The UNIX system administrator has not included you in the configured list of 3270 emulation users
- ◆ You are included as a 3270 user in the configuration, but you're currently logged on with a different user ID from the one specified in the configuration

Failed to initialize Windowing system

If you receive this message when trying to start the character-based 3270 emulation program in an X-Window, this generally indicates that the window is too small to display the program's main screen. The screen requires a window large enough to display at least 80 by 24 characters. Correct the window size and retry.

Fields Cannot be Changed or Edited

The contrast on your terminal maybe misadjusted, so grayed and non-grayed disabled (grayed) fields appear the same. Adjust the contrast.

File Menu Options Disabled (Grayed)

If the Open, New, Save and Save As options on the File menu are disabled (grayed), your system administrator has not given you permission to customize LINKix 3270's default settings. See your system administrator if necessary.

File Transfer Process Appears Suspended or "Hung"

If a file transfer process appears to be suspended or "hung" without having transferred any data, switch to the 3270 emulation display (unless you're running 3270 emulation in the background). Some host systems may write error messages to the 3270 display instead of using TRANSnnn file transfer messages. If no host messages are displayed, check the Messages list box in the Control File Transfer dialog. This may include the message "Error - host initiated transfer in screen mode". If so, there's an error in the host 3270 configuration. Disable the session using the Control Display Sessions dialog box (this is the only way to clear the condition) and contact your system administrator. Do not try to start any more file transfers on this session until the host configuration has been corrected.

Solving Problems

HLLAPI Application does not Work

If a HLLAPI application does not run, make sure:

- ◆ You are using the right program name for the application
- ◆ You are using the same login ID for the HLLAPI application program and for the 3270 emulation program; the two programs must run with the same user ID
- ◆ The session ID which the HLLAPI application is setup to use matches the session ID of one of your 3270 display sessions and this session is active and is not in use by another HLLAPI application or by a file transfer process.

For more information, refer to the *HLLAPI Programmer's Guide*.

Host Language Option Disabled (Grayed)

The Host language option in the Customize menu, which allows you to specify the language variant of EBCDIC used by the host computer, is only available if you started the program using the `-i` (international) option. If you need to use this feature, restart the program with the `-i` option (see chapter 2 for more information).

Host Response Times Option Disabled (Grayed)

Your system administrator has not configured LINKix 3270 to display host response times or has not given you permission to view them. If you want to use this option, ask your system administrator to configure it.

Last Transaction Time Indicator (LTTI) does not Appear

Your system administrator has not configured LINKix 3270 to display host response times or has not given you permission to view them. If you want to use this option, ask your system administrator to configure it.

Main Screen Invoked Instead of 3270 Session

If you see the main screen when you start the 3270 emulation program, instead of a 3270 display, none of your 3270 sessions are set to be active at initialization. Update your style file so that a session is active at initialization. Refer to Customizing Display Sessions in chapter 5.

Model Type Radio Group is Disabled (Grayed)

Your system administrator has configured your 3270 sessions so that you do not have permission to override this option from the emulation program, or the session is enabled (the session must be disabled before you can change the model type). If you want to be able to make this change while in the 3270 emulation program, ask your system administrator to change the configuration.

NetView Alerts Option is Disabled (Grayed)

If the NetView Alerts option in the Control menu is grayed, this means that the system administrator has not given you permission to send NetView alerts.

If your network supports NetView and you want to use this option, ask your system administrator to change the configuration.

NetView Alert not Accepted

If you try to send a NetView alert from the 3270 emulation program and receive an error message indicating that the host did not accept the alert, check first that you sent the correct alert to the correct host. Alerts are always sent on the current 3270 session to the host for that session. Your system administrator should tell you which alerts you can send to which host; if you send one to the wrong host it may not be accepted.

If you are sure that the alert was correct for this particular host, this probably indicates a problem with the host's NetView program. Contact your system administrator or the host personnel.

Printer Session does not Function

If a printer session does not function, you maybe attempting to use a session that's not enabled. To correct the problem, enable the printer session (see the steps in chapter 3, **Controlling 3270 Emulation**, for details on enabling a printer session).

A. Default Keyboard Mappings

This appendix describes the default keyboard definitions for your terminal during 3270 emulation. For ease of reference the keys are grouped first by function, then alphabetically.

You can check the definition of an individual key or key combination, or change a key's definition, either from the main menu or during a 3270 emulation session. Press KEYS to call up the View and Define Key Definitions dialog. See Customizing Key Definitions, in chapter 5, for more information about customizing the keyboard for your applications.

The tables in this section describe the attributes and functions associated with 3270 emulation and file transfer. The tables also list the specific terminal keys and key combinations assigned to these attributes and functions.

The functions are grouped as follows:

- ◆ Program function keys
- ◆ Program access keys
- ◆ Printer control keys
- ◆ 3270 system keys
- ◆ Miscellaneous system keys (including session selection keys)
- ◆ Edit keys
- ◆ Character keys

The keys assigned to these functions and attributes can be remapped (by you or your system administrator) to match your terminal type or a specific host application. To see what the current keyboard mappings are, press KEYS during a 3270 session, and the View and Define Key Definitions dialog will be displayed.

Program Function Keys

The function of each of these keys is determined by the particular host application; different applications will use them for different purposes or may not use all of them.

Host Key	Default	Host Key	Default
PF1	F1	PF13	Esc F1
PF2	F2	PF14	ESC F2
PF3	F3	PF15	ESC F3
PF4	F4	PF16	ESC F4
PF5	F5	PF17	ESC F5
PF6	F6	PF18	Esc F6
PF7	F7	PF19	Esc F7
PF8	F8	PF20	Esc F8
PF9	F9	PF21	ESC F9
PF10	F10	PF22	Esc F10
PF11	F11	PF23	EscF11
PF12	F12	PF24	ESC F12

Program Access Keys

3270 key	Default key combination	Function
ATTN	CTRL + A	Attention: requests the attention of the host application. The actual function of this key is determined by the host application; some do not support it.
CLEAR	CTRL + C	Clear: clears the display of all data and formatting information. The host application may or may not redraw the screen.
CURS SEL	ESC C	Cursor Select: selects application fields that use a light pen rather than the keyboard to send input to the host. To select a field, move the cursor to the start of the field and press CURS SEL.
ENTER	ENTER CTRL + M	Enter: sends data to the host for processing. Your keyboard is locked until the host responds.
PA1 PA2 PA3	ESC ESC F1 ESC ESC F2 ESC ESC F3	program Access keys: instruct an application to perform a single function rather than processing display data. The function performed by each of these keys is determined by the host application; some programs do not support them.

Printer Control Keys

3270 key	Default key combination	Function
DEV CNCL	ESC D	Device Cancel: cancels a local copy printing operation, or (if you receive a message on the status line indicating that the printer is busy or has an error) it cancels the print attempt and continues 3270 emulation.
PRINT	CTRL + P	Print: sends a copy of the 3270 display to the local printer.

3270 System Keys

3270 key	Default key combination	Function
SYS REQ	CTRL+ Y	System Request: switches to the host control session and clears the screen. To return to the application session, press SYS REQ again. The host application redraws the screen. In the remote BSC product, this key sends a TEST REQ message to the host.
ALT CURS	ESC A	Alternate Cursor: toggles between a normal cursor and a “visible” cursor. (“Visible” means that the cursor appears highlighted to make it easier to see; the appearance depends on your terminal, and some terminals cannot distinguish between this and a normal cursor.)
RESET	ESC R	Reset: unlocks the keyboard when it locks due to an error. (If the keyboard locks due to a printer error, press DEV CNCL, ESC D .) Note: This key cannot be used to unlock the keyboard if the WAIT message appears on the status line. In the remote BSC product, this key cancels the last transmission to the host if it has not yet actually been sent.

Miscellaneous System Keys

3270 key	Default key combination	Function
EXIT	CTRL + X	Ends 3270 emulation and returns to the operating system prompt.
TEMPEXIT	CTRL + Z	Performs a temporary exit to the UNIX command prompt while the 3270 emulation program is running. This starts a new UNIX shell of the type specified by the SHELL environment variable (the default is <i>/bin/sh</i>). To return to 3270 emulation, use your terminal's EOF sequence (typically CTRL + D).
REDRAW	CTRL+R	Redraws the screen; you can use this keystroke if the display has become corrupted, for instance by another program running on your terminal.
ACTIONS	CTRL + U	Displays the main menu screen for 3270 emulation. To move back to the current 3270 session, press ESC .
KEYS	CTRL + K	Displays the View and Define Key Definitions dialog box.
FILE XFR	CTRL + F	Displays the File Transfer dialog to monitor, start or terminate file transfer processes.

Default Keyboard Mappings

3270 key	Default key combination	Function
STAT HELP	CTRL + O	Status Line Help: interprets the contents of the current session's status line, and displays the information on your screen in text format. To remove the status line help information from the screen when you have finished viewing it, press STAT HELP again.
NUM OVR	ESC N	Numeric Override: lets you enter non-numeric information in a field designated as a numeric field.
STAT TOG	CTRL + W	Status Line Toggle: switches between displaying the last line of the 3270 display and the status line. Used only when your terminal's screen does not have enough lines to display the Status Line below the 3270 screen. See Appendix B for more information. On any screen size, you can also use this key to turn off display of the status line; it will reappear when the information on it changes, for example to show a "Do Not Enter" message.
RECORD	ESC S	Records a sequence of 3270 keystrokes so that you can replay them later. Press RECORD followed by one of the keys PF1 - PF24 (these are used as identifiers for up to 24 keystroke sequences) to start recording; press RECORD

Default Keyboard Mappings

3270 key	Default key combination	Function
		again to finish recording and save the keystrokes, or press QUIT to abandon recording.
REPLAY	ESC P	Replays a sequence of 3270 keystrokes that you recorded using the RECORD key. Press REPLAY followed by one of the keys PF1 - PF24 (these are used as identifiers for up to 24 keystroke sequences) to identify the keystroke sequence to replay; press QUIT to abandon replay. Also, press REPLAY to continue when the sequence you are replaying has paused to allow you to enter variable data.
PAUSE	ESC W	Pauses while recording a sequence of 3270 keystrokes so that you can insert variable data. Press PAUSE, type the variable data (which is not recorded), and press PAUSE again to continue recording.
QUIT	CTRL + T	Abandons recording or replaying a keystroke sequence. When recording, QUIT stops recording and restores the previous keystroke sequence, if any, assigned to the PF key in use; when replaying, QUIT stops the output of further keystrokes from the recorded sequence.
3270 keysNEXTSESS	CTRL + V	Moves to the next active 3270 display session in session number order.

Session Selection Keys

The Session Selection keys are used to jump to a particular 3270 session. (You can also get to a particular session by pressing NEXTSESS until you get to the session you want.)

Session	Keys
SESS 1	Esc 1
SESS 2	Esc 2
SESS 3	Esc 3
SESS 4	Esc 4
SESS 5	Esc 5
SESS 6	ESC 6
SESS 7	Esc 7
SESS 8	Esc 8
SESS 9	Esc 9
SESS 1-10	Esc 0

Edit Keys

3270 key	Default key combination	Function
UP	UP	Moves the cursor up 1 line. If the cursor is on line 1 (at the top of the display) it moves to the last line.
DOWN	DOWN	Moves the cursor down 1 line. If the cursor is on the bottom line of the display it moves to the first line.
LEFT	LEFT	Moves the cursor 1 space to the left. If the cursor is at the first character position in a line, it moves to the last character position of the preceding line. If the cursor is at the first character position of line 1, it moves to the last character position on the bottom line of the display.
LEFT DB	PGUP	Left Double: moves the cursor left two spaces.
RIGHT	RIGHT	Moves the cursor 1 space to the right. If the cursor is at the last character position in a line, it moves to the first character position of the next line. If the cursor is at the last character position of the last line of the display, it moves to the first character position on the first line of the display.

Default Keyboard Mappings

3270 key	Default key combination	Function
RIGHT DB	PGDN	Moves the cursor right two spaces.
BACKSPACE	CTRL + H	Backspace: moves the cursor 1 space to the left. This key provides the same function as LEFT.
TAB	TAB CTRL + I	Tab: moves the cursor to the first position of the next input field. If the display is unformatted, the cursor moves to the first position of line 1.
BACKTAB	BACKTAB CTRL + B	Moves the cursor to the first position of the current input field, or, if the cursor is already in this position, it moves the cursor to the first position of the preceding input field. If the display is unformatted, the cursor moves to the first position of line 1.
HOME	HOME	Home: moves the cursor to the first position of the first input field of the display. If the display is unformatted, the cursor moves to the first position of line 1.
NEW LINE	CTRL + N	New Line: In an unformatted display, the cursor moves to the first position in the next line. In a formatted display, it moves forward to the first unprotected character position on the next line. When the display has no unprotected lines, the cursor moves to the first position on line 1.

Default Keyboard Mappings

3270 key	Default key combination	Function
INSERT	INSERT	Insert: New characters are inserted at the cursor and existing text shifts to the right. To quit insert mode, press RESET, ESC R . Sending data to the host with an attention key also clears insert mode.
DELETE	DEL	Delete: if the cursor is in an input field, the character at the cursor position is deleted and characters to the right of the cursor shift 1 space to the left, otherwise, the terminal beeps and the keyboard locks until you press RESET, ESC R .
DUP	CTRL + D	Dup: puts the Dup character (“*”) in the selected field and moves the cursor to the first position in the field. Some host programs use this key.
FMARK	ESC F	Field Mark: puts the Field Mark character (“;”) in the selected field. Some host programs use this key.
ERASE EOF	CTRL + E	Erase to End of Field: erases all data from the cursor position to the end of the current field. If the display is not formatted, all characters from the cursor to the end of the display are erased. If the cursor is not in an input field, the terminal beeps and the keyboard is disabled until you press RESET (Esc R).
3270 key	Default key combination	Function

Default Keyboard Mappings

ERASE INP	ESC E	Erase Input: erases all input fields and moves the cursor to the first input field. If the cursor is not in an input field, the terminal beeps and the keyboard is disabled until you press RESET, ESC R .

Character Keys

Character keys include uppercase and lowercase letters, numbers, and punctuation.

If you started the 3270 emulation program with the -i (international) option, the set of valid characters will vary with your choice of host language. You should select the host language you are using before attempting to remap character keys. The default in all cases is the standard character key corresponding to the character.

If you are not using the international option, the following punctuation keys and special characters can be used:

(space)	!	"	#	\$	%	&	'	(	)
	*	+	-	°	/	:	;	<	>
	=	?	@	\	_	^	{	}	~

| (this appears as a split vertical bar)

In addition, the following characters maybe used, although your terminal may not be able to display them correctly on the screen.

Character	Displayed as	Default Mapping
¢	[	[
VERTBAR	o p]	]
NOT	] or ^	^

Default Keyboard Mappings

3270 Functions and Corresponding Keys

Function	Key(s)	Function	Key(s)	Function	Key(s)
ACTIONS	Ctrl + U	ERASE INP	ESC E	PA 1	ESC ESC F1
ALT CURS	ESC A	EXIT	CTRL + X	PA2	ESC ESC F2
ATTN	CTRL + A	FILE XFR	CTRL + F	PA3	ESC ESC F3
BACKSPACE	CTRL + H	FMARK	ESC F	PF1	F1
BACKTAB	Backtab CTRL + B	HOME	Home	PF2	F2
CLEAR	CTRL+ C	INSERT	Insert	PF3	F3
CURS SEL	ESC C	KEYS	CTRL + K	PF4	F4
DELETE	Delete	LEFT	Left	PF5	F5
DEV CNCL	ESC D	LEFT DB	PageUp	PF6	F6
DOWN	Down	NEW LINE	CTRL + N	PF7	F7
DUP	CTRL + D	NEXTSESS	CTRL + V	PF8	F8
ENTER	Enter CTRL + M	NUM OVR	ESC N	PF9	F9
ERASE EOF	CTRL + E	PAUSE	ESC W	PF10	F10

3270 Functions and Corresponding Keys			
Function	Key(s)	Function	Key(s)
PF11	F11	PF24	Esc F12
PF12	F12	PRINT	Ctrl + P
PF13	Esc F1	QUIT	Ctrl + T
PF14	Esc F2	RECORD	Esc S
PF15	Esc F3	REDRAW	Ctrl + R
PF16	Esc F4	REPLAY	Esc P
PF17	Esc F5	RESET	Esc R
PF18	Esc F6	RIGHT	Right
PF19	Esc F7	RIGHT DB	PageDown
PF20	Esc F8	SESS 1	Esc 1
PF21	Esc F9	SESS 2	Esc 2
PF22	Esc F10	SESS 3	Esc 3
PF23	EscF11	SESS 4	Esc 4

Function	Key(s)
SESS 5	Esc 5
SESS 6	Esc 6
SESS 7	Esc 7
SESS 8	Esc 8
SESS 9	Esc 9
SESS 10	Esc 0
STAT HELP	Ctrl + O
STAT TOG	Ctrl + w
SYS REQ	Ctrl + Y
TAB	Tab
	Ctrl + I
TEMPEXIT	Ctrl + Z
UP	up

B. Status Line Information

This appendix describes and explains the messages and symbols that appear on the status line. The messages are grouped into the following categories:

Session Status Messages

- ◆ Do Not Enter Messages
- ◆ Communications Check Codes
- ◆ Indicators:
 - ◆ Last Transaction Time Indicator (LTTI)
 - ◆ Numeric Indicator
 - ◆ Insert Indicator
- ◆ Identifiers:
 - ◆ Printer Identifier
 - ◆ Record/Replay ID
 - ◆ LU Name
 - ◆ Cursor Position
 - ◆ Session ID
- ◆ Program Check Codes

Status Line

The way in which the status line is displayed on the screen depends on your terminal's hardware and software and on your 3270 customization, as follows:

- ◆ If your terminal's hardware and its terminfo entry both support a status line to which user programs can write, you can customize 3270 sessions to write the 3270 status line to the terminal's status line. Check with your system administrator for more information on terminfo requirements.
- ◆ If this feature is not supported, or if you choose not to use it, LINKix will use the last line of the terminal's screen to display the status line. If your terminal's screen has more lines than it requires for the 3270 emulation display (for example, if you are using screen model 2, which has 24 lines, on a 25-line screen), the status line is shown below the 3270 emulation display. If not (for example, if you are using screen model 4, which has 43 lines, on a 43-line screen), the status line shares the last line of the screen with the last line of the 3270 emulation display; pressing the STAT TOG (Status Line Toggle) key (default keystroke **CTRL + W**) alternates between displaying the status line and displaying the last line of the 3270 display.

You can also use STAT TOG to turn off display of the status line. It will reappear when the information on it changes, for example to show a "Do Not Enter" message.

Status Line Display for Keystroke Recording and Replay

When you use the RECORD or REPLAY keys, the status line display is replaced by a list of the identifiers assigned to recorded keystroke sequences. The normal status line display is restored when you select an identifier for the keystroke sequence to record or replay. At any time other than when you are about to select an identifier for a keystroke sequence, the status line display is as described in this appendix.

Status Line Help

To obtain more information about the contents of the current session's status line, you can use the STAT HELP key (default keystroke **CTRL + O**) to display a screen of help information. This screen interprets the status line information for the current session and displays it as text strings instead of the symbols used on the status line. The screen includes the session state, emulator state (for example a communications check code or a Do Not Enter message), input state (insert/replace, numeric), LU name, session ID (if one is assigned), last host response time, and the name of the printer or file to which local copy printing on this session is directed.

Status Line Information

When you have finished viewing this help information, press STAT HELP again to remove it from the screen.

The status line is shown below.

SB#	X o-o()	LTTI 00.03.1	NUM ^	/usr/bin/lp	COL001	03-020 B
Session Status	Do Not Enter	Comm. Check or LTTI	Numeric Field Insert indicator	Printer Identifier or Record/ Replay ID	LU Name	Cursor Position Session ID

Session Status

Session Status messages indicate whether or not the UNIX computer and the host system are ready to communicate on this session or if they're already communicating.

Do Not Enter

Do Not Enter messages indicate the current state of the 3270 session. Some of these messages indicate normal conditions; others indicate that your 3270 keyboard is disabled temporarily and specify the reason for this.

Comm. Check or LTTI

Communications Check codes indicate the current state of the communications link to the host. Some of these messages indicate normal conditions which occur during initialization; others indicate problems with either hardware or configuration.

The Last Transaction Time indicator (LTTI) indicates the time taken for the most recent transaction with the host. This information is only displayed if there is no communications check code displayed; it also does not appear if your system administrator has not given you permission to view response time information.

Numeric Field

The Numeric field indicator appears if the cursor is in a numeric-only field.

Status Line Information

Insert Indicator

The Insert indicator shows when your keyboard is in insert mode (you can insert a character at the cursor location, not typeover). Press the INSERT key to enter insert mode; press RESET to exit insert mode.

Printer Identifier

The Printer identifier shows the name of the printer or file to which local copy print requests on the current 3270 session are directed. This is specified in the Customize Display Session dialog box (see chapter 5, **Customizing 3270 Emulation**).

Record/Replay ID

The Record / Replay ID messages remind you of the PF key identifier for a keystroke sequence you are recording or replaying.

Message	Explanation
RECORD nn	This message indicates that you are recording a keystroke sequence identified by the PFnn key.
RECORD PF KEY (list of PF keys)	This message indicates that you're recording a keystroke sequence, and shows the PF keys that already have sequences assigned to them. Press a PF key to identify the sequence.
REPLAY nn	This message indicates that you are replaying a keystroke sequence identified by the PFnn key.
REPLAY PF KEY (list of PF keys)	This message indicates that you want to replay a keystroke sequence, and shows the PF keys that already have sequences assigned to them. Press one of the PF keys shown to replay the sequence.
BUFFER LOW (mm)	This message appears when you've entered 1475 keystrokes or more while recording a keystroke sequence (the maximum number of keystrokes in any one recorded sequence is 1500). <i>mm</i> is the number of remaining keystrokes you can record before reaching the limit (up to 25).

LU Name

The LU Name shows the name of the 3270 LU being used for the current 3270 session. This name is specified in the configuration file; you can change it if your system administrator has given you permission to remap sessions (see chapter 3, **Controlling 3270 Emulation**).

Cursor Position

This information shows the cursor's current position on the screen (row-column).

Session ID

This is the one-character session ID of the current 3270 session, if one is assigned. You can change the session ID by using the Customize Display Session dialog box (see chapter 5, **Customizing 3270 Emulation**).

Session Status Messages

The Session Status messages tell you whether or not your computer and the host system are ready to communicate.

Message	Explanation
S	At initialization, this indicates that the 3270 emulation program is not yet connected to the host; it should change to one of the following messages after a few seconds. If the session has started but has been interrupted, this indicates that the connection to the host has failed. There should be a communications check code (-+z_nnn) displayed; check this in Communications Check Codes, later in this appendix.
SB?	The 3270 emulation program is on line to the host, but the 3270 LU has not yet been activated. This should change to the following message after a few seconds.
SB*	The display station is in session with the host component called SSCP (System Services Control Point) which handles logon, and the LU has been successfully activated. Either you have not yet logged on to a host application (in which case you should enter the appropriate logon command for the required host application) or you have used the SYSREQ key to switch to the host control session (in which case you should use SYSREQ again to return to the application session). In BSC there is no SSCP session, although this message may flash in the status line during startup.
SB#	The 3270 emulation session is bound to a host application program. The name of the host application is shown in the Control Display Sessions dialog.
BSC	The 3270 session is connected to a BSC host.

Do Not Enter Messages

Some Do Not Enter messages indicate normal conditions that occur during most sessions; others indicate that your keyboard is disabled temporarily due to an error or indicate abnormal conditions which need to be corrected by your system administrator. See the following table for details of these messages.

Message	Explanation
x - f	Function Unavailable: You used an invalid key. For example, you used a Program Function key when the host SSCP is waiting for a logon message. Press RESET to continue.
x o-o ()	Printing: Local Copy printing in progress. No action necessary.
x o-o () ()	Printer Busy: Your printer is printing data from the host and cannot perform the local copy print you requested. The RESET key does not unlock the keyboard when this message is displayed. Press DEV CNCL to continue.
x o - _	Printer Error The Local Copy printer or print spooler is not working or LINKix 3270 cannot write to the file to which local copy output is directed. The print request has been cancelled. (Note that _ here indicates a zero character, which appears on most terminals as a crossed-out circle.) Press DEV CNCL to proceed. Rectify the printer problem and retry the operation.
x ?+	Keystroke Lost: A keystroke has been lost. Either you were typing at a faster rate than can be serviced, or a keystroke you typed was not understood. Press RESET to continue.
X PROGnnn	Program Check Code: LINKix 3270 has detected a protocol error in the data coming from the host. <i>nnn</i> is a 3-digit number. Press RESET to continue. If the error recurs, note the program check code (<i>nnn</i>), and report the problem to your system administrator. (Program check codes are listed at the end of this appendix.)

Status Line Information

Message	Explanation
x *>	Field Full: An attempt has been made to insert data into a full field. Press RESET to continue. If the field does not appear to be full, there may be blank characters at the end of the field; to remove them, move the cursor past the last visible character and use ERASE EOF.
x <*>	Protected: An attempt has been made to enter data into a protected screen position. Press RESET to continue.
x *NUM	Range Check: An invalid character was entered for a numeric field. Press RESET to continue.
X SYSTEM	System Lock: The host has locked the keyboard. Press RESET to continue.
x ()	Wait: The system is waiting for the host to reply, or for local processing to complete. Wait. If the situation persists for an unreasonable time, contact the system administrator. You may be able to stop and restart the 3270 session by using the ATTN key or SYSREQ and logging off.
X () FT	Wait for File Transfer: File transfer is in progress. Wait for the file transfer to complete. You can use the Control File Transfer dialog to check the progress of the file transfer; some hosts may also write messages to the screen.
X () AL	Wait for Alert: You sent a NetView User Alert to the host from the 3270 control interface. Wait for LINKix 3270 to finish sending the alert.

Communications Check Codes

Communications Check codes indicate the status of your communications link with the host. They appear on the status line as `-+z_nnn`, where `nnn` is a 3-digit number. All `5nn` codes correspond exactly to what an IBM controller would display when an error occurs. All `9nn` error codes are specific to LINKix 3270.

500 - Link Inactive

All the links configured for use by the connection were inactive.

501 - Link Failure/Line Disconnected

The link from LINKix 3270 to the host is not active (DSR failure). This message remains until the communication link is restored.

502 - Transmit Data Interrupt

A problem occurred in the link between the host and LINKix 3270 while data was being transmitted (CTS failure). Report the message to your system administrator.

504 - Link Not Established

Communications are not established with the host. This message disappears as soon as communication is established with the host. (Could also be CD failure.)

505 - Connection Terminated by Host

The host terminated the connection used by your 3270 session (DISC received).

510 - Physical Unit Not Active

The host has sent a DACTPU message, or has not sent an ACTPU message. This may indicate a problem with the host's configuration. Report the message to your system administrator.

519 - Receive Overrun

A connection error has occurred due to errors on the communications link.

520 - No Polls Received

Host is not polling the server. Report the problem to your system administrator.

Status Line Information

521 - No Flags Received

The modem is functioning, but the host is not sending data. Note the error code number and report it to your system administrator.

522 - Read Control Block Overflowed (BSC only)

The control unit's Read Control Block has overflowed. Verify that the connection between the controller and the host is operational. The error log may indicate a specific cause; check your controller's documentation for recovery suggestions.

525 - Controller Problem

A connection problem prevents establishing or reestablishing host communication. Check the error log for specific cause.

528 - Invalid Frame Received

Invalid data was received from the host ÷ LINKix 3270 sent a ÷ command rejected message. This may be a protocol error at the host, or the data may have been corrupted. Check the error log for more information.

529 - Abnormal Modem Response

A connection problem occurred because the modem did not supply the expected signals. Check the modem connections.

530 - Connection Failure/Write Timeout

The connection between the controller and the host is down ÷ usually a clocking problem or missing CTS. Check the error log for a specific cause, and check your controller's documentation for recovery suggestions.

531 - Receive Limit Exceeded (BSC only)

The control unit's Receive Limit has been exceeded ÷ the maximum number of consecutive receive timeouts occurred before LINKix 3270 dropped the connection. (A receive timeout happens when LINKix 3270 waits three seconds without receiving a response to the message it just sent.) Report this error to your system administrator.

532 - No SYNCS Detected (BSC only)

No SYN characters were detected on the line for 20 seconds. Verify that the connection between the controller and the host is operational. The error log may

indicate a specific cause; check your controller's documentation for recovery suggestions, or report the message to your system administrator.

533 - No ETX or ETB before ENQ (BSC only)

The control unit did not receive an ETX or ETB before receiving an ENQ. Report the message to your system administrator.

534 - No Response to Last Block (BSC only)

The control unit did not receive a response to the last block it sent, and it sent the specified number of ENQs without a response from the host before LINKix 3270 dropped the connection. Report the message to your system administrator.

535 - Transmit Limit Exceeded (BSC only)

The control unit's Transmit Limit was exceeded. The data was retransmitted to the host, unsuccessfully, the maximum number of times specified before LINKix 3270 disconnected. Report the message to your system administrator.

536 - Received ACKOs Instead of ACK1s or vice-versa (BSC only)

The control unit received 15 consecutive ACK0s instead of ACK1s (or vice-versa). Report the error to your system administrator.

537 - Call timed out

An attempt to connect to the host through an X.25 switched virtual circuit has timed out. This can happen, for instance, if incorrect channel number ranges are assigned when configuring the X.25 service. Report this error to your system administrator.

539 - QDISC Received

The connection has been lost due to receipt of a QDISC command from the host.

580 - SVC failed

The switched virtual circuit to the host has been cleared down (broken) either by the host or by the X.25 network itself.

581 - PVC failed

The permanent virtual circuit to the host has been reset (broken) either by the host or by the X.25 network itself.

Status Line Information

582 - Call failed to connect

An attempt to connect to the host through a switched virtual circuit has failed.

583 - SVC protocol error

A protocol error has been detected on an X.25 switched virtual circuit. The circuit has been cleared down and the connection to the host lost.

584 - PVC protocol error

A protocol error has been detected on an X.25 permanent virtual circuit. The circuit has been reset and the connection to the host lost.

655 - SABME received

The Token Ring connection has been lost because the host sent a SABME command.

656 - FRMR sent

The Token Ring connection has been lost because the host sent a frame containing an error (and LINKix responded with a frame-reject).

657 - FRMR received

The Token Ring connection has been lost because the host sent a frame-reject command.

658 - Connection terminated by host

The Token Ring connection has been terminated by the host.

659 - Connection lost

The Token Ring connection has been lost.

730 - Control Unit No Longer Responding (DFT only)

The control unit is no longer responding to its devices. Check all hardware connections, or contact your system administrator.

731 - DFT PDAT Buffer Too Small (DFT only)

The control unit exceeded the agreed upon maximum RU size. Contact the host system personnel and reconfigure the host for larger RU sizes.

732 - Device Has Been Reset By Control Unit (DFT only)

This is a transitional condition; LINKix 3270 should go on-line almost immediately after this condition is reported and the status line should be cleared. No action is required unless the condition is not cleared, in which case you should contact your system administrator.

739 - Error Logged to the Error Log File (DFT only)

The error could not be displayed on the status line; it has been logged to the error log file. Contact your system administrator, who should examine the error log file and check the controller's documentation for an explanation of the reported error code.

900 - Idle State

Session has not yet tried to find an available LU. Session is disabled.

910 - Hunt State

Session is looking for an available LU.

920 - Local Node Not Available

The local node is not active. Report the problem to your system administrator.

921 - LU Not Available

Session tried to find an LU, but the LU was not available.

922 - LU in Use

Session found an LU, but it was in use.

923 - No Free LU in Pool

The session is configured to use an LU pool, but all the LUs in the pool are already in use, or one or more LUs cannot be used because their connection is not active. (If your system administrator has given you permission to remap sessions, you can remap the session to use a different LU or LU pool.)

924 - Node Failed

Session found an available LU, but failed because of a resource shortage at the node.

Indicators

Last Transaction Time Indicator

The Last Transaction Time Indicator (LTTI) measures the time between pressing ENTER or an AID key to send data to the host, and receiving the response from the host. (It only records response times for application data, not for messages to SSCP.) If your system administrator has not given you permission to view host response times for your 3270 emulation sessions, this indicator does not appear.

This information occupies the same location on the status line as Communications Check codes, which override LTTI. Information is displayed as LTTI *mm:ss.t*, in which *mm* represents minutes, *ss* represents seconds and *t* represents tenths of seconds.

Numeric Indicator

When the cursor is in a field defined as being numeric only, the symbol NUM is displayed on the status line, and the only valid characters you may enter are numerics (0 - 9), the minus (-), period (.) and comma (,) characters, and the DUP character; any other characters will cause a *NUM message to be displayed. You can override the numeric restriction, if the host will accept non-numeric data, by using the NUM OVR keystroke.

Insert Indicator

When your keyboard is in , the caret symbol (^) appears on the status line. When you enter characters under Insert mode, they are inserted at the cursor position and characters to the right of the cursor are moved to the right. When Insert mode is not active, characters entered from the keyboard overwrite the existing data at the cursor position. Press the INSERT key to enter insert mode, and press the RESET key to leave insert mode.

Identifiers

Printer Identifier

The Printer Identifier message displays the name of the UNIX printer or print spooler to which local copy print data is sent for this session, or the file name if you specified Print to File. See chapter 5 for details of how to specify the printer or file name for a 3270 session. Where a file name is shown, the full directory path is included, unless this contains too many characters for the available space (in which case only the file name is shown).

LU Name

The LU Name indicates the LU which is being used for the current session. If the 3270 session is mapped to an LU pool, this will be the name of the actual LU within the pool and not the name of the pool.

If you're reporting an error condition to your system administrator, you should also report this LU name for the session on which the problem occurred.

Cursor Position

This message shows the current row-column position of the cursor on the 3270 display. It appears as *rr-ccc*, where *rr* is the row number and *ccc* is the column.

Session ID

This indicates the session ID of this session, if you have assigned one. The session ID appears as a single uppercase character.

You must assign a session ID if the session is to be used for file transfer or for running HLLAPI applications; otherwise it is optional. Use the Customize Display Session dialog to assign a session ID. (Note: You must save the configuration with the new session ID, exit from 3270 emulation and restart the emulation program before the session ID is effective.)

Program Check Codes

Program check codes appear on the status line of the 3270 emulation program as PROG *nnn* messages. They indicate protocol or data stream errors detected by either the local node or the 3270 emulation program.

401	Invalid command received Invalid alias for SFE, MF, or SA
402	Out of range address for SEA, RA or EUA order MF order addressed non-field attribute location
403	Data stream data after Rd, Rd Mod, EAU command Invalid parameter following SFE, MF, SA, GE, or RA order
404	Data stream terminated before end of order
413	Function not supported
420	Exception response chain received when BIND specified definite response
421	Definite response chain received after BIND specified exception response
422	New response chains not allowed
423	F1 bit not allowed for FM data
430	Sequence number error
431	Chaining error
432	Bracket error
433	Data traffic reset
434	Direction error
442	Request not executable

443	CD required
445	ACTLU type not COLD or ERP
450	Profile error on BIND
451	Primary protocol error on BIND
452	Secondary protocol error on BIND
453	Common protocol error on BIND
454	Screen size error on BIND
455	LU profile error on BIND
456	Negotiable BIND rejected (sense code 0821 sent)
470	Unsupported code point received
471	Extended data stream function not supported
472	Read state error
474	The device does not support extended data stream
475	WCC had öStart Printì but not last structured field
497	Connection between local node and 3270 link service has failed
498	Negative response received
499	An internal error occurred; see the error log file for details

Status Line Information

C. File Transfer Messages

This appendix explains all the file transfer error messages generated by the 3270 emulation program and the TRANSnnn messages generated by the host file transfer program. It includes the following:

- ◆ Messages generated by LINKix 3270 during file transfer, when using the menu system.
- ◆ Messages generated during command-line file transfer.

The second group of messages includes some which are generated by the host program, IND\$FILE; these may also appear in the File Transfer dialog box. Others are generated by LINKix 3270 and apply only to command-line file transfer.

If your host system generates messages other than TRANSnnn (e.g., INWnnnn), file transfer can still be used but is not fully supported. When using the menu interface, the host file transfer message number and text are displayed as for TRANSnnn messages, but you will need to refer to your host documentation for explanations of the alternate messages.

Also, LINKix 3270 command-line file transfer does not fully distinguish between different INWnnnn messages. INW messages indicating a successful file transfer are replaced by the message TRANS098 and those indicating an unsuccessful file transfer by the message TRANS099.

File Transfer Messages

This section describes the file transfer messages generated during file transfer processes initiated from the menu system. These messages are displayed in the File Transfer dialog box.

Completed successfully

The operation is complete; no action is required.

Error - host initiated transfer in screen mode

The host tried to start the file transfer in screen mode, which LINKix 3270 does not support. Use the Control Display Sessions dialog to disable the session (this is the only way to clear this condition) and report the error to your system administrator.

File transfer aborted by user

You canceled the transfer operation. No action is required. (This message occurs if you cancel the file transfer before the host has accepted your file transfer request, or after the host has completed it.)

File transfer could not be started due to lack of local resources

LINKix 3270 cannot allocate enough memory for the data buffer needed for a file transfer process. This is generally because too many other processes are running at the same time. Retry when other file transfer processes have finished or when there aren't so many users on the system. If this problem occurs repeatedly, contact your system administrator.

General I/O fault

The UNIX computer has a problem reading or writing (this can occur for a number of reasons, including incorrect file access permissions). Retry the operation. If unsuccessful, consult the system administrator or technical support.

Not allowed to delete the specified local file.

A local file was created during an attempt to receive a host file, but the file transfer has failed. The access permissions of the file do not allow LINKix 3270 to delete it; you will have to remove it manually.

Not allowed to read from the specified local file.

You attempted to send a file to the host, but you do not have read permission for the file. The file transfer was canceled.

Not allowed to write to the specified local file.

You attempted to receive a file from the host but you do not have write permission for the local file or directory which was to receive the host data. The file transfer was canceled.

Number of bytes transferred: *nn*.

Information message, appears after completion of the file transfer command.

Operator cancel

You canceled the file transfer. No action is required. (This message occurs when you cancel a file transfer in progress, and the host returns an acknowledgment that the process has been canceled.)

Request received out of sequence

This usually indicates that the version of IND\$FILE (or other file transfer program) at the host is incompatible with LINKix 3270. Report full details to the system administrator.

Structured field not supported

LINKix 3270 supports structured field file transfer only if the mainframe administrator has set the query bit in VTAM. Report full details to the system administrator or host administrator.

The specified local file or directory does not exist

The local file name or directory you specified does not exist. The file transfer was canceled. Check for the correct file and directory and retry.

IND\$FILE File Transfer Messages

This section explains all the file transfer messages generated during command-line file transfer. Some of these are generated by IND\$FILE and may appear when transferring files from the command-line or with the menu system; note that the exact text of the message associated with a particular message number may vary with different versions of the host file transfer program. Other messages are generated by LINKix 3270 and appear only during command-line file transfer.

TRANS00: An error occurred during file transfer. File transfer was canceled.

An error occurred during file transfer that prevented the operation from completing successfully.

TRANS01 : File transfer command being processed.
The host file transfer program has responded to the send or receive file command.

TRANS02: Number of bytes transferred so far ==>
After the file transfer is started, this messages displays the count of transferred bytes as the file transfer progresses.

TRANS03: File transfer complete.
File transfer completed successfully.

TRANS04: File transfer is complete. Some records were segmented.
File transfer completed successfully. Some records were longer than specified host record length and were divided into multiple records.

TRANS05: The PC file name was incorrect or not found. File transfer was canceled.
The UNIX file name specified was not a valid file name, or (for sending to the host) the file with that name was not found. Check the correct file name and retry.

TRANS06: A required parameter was not specified. File transfer was canceled.
A parameter required in the file transfer command was not supplied. Check the correct syntax of the command for the appropriate host type, and retry the command.

TRANS07: The host connection was not established. File transfer was canceled.

The specified 3270 session was not active, or you did not specify a 3270 session in the file transfer command; LINK.3270 will find the configured session with the session ID that occurs first alphabetically and try to use that session. Activate the required session and retry, or retry the command specifying an active session.

TRANS10: The host has not responded. Switch to the emulator session.

You used the **T(nn)** option on a command-line file transfer to specify a timeout period in which the host must respond. This message appears every 30 seconds until the specified number of timeouts (**nn** 30-second periods) have expired or until the host responds. If you're not running 3270 emulation in the background, switch to the 3270 display session to see if the host has written any messages to the screen.

TRANS11: The host connection was lost. File transfer was canceled.

The connection with the host was lost during file transfer. This may be due to a link failure, or the session may have been deactivated. Check the reasons for the failure, reactivate the session and retry.

TRANS12: No session was found configured for HLLAPI

You issued a file transfer command from the command line without specifying a session identifier, and no 3270 session identifiers were configured. You should assign session IDs to one or more sessions, restart the 3270 emulation program so that the new session IDs take effect, and then reissue the command.

TRANS13: An error occurred sending the file to the host. File transfer was canceled.

This message normally means that the file transfer was aborted using the File Transfer dialog in the 3270 emulation program. In other cases, it indicates an unidentified error. Retry the command.

TRANS14: An error occurred receiving the file from the host. File transfer was canceled.

This message normally means that the file transfer was aborted using the File Transfer dialog in the 3270 emulation program. In other cases, it indicates an unidentified error. Retry the command.

File Transfer Messages

TRANS15: The required amount of host storage (memory) is not available. File transfer was canceled.

There was not enough storage on the host for the file transfer to be completed. Contact host personnel to arrange for more storage.

TRANS16: An incorrect request code was detected. File transfer was canceled.

This error should not occur during normal operation. Retry the command; if it continues to fail, contact technical support.

TRANS17: The host data set name is incorrect or missing. File transfer was canceled.

The host data set name or file name specified was not valid or did not exist. Check the correct name and retry the command.

TRANS18: An incorrect option was **specified**. File transfer was canceled.

You specified an incorrect option on the file transfer command. Check the correct syntax of the command for the appropriate host type, and retry the command.

TRANS19: An error occurred while handling the host file. File transfer was canceled.

This indicates a problem at the host. Retry the command; if it continues to fail, contact the host personnel.

TRANS22: The host session ID *sessid* is not valid. File transfer was canceled.

The session identifier you specified on the file transfer command did not match the long name or session ID of any of your 3270 sessions. Check the correct session identifier and retry.

TRANS25: The keyboard is inhibited for the host session. File transfer was canceled.

The specified session cannot be used for file transfer because the keyboard is locked. This may be due to an error, such as typing invalid characters (clear it by pressing the RESET key) or the session maybe waiting for a host response to a 3270 command. Retry the command when the keyboard has been unlocked.

TRANS26: A critical system error occurred. File transfer was canceled.

An error occurred on the UNIX computer, such as termination of the 3270 emulation program. Check for error conditions, correct, and retry.

Different hosts use the message number TRANS27 to indicate different errors; the following three messages are all returned as TRANS27 by different hosts.

TRANS27: File transfer was canceled when maximum timeout was reached.

The host did not respond within the timeout value you specified on the file transfer command. Retry, using a longer timeout or specifying no timeout.

TRANS27: The communication sequence with the host was disrupted. File transfer was canceled.

Retry the command. If it continues to fail, contact host personnel.

TRANS27: File transfer was canceled at the users request.

You canceled the command-line file transfer by choosing Terminate Transfer from the menu interface.

TRANS28: Option **optionname** not valid. File transfer was canceled.

You specified an option on the file transfer command that the host did not accept. Correct the syntax and retry the command.

TRANS29: Option **optionname** not valid with RECEIVE. File transfer was canceled.

You specified an option on the RECEIVE file transfer command that was valid only for the SEND command. Correct the syntax and retry the command.

TRANS30: Option **optionname** not valid with APPEND. File transfer was canceled.

You specified an option that was not valid when the APPEND option was also specified. Correct the syntax and retry the command.

TRANS31 : Option **optionname** cannot be used without the SPACE option. File transfer was canceled.

You specified an option that is valid only when the SPACE option also is specified. Correct the syntax and retry the command.

TRANS32: Option **optionname** not valid with a partitioned data set. File transfer was canceled.

You specified an option that was not valid with the host data set being used. Correct the syntax and retry the command.

File Transfer Messages

TRANS33: Only one of these option (TRACKS, CYLINDERS, AVBLOCK) is allowed. File transfer was canceled.

You specified more than one of the options shown; only one can be used. Connect syntax and retry the command.

TRANS34: The specified VM/CMS file was not found. File transfer was canceled.

The host file you specified could not be found. Retry the command using the correct name of the file.

TRANS35: The specified VM/CMS disk is read-only. File transfer was canceled.

You attempted to send a file to a host disk for which you do not have write access. Check that you are accessing the correct disk, or contact host personnel if you need to change access permissions.

TRANS36: The specified VM/CMS disk cannot be accessed. File transfer was canceled.

You attempted to access a host disk that is not available, or for which you do not have access permission. Check that you are accessing the correct disk, or contact host personnel to change access permissions.

TRANS37: The specified VM/CMS disk is full. File transfer was canceled.

You attempted to send a file to a host disk on which there was insufficient disk space for the file transfer. Contact host personnel.

TRANS38: You have started a file transfer in screen mode. Please abort it.

The LU used by the 3270 session on which you are attempting to transfer a file is configured as CUT mode (screen mode) rather than Write Structured Field (WSF) mode (also called D= mode) at the host. The file transfer programs *comsend* and *comreceive* can only perform WSF file transfers. Abort the file transfer command (or disable the session from the menu interface) and contact your system administrator or host personnel. Do not attempt any more file transfers on this session until the problem has been corrected.

TRANS039 Either specify both code pages or neither when the emulator is loaded with the -t option.

You started the 3270 emulation program with the -t (user-defined translation table) option. This means that ASCII-EBCDIC translation on a command-line file transfer

normally uses the user-defined translation table. If you want to override this, you must specify both the host code page H(nnn) and the local code page P(nnn); you cannot specify one without the other.

Retry the file transfer command, specifying both code pages or neither of them. Alternatively, restart the 3270 emulation program without the -t option and retry the file transfer command.

TRANS098: File transfer to non-standard host program complete **successfully.**

The host file transfer program generated an INWnnnn message (instead of a TRANSnnn message) indicating that the file transfer completed successfully.

TRANS099: File transfer to non-standard host program failed. Retry from the menu interface, or check trace file.

The host file transfer program generated an INWnnnn message (instead of a TRANSnnn message) indicating that the file transfer failed. LINKix 3270 command-line file transfer does not distinguish between INW messages. To obtain more information about the failure, retry the file transfer from the 3270 menu interface (which will display the INW message number and text), or check the message trace for either the local node or the 3270 emulation program.

TRANS108: The specified host code page is not valid. File transfer was canceled.

The host (EBCDIC) code page you specified on the file transfer command was invalid. See chapter 4 for valid code pages, and retry the command.

TRANS109: The specified local code page is not valid. File transfer was canceled.

The local (ASCII) code page specified on the file transfer command was invalid. See chapter 4 for valid code pages, and retry the command.

TRANS110: The specified emulator session is already in use. File transfer was canceled.

The session you specified on the file transfer command is already in use by another file transfer process or by a HLLAPI application. Retry the file transfer when the current process has finished, or use another 3270 session.

File Transfer Messages

TRANS111: An error occurred while accessing the local file. File transfer was canceled.

This error may be caused by a faulty disk or diskette, or by not closing the diskette drive door properly, or the specified UNIX file name may be invalid. Check for these conditions and retry the command.

TRANS112: The specified local drive is locked. File transfer is canceled.

Unlock the drive and retry the command.

TRANS113: The local disk or diskette is full. File transfer is canceled. Disk or diskette space on the UNIX computer is insufficient for the new file. Clear some space by deleting unnecessary files and then retry; or use another diskette.

TRANS114: Cannot read or write to the specified local disk or diskette.

A data error on the disk or diskette may have caused this error. Retry the operation.

TRANS115: The maximum number of files for the specified directory has been reached. File transfer was canceled.

The new file cannot be created because the directory already contains its maximum number of files. Delete any unnecessary files from the directory and retry; or transfer the file to another directory.

TRANS116: The maximum number of local files are open. File transfer was canceled.

File transfer cannot take place because the maximum number of open files (either the overall limit for the UNIX computer, or the limit for an individual process) has been reached. Close any unnecessary files used by other processes, or wait for any other file transfers currently in progress to complete; then retry the command.

TRANS117: The specified local file is already in use. File transfer was canceled.

The file specified has already been opened by another process that prevents you from accessing it. Stop the other process and retry the command. (This message might also be displayed if the file name you specified was the name of a directory.)

TRANS118: The local disk or diskette is write protected. File transfer was canceled.

Check that you are accessing the correct disk or diskette, or remove the write-protect tab from the diskette. Then retry the operation.

File Transfer Messages

TRANS119: The local diskette drive is not ready. File transfer was canceled.

Check that you are accessing the correct diskette drive, or that the diskette is inserted properly and the drive door is closed. Then retry the operation.

TRANS124: Message below was **received** from the host: *messagetext*
Refer to your host personnel for information on the host message shown.

TRANS999: The 3270 emulator has not been loaded. File transfer was canceled.

You tried to use a file transfer command when the 3270 emulation program had not been loaded, or you did not specify a session identifier on the command and no 3270 session IDs were configured.

File Transfer Messages

E. Glossary

3270	IBM's information display system ò a family of products composed of displays, printers and controllers that act as terminals to IBM systems. <i>See also</i> display emulation; emulation; printer emulation.
3270 emulation program	The LINKix 3270 program which allows a UNIX computer to emulate an IBM 3270 terminal and allows you to control the appearance and function of this emulation.
accelerator key	A key that corresponds to a letter in the name of a menu option or dialog item that can be pressed to select the option or item.
active session	The status of an LU-to-LU session where the primary and secondary LUs have agreed to the bind request session parameters. <i>See also</i> disabled; enabled
ACTLU	Activate Logical Unit request. In SNA, a request issued by an SSCP to an LU to initiate an SSCP-to-LU session. <i>Contrast with</i> DACTLU .
ACTPU	Activate Physical Unit request. In SNA, a request issued by an SSCP to a PU to initiate an SSCP-to-PU session. <i>Contrast with</i> DACTPU .
AE-string	A character string that contains uppercase A-Z, lowercase a-z, 0-9, . (period), #, \$ or @.
AW key	Attention identification key. A 3270 control key, such as PF1 , CLEAR or ENTER , that causes data to be sent from a device to the mainframe.
alert	<i>See</i> NetView alert; NetView user alert.
allocate	In APPC, to request the establishment of a conversation between two APPC TPs. One TP requests a conversation with another TP by issuing the APPC allocate verb in which the other TP is named.
API	Application Program (or programming) Interface. A programming language interface that allows a user-written

Glossary

	application program to access the functions of a hardware or software system.
APPC	Advanced Program-to-Program Communications. An application program interface used to access LU 6.2 services, allowing peer-to-peer communication between TPs in a network.
APPC verb	An APPC programming command.
append	To add data to the end of an existing file. <i>Contrast with overwrite.</i>
application TP	<i>See TP.</i>
ASCII	American Standard Code for Information Interchange. An encoding method that uses 7 bits to represent standard graphical characters. The PC code pages are based on ASCII, but they use the remaining 128 bit patterns to represent additional extended graphical characters. <i>See also PC ASCII. Contrast with EBCDIC.</i>
A-string	A character string that contains uppercase A-Z, 0-9, #, \$ or @.
attribute byte	On a 3270 display, a byte that specifies either the characteristics of a field (such as numeric or protected) or the properties of a character or field (such as blinking or reverse video).
audit log	A log file in which a record of normal system events is kept. <i>Contrast with error log.</i>
auto-skip	A characteristic of a 3270 protected numeric field that causes the cursor to skip automatically to the next unprotected field when the current field is completed.
base terminal attributes	Basic features that a 3270 display terminal provides, including protected and unprotected fields.
basic conversation	A type of APPC conversation generally used by low-level service TPs. <i>See also mapped conversation.</i>

basic conversation verb	A type of verb used by APPC TPs in basic conversations.
basic information unit	A unit of SNA information used to exchange data between two network-addressable units. <i>See also request header; request unit; response header response unit.</i> Abbreviated to BIU.
BIND	In SNA, a request to activate a session between two LUs. <i>Contrast with UNBIND.</i>
Browse program	The LINKix program that lets you print or view data from a trace or log file.
CCITT	Comite Consultatif International Telephonique et Telegraphique (International Telephone and Telegraph Consultative Committee). Among other things, this international standards body defines the X.25 protocol and sets standards for data transmission over long distances.
CCS	Common communications support. A major element of SAA. CCS specifies guidelines for communication between different computers, systems and programs.
check box	A pair of square brackets (box) followed by a description, that permits you to turn a selection on or off. An X appears in the brackets when the selection is turned on.
CICS or CICS/VS	Customer Information Control System or Customer Information Control System/Virtual Storage. An IBM transaction processing subsystem that provides an environment on IBM mainframes in which applications can communicate with terminals or other applications. LINKix 3270 supports communication with CICS/VS, via both 3270 emulation and APPC.
CMS	Conversational Monitor System. An operating environment available under VM, which provides time-sharing and program-development facilities to terminal users. Also called VMICMS.
code page	A table that defines the mapping between graphical characters and 8-bit numbers in the range 0-255. Code pages are typically based on EBCDIC or ASCII and vary slightly from country to

Glossary

	country.
color mapping	The action of changing the foreground, background and highlight colors used on a display and status line during a 3270 session. Also called color remapping.
command line	The line on a computer display where commands appear as they are typed in. The command line provides access to a shell such as the C shell or Bourne shell. Depending on the computer and the software that's running, the command line may contain a prompt such as the UNIX % or \$ prompt.
Common Service Verbs (CSV)	A CLEO API that provides facilities for translating characters, for tracing and for sending network management information to a host.
communications controller	A control unit that manages line control and the routing of data through a network. It is controlled by a program or programs stored and executed in the device.
conjunction file	The file containing parameters that define the network. The parameters define connections, the local node, LUs, users and so forth. Configuration files always have the extension .cfg. See also security file .
configuration program	A program used to configure the software or to create a file to contain run-time system parameters.
configure	To setup values that define a system.
confirmation processing	A process through which communicating APPC TPs synchronize their processing by confirming to each other that data has been received.
connection	The data communication path to another node in the SNA network.

contention loser	In an APPC LU-to-LU session, the LU that does not have the right to start a conversation with its partner (the contention winner) without first asking permission of the partner LU. <i>See also</i> contention winner .
contention winner	In an APPC LU-to-LU session, the LU that has the right to start a conversation with its partner (the contention loser). If parallel sessions between the two LUs are being used, each LU may be the contention winner for some sessions and the contention loser for other sessions. <i>See also</i> contention loser .
control daemon	The daemon program which controls the kernel components (local node and link services), and also controls the error log daemon and network management daemon .
control interface	The menu interface of the 3270 emulation program. It allows you to control the appearance and functions of 3270 emulation.
Control menu	The pull-down menu in the 3270 control interface which is used to control display and printer sessions, transfer files between the LINKix 3270 computer and a host, view host response times and send NetView alert messages.
conversation	A complete transaction between two APPC TPs starting with an allocate verb and ending with a deallocate verb.
conversation state	The current condition of an APPC conversation, which indicates allowable verbs at that point in the conversation.
CSV	<i>See</i> Common Service Verbs .
CUA	Common User Access. A set of SAA elements that define how screen layouts, menus, selection techniques, keyboard layouts and display options should look and function. CUA helps application developers create programs that can be used on different computers in different environments with little or no retraining of users.
cursor control keys	Keys used to move the cursor up, down, left and right on your terminal's display. They are LEFT , RIGHT , UP , DOWN , TAB and HOME . Also called cursor keys.

Glossary

customization	The process of altering the appearance of 3270 emulation. The 3270 control interface allows you to customize display and printer sessions, character sets, the representation of 3270 keys on your terminal, language support and hardware setup.
Customize menu	The pull-down menu in the 3270 control interface which is used to alter the appearance and performance of 3270 emulation.
DACTLU	Deactivate Logical Unit request. In SNA, a request issued by an SSCP to an LU to terminate an SSCP-to-LU session. <i>Contrast with ACTLU.</i>
DACTPU	Deactivate Physical Unit request. In SNA, a request issued by an SSCP to a PU to terminate an SSCP-to-PU session. <i>Contrast with ACTPU.</i>
daemon	A UNIX user-space process, not attached to any terminal or user ID, which manages the interaction between kernel-space processes and user-space processes. See also control daemon , diagnostics daemon , network management daemon .
data set	A collection of data elements on a tape or disk or in random access memory (RAM) or read-only memory (ROM). Generally called a file or record; the term data set is used mostly in MVS .
deallocate	To terminate a conversation between two APPC TPs.
default	A value that is automatically used if one is not specified.
default LU pool	A pool of ARC LUs configured to be used as defaults by any local TP in which no particular local LU is specified for the allocate request.
dependent APPC LU	An LU 6.2 whose address must be defined, both locally and at VTAM on the host node, before communications can take place. Dependent LUs can communicate only with hosts, not with peers, and are required to communicate with some versions of VTAM. <i>Contrast with independent APPC LU.</i>

DES	Data Encryption Standard. An encryption method used during security validation for an LU-to-LU session.
diagnostics daemon	The daemon which logs error log and audit log messages and also controls tracing.
dialog	A window that contains prompts and fields for entering information and for activating and deactivating features.
dialog item	A box, button or edit field appearing in a dialog. Dialog items consist of check boxes, radio buttons, edit fields, list boxes, tracking fields and pushbuttons.
disabled	The state in which a 3270 emulation session is not associated with an LU.
display emulation	The feature that allows a UNIX terminal to emulate an IBM terminal such as 3278 or 3279. This feature supports display Models 2 through 5, EAB, color mapping, keyboard remapping, multiple sessions and 3270 status messages. <i>See also</i> emulation .
display model	A number between 2 and 5 that identifies a type of 3278 or 3279 video adapter and monitor. Model 2 is the standard 24-row-by-80-column display. Models 3 through 5 are used for larger displays.
display session	A 3270 emulation session between a local computer and a host. It is an LU type 2 session and is used to emulate a 3278 or 3279 display. Also called host display session.
DLC	Data link control. In SNA, the protocol stack layer that transmits messages across links and manages link-level flow and error recovery.
dual list boxes	Two list boxes that work together. One of the list boxes provides a list of choices; when a choice is selected it appears in the other list box. <i>See also</i> list box .
EAB	Extended attribute byte. A byte whose bits define the extended attributes of a field. <i>See also</i> ECA ; attribute byte ; EFA .

Glossary

EBCDIC	Extended binary coded decimal interchange code. An 8-bit code in which all 8 bits are used for data and control word combinations. EBCDIC is used in IBM and IBM-compatible mainframe computer systems. <i>Contrast with ASCII.</i>
ECA	Extended character attribute. An EAB that defines highlighting, color and the character set for individual characters. <i>See also EAB; EFA; extended attributes.</i>
edit field	In a dialog, a field that's filled by typing data directly into it, as opposed to a field that can be filled only by selecting an option from a corresponding list of choices.
edit keys	Keys used to modify edit fields in a dialog, for example, DELETE .
EFA	Extended field attribute. An EAB that defines highlighting, color and the character set for a field unless overridden by an ECA. <i>See also EAB; ECA.</i>
EHLLAPI	Emulator High-Level Language Application Programming Interface. IBM's implementation of HLLAPI on OS/2. CLEO's implementation of HLLAPI is compatible with EHLLAPI version 1.0.
emulation	The process whereby one device imitates another for example, a UNIX terminal can emulate a 3278 terminal. <i>See also display emulation; printer emulation; 3270 emulation program.</i>
emulation program	<i>See 3270 emulation program.</i>
enabled	The state in which a 3270 emulation session is associated with an LU. If the host connection is active, the session may be used.
EOF	End-of-file. (1) A character which appears as the last character of a text file and indicates the end of the file. (2) A key sequence used on a UNIX terminal (typically CTRL + D), used to signal the end of a sequence of operations.

error log	A log file in which warning messages indicating problems with software operation, and error messages that indicate the reasons for a failure. <i>Contrast with</i> audit log .
extended attributes	Visual features provided by a terminal including blinking, reverse video, underline and seven-color (on a color monitor) character display.
FEP	Front End Processor. A communication control unit for a mainframe computer for example an IBM 3745. The FEP processes communications for the mainframe, reducing the processing demands of the host CPU.
File menu	The pull-down menu on the main screen of some menu interface programs, used to open files, to save files and to exit the program.
file transfer	The process of transmitting data files between computers.
flush	An APPC conversation verb that causes queued data from previous verbs to be sent to the partner TP.
formatted screen	Formatted screens have one or more fields that contain specific types of data. <i>Contrast with</i> unformatted screen .
full-duplex	The ability (of a modem, communications link and so on) to transmit and receive data at the same time. <i>See also</i> half-duplex .
grayed item	In a dialog, an item which is distinguished to indicate either that no entry is required for the item or that none is allowed. On a color screen, grayed items appear gray instead of black; on a monochrome screen, grayed items have no highlighted character shown in their name (this character indicates the accelerator key which can be used to select the item). The only exception to this is the tracking text box which is always grayed, indicating that no entry can be typed directly into the box. Instead you change the contents of a tracking text box by selecting an entry from the associated list box.

Glossary

half-duplex	The ability to transmit data in only one direction at a time; that is, when the program or computer on one end of an exchange is sending, the program or computer on the other end must wait for the transmission to end before responding. <i>See also</i> full-duplex .
handshaking	The process in which two devices or programs exchange control messages to synchronize their states.
HLLAPI	High-Level Language Application Programming Interface. A programming interface that allows you to develop and run programmer-operator application programs on UNIX computers that communicate with IBM mainframes using 3270 emulation- <i>See also</i> EHLLAPI .
host-addressable printer	A logical printer that is defined as a device associated with an LU, configured as LU type 1 or 3, and can therefore support local copy printing and host printer session printing.
host character set	The language variant of EBCDIC (for example, French or German) specified for a host computer .
host computer	The main large computer participating in an SNA network; for example S/370, 308x, 309x. Also called a mainframe.
host display session	<i>See</i> display session .
host file transfer parameters	In 3270 file transfers, parameters that specify the characteristics of a file being transferred; for example, the format and size of the records in the file.
host printer session	<i>See</i> printer session .
host response time	The amount of time a host computer takes to reply to a message sent to it by the 3270 emulation program. <i>See also</i> Response Time Monitor , LTTI .

hot key	A keystroke or combination of keystrokes that transfers control from one part of an application to another or from one application to another when both applications are running at the same time. LINKix 3270 has hot keys that (for example) allow a user to move from a 3270 emulation session to a specific point in the control interface or can start a sub-shell while the 3270 emulation program is still running.
implicit mode	A mode entry in the configuration which specifies that communication between a pair of APPC LUs can occur when the mode name specified by the invoking TP does not match a mode name configured for the LU pair. The LU being used by the invoked TP must have an implicit mode configured for it.
implicit partner LU	Remote LUs that do not have specific entries in the configuration may initiate sessions with local LUs.
IMS	Information Management System. An IBM database and data-communications-management system that runs on an IBM mainframe.
incoming allocate	An APPC request to start an APPC conversation. It is generated by the invoking TP through the [MC_] ALLOCATE verb and accepted by the invoked TP through the RECEIVE_ALLOCATE verb.
independent APPC LU	An LU 6.2 whose address need not be defined before communications can take place. Independent LUs can communicate either with hosts or with peer computers. <i>Contrast with dependent APPC LU.</i>
IND\$FILE	IBM's file transfer program which allows files to be transferred between a local computer and the host. It operates in three host environments: CICS, VM/CMS and MVS/TSO.
insert mode	The state of a computer keyboard when typed characters are inserted at the cursor location without overwriting other characters. Characters at the cursor position and to the right of it are shifted to the right as new characters are inserted.

Glossary

invokable TP	An invokable APPC TP is one that is started (either by an operator or automatically) in response to a request from another TP (the invoking 'IT).
invoked TP	A TP started by another (invoking) TP.
invoking TP	A TP that starts another TP. The invoking TP starts the other TP by instructing the other TP's node to load the invokable TP.
keyboard mapping	The act of defining keys on a UNIX terminal as host 3270 keys, to allow the host to recognize keystrokes entered at the UNIX terminal. Also called keyboard remapping.
LAN	Local Area Network. A high speed communication system consisting of hardware (computers and peripherals) and software (programs and data files) that are interconnected by cable in a way that allows these resources to be shared. The connected devices are located within a limited geographic area such as a building or campus. <i>Contrast with</i> WAN .
least significant bit	The lowest bit of an 8-bit binary number, corresponding to 2 ⁰ (2 to the power zero).
link service	The software component that communicates with the device driver for a particular communication adapter (for example SDLC or X.25).
list box	In a dialog, a box that contains a list of choices.
local copy	The process of printing the contents of a 3270 display on a printer or into a disk file. Also called local copy printing.
local LU	In an APPC conversation, the LU on the local end of the session. <i>Contrast with</i> partner LU ; remote LU . <i>See also</i> local TP .
local node	The software component that implements an SNA PU 2.1 and provides SNA services.
local printer	A printer attached directly to a UNIX computer, possibly using a print spooler.

local printing	See local copy .
local TP	In an APPC conversation, the TP on the local end of the conversation. <i>Contrast with</i> remote TP . See also local LU .
log file	A file used to store messages generated by the software, which report (for example) software, hardware and link service failures; damaged or missing files; connection statistics; configuration problems; and system status. This file can be viewed. See also audit log , error log .
logical record	A group of data items that have some logical relationship to one another; for example, an inventory part record that contains all the data about the part.
logical unit	See LU .
logon	The process by which a user initiates contact with a computer system and identifies himself or herself as someone authorized to use the system.
long name	A unique identifier for a 3270 session, consisting of up to eight characters.
LTTI	Last transaction time indicator. An indicator, displayed on the 3270 status line that shows the time it took for the host to respond the last time an AID key was pressed. See also RTM , host response time .
LU	Logical unit. A logical entity that is the point of contact between a user and a network. See also LU type .
LU name	The unique name that identifies an LU.
LU-to-LU session	A logical, two-way exchange between two LUs over a specific connection for a specific amount of time.
LU type	Logical unit session type. A subset of the SNA protocol that characterizes the communication between two LUs during an LU-to-LU session. Types 1,2,3 and 6.2 are supported.

Glossary

LU type 1	A protocol used by a host application program that is communicating with a printer, sending data that conforms to the SCS definition. Also called LU1 printing or SCS printing.
LU type 2	A protocol used by a host application program communicating with a 3270-type display terminal, using the SNA 3270 data stream.
LU type 3	A protocol used by a host application program communicating with a printer, using the SNA 3270 data stream. Also called LU3 printing or DSC printing.
LU type 6.2	A protocol used by two TPs communicating as peers. LU 6.2 works in combination with node type 2.1 to provide APPC communications. <i>See also</i> dependent LU ; independent LU .
Manage program	The program that allows administrators to manage link services, sessions, APPC TPs and so forth.
mapped conversation	A type of APPC conversation generally used by application TPs. <i>See also</i> basic conversation .
mapped conversation verb	A type of verb used by APPC TPs in mapped conversations.
mapping	In 3270 emulation, the process of translating a set of host parameters into a corresponding set of local parameters. For example, keyboard mapping specifies the UNIX terminal keystrokes used to represent 3270 keys.
menu	A list of choices, shown in the user interface of a program. Each choice leads to different functions within the program.
menu bar	The top line of the main screen of programs such as the Configuration program and the 3270 emulation program's control interface. This lists the actions available from the main screen. When an action is selected, a pull-down menu appears.
message file	(1) The file <i>com.msg</i> , which contains the messages the software writes to audit log or error log files. (2) A similar file which you can create in order to log your own messages using the CSV interface.

mode	A specification for the network properties that govern an LU-to-LU session.
model	See display model .
modem	modulator/demodulator. A device that converts digital data into an analog signal that can be transmitted over communication lines and that reconverts received analog signals into digital data.
mode table	The file on the host that contains the specifications for all possible modes.
most significant bit	The highest bit of an 8-bit binary number, corresponding to 2^7 (2 to the power 7).
multidrop	A connection in which one primary node communicates with multiple secondary nodes over the same physical transmission medium.
MVS	Multiple Virtual Storage. An IBM System 370 operating system that supports multiple programming.
MVS/130	An IBM host time-sharing environment.
NAU	Network addressable unit. An LU , PU , PNCP or SSCP . Each NAU has an address that identifies it to the network and allows the network to route information to and from other NAUs.
NCP	Network control program. An IBM program that runs on a host FEP and supports communication controllers in single-domain, multiple-domain and interconnected networks.
NetView	An IBM monitoring and reporting system that runs on a host computer and gathers data for administrative tasks such as operations management, problem determination and performance management.
NetView alert	A message to NetView indicating an abnormal event or a failure.
NetView user alert	A message sent by a 3270 emulation user to NetView requesting an operator action such as mounting a tape or

Glossary

	changing forms on a printer. Also called user alert.
network	A collection of computer systems, controllers, terminals and software connected in a way that allows them to communicate with each other.
network management daemon	The daemon which controls interaction with the host NetView program.
NMVT	Network Management Vector Transport. A message sent between a LINKix product and the NetView program, used (for example) to send alert data or to request or send response time data.
node	A device that contains hardware, software and/or microcode to implement SNA functions. SNA defines three kinds of nodes: the host subarea node, which functions to control and manage a network; the communication controller subarea node, which routes and controls data flow through the network; and peripheral nodes, which include printers, workstations, cluster controllers and distributed processors. The local node is a peripheral node. <i>See also</i> PU .
node type 2.1	An SNA component, such as an intelligent terminal or a PC, that works together with LU type 6.2 to support peer-to-peer communications, allowing the LUs to function independently from the host.
non-queued TP	An invokable TP that is loaded automatically every time an incoming allocate command from an invoking TP arrives at the local LU. A non-queued TP can be started more than once by multiple incoming allocate commands, even if the program is already running. <i>Contrast with</i> queued TP .
NRZ/NRZI	Non-return-t-zero / non-return-to-zero-inverted. Two methods for encoding data to be sent along a communications link. The local and remote ends of the link must both use the same encoding method.
operator information area	See status line .

(OIA)

operator-loaded TP	An invokable TP that is manually loaded and started by an operator.
overwrite	To replace the contents of a file with new data by writing the new data over the existing contents. <i>Contrast with</i> append .
pacing	A technique in SNA for implementing flow control on RUs sent on a session. After sending a configurable number of RUs (the pacing count), the sender must wait for a pacing response (indicating that more data can be received). This ensures that the sender cannot send more data than its partner is able to accept.
pacing count	See <i>pacing</i> .
parallel sessions	Multiple concurrent sessions between a pair of APPC LUs, allowing two TPs to perform multiple (different) operations simultaneously.
partner LUs	Two LUs that are configured to communicate with each other. Partner LUs can reside on the same or different nodes.
partner TPs	Two TPs, residing on the same or separate nodes, that are configured to communicate with each other. Partner TPs use partner LUs.
path	(1) In SNA, the series of nodes and communication links over which data must travel from one LU to another. (2) The sequence of directories from a root segment to the directory where a file resides.
PC ASCII	A code set that uses standard ASCII as a base, but which uses a full 8 bits to represent a character. PC ASCII is commonly called PC code page 437.

Glossary

peer computer	A mainframe, mini or PC that communicates with another computer as an equal partner, with both computers sharing control over the communication. <i>See also</i> peer-to-peer , peer node .
peer node	A node which communicates with another node as an equal partner, rather than a host node which is always in control of the communications process. <i>See also</i> peer computer , peer-to-peer .
peer-to-peer	A kind of communication in which two systems communicate as equal partners sharing the processing and control of the exchange, as opposed to host-terminal communication in which the host does most of the processing and controls the exchange. <i>See also</i> peer computer , peer node .
PIP	Program Initialization Parameters. In APPC , parameters and environment setup information sent to an invokable TP during initialization of the TP.
PLU	Primary Logical Unit. In SNA, the logical unit that sends the session activation request and BIND for an LU-to-LU session. <i>Contrast with</i> SLU .
PNCP	Peripheral node control point. Similar to SSCP , but found in a node type 2.1 (such as the local node); controls the resources of the local node.
pop-up message	A message automatically generated by user interface programs, that notifies the operator of an error or to request information from the operator. It appears in a box on the screen and requires the operator to press a key to continue.
presentation space	An area of computer memory (used by HLLAPI, the presentation layer of the 3270 emulation program and the read editor within the emulator) that holds 3270 screen data and attributes to be displayed.
print spooler	Most UNIX systems do not allow direct access to a printer; you access it by means of a print spooler (for example lp).

printer emulation	The ability to emulate an IBM 3287 printer to print host data or to do local copy printing.
printer session	A 3270 emulation session between LINKix 3270 and a host, used to print to a local printer that is emulating a 3287 printer (or to print to a file).
process	A single sequence of operations executed by the UNIX computer; a process exists for each operating system command being executed at a given time. Onerogram may contain more than one process, or it may issue an operating system call which starts a new process.
protected field	A field on a 3270 screen or in a presentation space into which no data can be entered by the operator. The field is restricted by the value in its attribute byte.
PS	See presentation space .
PS identifier	Presentation space identifier- A one-character name, AÆZ, that identifies a presentation space. In LINKix 3270 this corresponds to the 3270 session ID.
PU	Physical unit. An entity that provides the services needed to use and manage a particular type of device and the physical resources associated with the device, such as communications links. A PU is implemented with a combination of hardware, software and microcode. <i>See also node.</i>
PU 2.1	<i>See node type 2.1.</i>
pull-down menu	A menu accessible from the menu bar at the top of the main screen of a menu interface program.
pushbutton	In the menu interface, a choice displayed in angle brackets (<>) at the bottom of a dialog box. When selected, a pushbutton executes an immediate action such as Cancel.
QLLC	Qualified Logical Link Control. The protocol used to permit SNA sessions to run over X.25 networks.

Glossary

queued TP	An invokable TP that can be started by only one incoming allocate command at a time. Incoming allocate commands that arrive while the queued TP is running do not start the program again, but are queued until the program issues another RECEIVE_ALLOCATE or until it finishes execution. <i>Contrast with non-queued TP.</i>
radio button	In a dialog, a selection button that allows you to choose one item from a group of mutually exclusive options. <i>See also radio group. Contrast with check box.</i>
radio group	In a dialog, a set of related radio buttons. When a button within the group is selected, any other selected button in the group is automatically turned off.
remap	In 3270 emulation: (1) to change the colors displayed on a terminal's screen (color remapping); (2) to change the functions of keys or key combinations on a terminal's keyboard (keyboard remapping); (3) to change the LU or LU pool currently associated with a session (session remapping).
remote LU	From the viewpoint of a local LU in an LU-to-LU session, the LU on the other end of the session. <i>See also remote TP.</i>
remote node	(1) The node at the other end of a connection. (2) The node that contains the LU at the other end of a session. (3) The node that contains the TP at the other end of a conversation.
remote TP	From the viewpoint of a TP conversing through an LU-to-LU session, the TP on the other end of the conversation. In order for two conversing TPs to be considered remote, they must be on different nodes. <i>See also remote LU.</i>
request header (RH)	In a basic information unit, a 3-byte field that identifies the type and format of the data contained in the associated request unit.

request unit (RU)	A unit of information, contained in a basic information unit, that contains either end-user data (data request unit) or an SNA command (command request unit).
response header (RH)	In a basic information unit, a 3-byte field that precedes a response unit, identifying the type of data in the response unit. A response header always precedes a response unit.
Response Time Monitor (RTM)	A 3270 and NetView facility that monitors the amount of time it takes for a host to respond during 3270 display sessions. <i>See also host response time, LTTI</i>
response unit (RU)	A basic information unit that is sent in response to a request. The response is either positive (indicating that the request was acceptable) or negative (indicating that the request was unacceptable). If the request was unacceptable, the response unit also contains an indicator giving the reason why the request was unacceptable.
RH	Request header or response header .
RTM	See Response Time Monitor .
RU	Request unit or response unit.
SAA	Systems Application Architecture. A set of guidelines created by IBM to help developers standardize applications so they function in different operating environments with minimal program modification and retraining of users. LINKix software products comply with two of the major elements of SAA - Common User Access and Common Communications Support.
SC	Session Control. In SNA: (1) The component of transmission control that performs error processing after an unrecoverable error. (2) A request or response unit category used for activating and deactivating sessions.

Glossary

SCS	SNA Character String. In SNA, EBCDIC control codes within a request unit that format printer and display data set modes of device operation, define data usage, and communicate between device operators and application programs. Examples of SNA character string controls are Backspace, Form Feed, and Carriage Return. SCS controls do not include data flow control functions.
SDLC	Synchronous data link control. A type of link service used for managing synchronous data transfer.
security file	The file that contains the security keys and user ID/password pairs for LU-to-LU session-level security. The security file is associated with a configuration file, and it has the same name as the configuration file except for the extension .sec, instead of .cfg.
security key	An identifier used by two APPC LUS to validate security when a session is activated. The security key performs a function similar to that of a password, but at the LU-to-LU session level rather than at the TP-conversation level.
sense code	In SNA, the first 4 bytes of a response unit of a negative response, giving the reason for the response, plus an optional 3 bytes that identify the rejected request. Also called sense data.
service TP	A TP that uses basic conversation verbs and provides services to other TPs.
session	A logical, two-way exchange between two LUs over a specific connection for a specific amount of time. In APPC, an LU can have multiple sessions with another LU, or with several LUs, at the same time. In 3270 emulation, each LU has one session with another LU and one session with the SSCP.
session ID	A one-character name that identifies a 3270 emulation session. <i>See also</i> PS identifier .

session limit	The maximum number of parallel sessions that can be active between two APPC LUs. Session limits are configured for the mode used by the LUs and for the LUs themselves. When the LU-to-LU session is established, the session limit is negotiated between the two LUs based on the configured values.
severity level	A number that indicates the severity of an audit or error message. Audit messages provide information and have severity 6, 8 or 10. Error messages have severity 12 or 16, indicating a problem that needs resolution.
shell	A command interpreter which allows you to communicate with the UNIX operating system, for example the C shell or Bourne shell.
SLU	Secondary Logical Unit. In SNA, the LU that receives the session activation request (BIND) for an LU-to-LU session. The secondary LU may acceptor reject a nonnegotiable bind request, negotiate session parameters for a negotiable bind request and request session termination. <i>Contrast with PLU.</i>
SNA	Systems Network Architecture. A comprehensive set of rules and protocols formulated by IBM to govern the way communications networks operate and interact.
spooler	<i>See print spooler.</i>
SSCP	System services control point. A part of VTAM that controls and coordinates components of an SNA network, handling routing and directory services, problem determination requests and session services on behalf of network users.
SSCP-to-LU session	A session between an SSCP and an LU. The session permits the LU to request help from the SSCP in starting LU-to-LU sessions. For example, in 3270 emulation the logon request flows on the SSCP-to-LU session.
status line	The bottom line on a 3270 emulation display. It is reserved for system messages and information about the current 3270 session. HLLAPI refers to this as the Operator Information Area (OIA).

Glossary

structured field	A means of encoding variable-length data, or data not intended for 3270 display, for transmission in a 3270 data stream.
style file	The file that holds the details about a user's 3270 customization. Style files always have the extension .stu.
synchronous transmission	Transmission in which both the data characters and the individual bits are transmitted at a fixed rate, with the transmitter and receiver being synchronized. This eliminates the need for individual start and stop bits surrounding each byte. Both SDLC and X.25 use synchronous transmission.
temporary exit	A feature of the 3270 emulation program, which allows you to invoke a second copy of the shell while the program remains active (for example to list the contents of a directory or to run a HLLAPI application).
terminal	A device that can send or receive data over a data communications channel. The 3270 emulation program emulates 3278 and 3279 terminals.
terminfo	The database on a UNIX computer which holds details of the capabilities of different terminal types; for example, whether the terminal supports an alternate character set, and what keys are available on the terminal's keyboard.
IT	Transaction program; also called an application TP. Generally, an application program that uses APPC to exchange data with another TP. (Occasionally, HLLAPI application programs are also refereed to as TPs.)
trace file	A file in which tracing data is stored. Trace files normally have the extension .trc.
tracing	The action of tracking the activities of APIs, communication links and internal flows, including the calls made to APIs. Tracing stores a history of activity in trace files.
tracking field	In a dialog, a field into which data is entered by selecting a choice from a corresponding list box, as opposed to a field into which data is typed directly.

transaction	A set of data entered into a system (e.g., a deposit to a bank account), triggering a certain action (e.g., updating a balance).
transaction program	<i>See</i> TP.
T S O	Time Sharing Option. An IBM full-function time-sharing system for MVS that provides interactive computing for a large system environment.
UNBIND	In SNA, a request to deactivate a session between two LUs. <i>Contrast with</i> BIND .
unformatted screen	An unformatted screen has no individually defined fields. It is treated as one large field. <i>Contrast with</i> formatted screen .
unprotected field	A field on a 3270 screen or in a presentation space into which an operator can enter data.
user alert	<i>See</i> NetView user alert
VM	Virtual machine. A group of functions in a mainframe that allows the mainframe to operate as if it were several different mainframes instead of one. Also, the operating system that controls these functions.
VM/CMS	<i>See</i> CMS .
VTAM	Virtual Telecommunications Access Method. A set of IBM mainframe programs that control communications between application programs and terminals by providing SNA services such as managing communication sessions and APIs.
WAN	Wide-Area Network. A high speed communication system, consisting of hardware (computers and peripherals) and software (programs and files), that provides communication services and allows resources to be shared over a larger geographic area than that served by a local area network. <i>Contrast with</i> LAN .
X.25	A CCITT standard for the interface protocol used to access a packet-switching network.

Glossary

Index

3270

- 3270 emulation, 5-5, 5-28
 - attribute character, 2-19
 - automating startup, 2-27
 - communicating with the host, 2-13
 - cursor control, 2-15
 - data fields, 2-19
 - exiting, 2-14, 2-30, 3-21
 - features, xiii
 - HLLAPI applications, 2-26
 - keyboard use, 2-15
 - resuming after temporary exit, 2-13
 - running in background, 2-10
 - scripts, 2-using with, 2-27
 - starting, 2-7, 2-28, 2-29, 4-28
 - status line, 2-16
 - temporary exit from, 2-13
- 3270 key
 - ACTIONS, 2-2, 2-12, 2-14, 2-24, 2-28, 4-29, 4-30, 5-3, 6-4, 6-5, A-6
 - ALT CURS, A-5
 - ATTN, A-3
 - BACKSPACE, A-11
 - BACKTAB, A-11
 - CLEAR, A-3
 - CURS SEL, A-3
 - DELETE, 2-21, 2-24, A-12
 - DEV CNCL, A-4
 - DOWN, A-10
 - DUP, A-12
 - ENTER, 2-5, 2-20, A-3
 - ERASE EOF, A-12
 - ERASE INP, A-13
 - EXIT, 2-2, 2-14, 2-30, A-6
 - FILE XFR, 2-24, 4-3, 4-4, A-6
 - FMARK, A-12
 - HOME, A-1 1
 - INSERT, 2-17, A-12
 - KEYS, 5-28, 6-4, 6-5, A-6
 - LEFT, A-10
 - LEFT DB, A-10
 - NEW LINE, A-1 1
 - NEXTSESS, 2-2, 2-14, 2-18, 2-24, 2-30, 3-8, 4-4, A-8
 - NUM OVR, A-7
 - PA1, 2-20, A-3
 - PA2, A-3
 - PA3, A-3
 - PAUSE, 2-22, 2-23, 2-25, A-8
 - PRINT, A-4
 - QUIT, 2-21, 2-22, 2-23, 2-24, A-8
 - RECORD, 2-18, 2-20, 2-21, 2-22, 2-23, 2-24, A-7
 - REDRAW, 2-20, 2-21, A-6
 - REPLAY, 2-18, 2-20, 2-22, 2-23, A-8
 - RESET, 2-2, 2-5, 2-17, 2-25, A-5
 - RIGHT, A-10
 - RIGHT DB, A-11
 - SESS, 2-2, 2-24, 2-30, 3-8, A-9
 - STAT HELP, A-7
 - STAT TOG, 2-2, 2-16, 6-6, A-7
 - SYS REQ, A-5
 - TAB, A-1 1
 - TEMPEXIT, 2-2, 2-13, 2-14, 2-20, 2-21, 2-24, 2-26, 2-30, 3-15, 4-16, A-6
 - UP, A-10
- 3270 keys
 - default keyboard mapping, 5-33, A-1
 - redefining, 5-29
 - undefining, 5-32
- 3270 session
 - 3278/79 host display sessions, 2-2
 - 3287 host printer sessions, 2-2

Index

- automatic logon, 2-26
- customizing, 2-6
- enabled at initialization, 2-12
- enabling, 2-28
- move to (from main screen), 2-13, 3-3
- moving to main screen, 2-12, 3-2, 5-3
- see also Display session
- see also Printer session
- switching between, 2-18, 2-30

3270 system keys, A-5

3270 terminal emulation, 2-2

3287 printer emulation, 2-4

A

- Accelerator key, 1-19
 - direct accelerator key, 1-8, 2-15, 5-28
- ACTIONS (3270 key), 2-2, 2-12, 2-14, 2-24, 2-28, 4-29, 4-30, 5-3, 6-4, 6-5, A-6
- ALT CURS (3270 key), A-5
- Alternate character mode, 1-2
- ATTN (3270 key), A-3
- Audible signal, 5-35

B

- Background processing, 2-3, 2-10
 - accessing sessions, D-2
 - command-line option, 2-7
 - stopping, 2-11
- BACKSPACE (3270 key), A-11
- BACKTAB (3270 key), A-11
- Beep, 5-35

C

- c3270
 - changing LU names, 2-9
 - command-line options, 2-7

- disable the menu interface, 2-7
- exiting, 2-14
- international support, 2-7
- model type, 2-9
- remapping session IDs, 2-10
- running in background, 2-10
- stopping, 2-11
- style file, 2-8
- Check box, 1-12
 - choosing, 1-12
- CICS host
 - file transfer, 4-14
 - see also File transfer
- CLEAR (3270 key), A-3
- Color mapping, 5-18
- Communications check code, 2-16, 3-5, B-3, B-9
- comreceive, 4-17
- comsend, 4-17
- Control menu, 2-6, 3-3
- CURS SEL (3270 key), A-3
- Cursor addressability, 1-2
- Cursor control, 2-15
- Customize menu, 2-6, 5-5
- Customizing
 - 3270 session, 2-6
 - custom translation tables, 5-37
 - display colors, 2-6, 5-18
 - display sessions, 5-11
 - key definitions, 5-28
 - miscellaneous options, 5-35
 - printer sessions, 5-15

D

- Data fields, 2-19
- DELETE (3270 key), 2-21, 2-24, A-12
- DEV CNCL (3270 key), A-4
- Dialogs, 1-9

-
- Abort File Transfer, 4-26
 - Choose Colors for session, 5-19
 - Choose Host Language, 5-25
 - Choose Session for Color Remapping, 5-19
 - Choose Session to Customize, 5-11, 5-15
 - Control Display Sessions, 3-4
 - Control Printer Sessions, 3-11
 - Copy Session Colors, 5-22
 - Customize Display Parameters, 5-12
 - Customize Printer Parameters, 5-16
 - File Transfer, 4-3, 4-23
 - Open Style File, 5-6
 - Options for session, 4-5
 - Redefine function to, 5-30
 - Redefining Key, 5-29
 - Remap Display Session, 3-9
 - Remap Printer Session, 3-13
 - Save Style File, 5-7
 - Save Style File As, 5-10
 - Send NetView Alert 3-19
 - Session Is In Use, 3-8
 - Specify Miscellaneous Options, 5-35
 - Transfer CICS File on session, 4-14
 - Transfer TSO File on session, 4-10
 - Transfer VM/CMS File on session, 4-12
 - View and Define Key Definitions, 5-28, 5-32
 - View Response Times, 3-16
 - View Response Times for Session, 3-16
 - Direct accelerator key, 1-8
 - Disabled (grayed) fields, 6-7
 - Disabled option, 1-20
 - Display
 - colors
 - background, 5-20
 - background and foreground, 5-20
 - base colors, 5-20
 - customizing, 5-18
 - default text, 5-20
 - extended colors, 5-20
 - foreground, 5-21
 - Host highlight attribute, 5-20
 - intense colors, 5-20
 - intense text, 5-20
 - screen blink, 5-20
 - status line, 5-20
 - features
 - blinking, 2-19, D-2
 - color, 2-19, D-2
 - reverse video, 2-19, D-2
 - underlining, 2-19, D-2
 - session
 - suppressed at startup, 5-17
 - V P
 - mono or color, 5-35
 - Display devices
 - Model 2, 2-3
 - Model 3, 2-3
 - Model 4, 2-3
 - Model 5, 2-3
 - Display session
 - application name, 3-6
 - connection name, 3-5
 - connection type, 3-5
 - controlling, 3-4
 - current session, 3-6
 - customizing, 5-11
 - deactivating, 3-8
 - description, 5-12
 - disabled, 3-5
 - disabling, 3-8
 - enabled, 3-5
 - enabled at initialization, 5-13
 - enabling/disabling, 3-6, 3-7
 - host screens suppressed at startup, 5-13
 - local copy destination, 5-14
 - long name, 3-5, 5-12

Index

- LU / pool, 3-5
- making current, 3-6, 3-7
- maximum structured field size, 5-14
- model type, 5-13
- moving between, 3-8
- node name, 3-5
- printer or file name, 5-14
- remapping, 3-6, 3-9
- session ID, 5-12
- status line option, 5-13
- Display session, 2-3, 3-7
 - status, 3-5
 - Status (SSCP), 3-6
- DOWN (3270 key), A-10
- Dual list box, 1-18
 - Moving Items, 1-18
- DUP (3270 key), A-12
- Dup character, A-12

E

- Edit box, 1-14
 - associated with list boxes, 1-17
 - entering information, 1-15
- Edit keys, A-10
- Elevator bar, 1-16
- ENTER (3270 key), 2-5, 2-20, A-3
- Environment variable
 - HOME, 5-2
 - IND_FILE, 4-1
- ERASE EOF (3270 key), A-12
- ERASE INP (3270 key), A-13
- Error messages
 - file transfer, C-1
- Exit
 - 3270 emulation, 3-21
 - ending emulation, 2-14
 - temp exit, 2-4
- EXIT (3270 key), 2-2, 2-14, 2-30, A-6

- Exiting, 2-30
- Extended Attribute Byte (EAB), 2-19

F

- Fast path
 - 3270 emulation, 2-28
 - file transfer, 4-28
- Field Mark character, A-12
- File menu, 2-5, 5-4
- File transfer, 2-4, 3-15, 4-10, 4-16
 - add to existing file, 4-7, 4-19
 - ASCII/EBCDIC conversion, 4-6, 4-19
 - average block size, 4-8
 - background processing, 4-16
 - CICS host, 4-14
 - CICS filename, 4-14
 - command line syntax, 4-17
 - comments, 4-15
 - local file name, 4-14
 - command line options, 4-19
 - command line syntax, 4-17
 - comreceive*, 4-17
 - comsend*, 4-17
 - error messages, C-1
 - exiting, 4-27
 - fast path, 4-28
 - host block size, 4-8
 - host record format, 4-7, 4-20
 - host record length, 4-8
 - incremental units of space, 4-8
 - initial units of space, 4-8
 - international language support, 4-6, 4-19
 - logging onto host, 4-4
 - maximum structured field size, 4-3
 - messages, C-2, C-4, D-3
 - monitoring, 4-12, 4-23
 - MVS/TSO host, 4-10
 - command line syntax, 4-17
 - host dataset, 4-11

- local file name, 4-11
- member name, 4-11
- password, 4-11
- sample commands, 4-22
- setting host parameters, 4-5
- status, 4-3
- stopping, 4-26
- strip/add LF, 4-7, 4-20
- text vs. binary files, 4-9, 4-19
- units of space, 4-8
- using menus, 4-5
- using the command line, 4-16
- VM/CMS host, 4-12
 - command line syntax, 4-17
 - host filemode, 4-13
 - host filename, 4-13
 - host filetype, 4-13
 - local file name, 4-12
- write structured fields (WSF) support, 4-2

FILE XFR (3270 key), 2-24, 4-3, 4-4, A-6

FMARK (3270 key), A-12

Foreign language support
see International support

Function keys, 1-2

H

h3270, D-1

- limitations, D-9
- screen capture file, D-7
- starting, D-4
- status line, D-5

HLLAPI, 2-6, 2-26

HLLAPI terminal emulator
see *h3270*

HOME (3270 key), A-11

HOME environment variable, 5-2

Host file transfer program, 4-1

Host language, 5-5, 5-25

Host response times, 2-6, 3-16

Host systems

- CICS, 2-4
- MVS/TSO, 2-4
- VM/CMS, 2-4

I

IND\$FILE, 2-4, C-4, D-3

IND_FILE, 4-1

INSERT (3270 key), 2-17, A-12

Insert mode indicator, B-14

International language support, 2-3

- command-line option, 2-7
- customizing, 5-25
- host code page, 5-25
- languages supported, 5-26
- local code page, 5-25

K

Key definitions, A-1

Keyboard

- 3270 keys, 2-15
- default keyboard mapping, 1-3, A-1
- mapping 3270 functions, 2-6

KEYS (3270 key), 5-28, 6-4, 6-5, A-6

Keystroke recording and replay, 2-5, 2-20

- restrictions, 2-25
- saving key sequences to a style file, 2-24

L

Last Transaction Time indicator (LTTI), 2-17, B-3, B-14

LEFT (3270 key), A-10

LEFT DB (3270 key), A-10

Index

List box, 1-15
 and associated edit boxes, 1-17
 choosing an item, 1-16
LTTI, B-14
LU names
 changing, 2-9
LU type 1 printing, 2-4
LU type 3 printing, 2-4
M
Main screen, 2-12, 3-2
Menu bar, 1-5
 choosing a menu, 1-5
Menu interface, 2-5
Menus, 1-7, 2-5
 choosing an option, 1-8
Message boxes, 1-22
Miscellaneous system keys, A-6
Model display support, 2-3
Model type
 changing, 2-9
 changing at the command line, 2-9
 changing in style file, 5-13
MVS/TSO host
 file transfer, 4-10
 see also File transfer

N

NetView alert messages, 2-6
 sending, 3-18
NEW LINE (3270 key), A-11
NEXTSESS (3270 key), 2-2, 2-14,
 2-18, 2-24, 2-30, 3-8, 4-4, A-8
NUM OVR (3270 key), A-7

O

On-line help, xi, 1-2, 1-20

P

PA1 (3270 key), 2-20, A-3
PA2 (3270 key), A-3
PA3 (3270 key), A-3
PAUSE (3270 key), 2-22, 2-23, 2-25
PRINT (3270 key), A-4
Printer control keys, A-4
Printer emulation, 3-13
Printer session, 2-3, 3-13
 connection name, 3-12
 connection type, 3-12
 controlling, 3-11
 customizing, 5-15
 description, 5-16
 disabled, 3-12
 enabled nnn, 3-12
 enabled at initialization, 5-17
 enabling/disabling, 3-13
 host app name, 3-12
 long name, 3-11, 5-16
 LU, 3-11
 node name, 3-12
 output destination, 5-17
 printer or file name, 5-17
 printing, 3-12
 remapping, 3-13
 SSCP, 3-12
 status, 3-12
Printing, 3-13
 local
 host-initiated, 2-4
 setting destination, 5-17
 user-initiated, 2-4
 local copy
 setting destination, 5-14
 LU type 1, 2-4
 LU type 3, 2-4
 to a file, 2-4
Problems

- solving, 6-1
 - PROG nnn messages, B-16
 - program access keys, A-3
 - Program check code, B-7, B-16
 - Program function keys, A-2
 - Pushbutton, 1-10
 - choosing, 1-10
- ## Q
- QUIT (3270 key), 2-21, 2-22, 2-23, 2-24
- ## R
- Radio button, 1-13
 - choosing, 1-13
 - Radio groups, 1-13
 - Receive command, 4-22
 - RECORD (3270 key), 2-18, 2-20, 2-21, 2-22, 2-23, 2-24
 - Record / Replay ID messages, 2-17, B-4
 - REDRAW (3270 key), 2-20, 2-21, A-6
 - REPLAY (3270 key), 2-18, 2-20, 2-22, 2-23
 - RESET (3270 key), 2-2, 2-5, 2-17, 2-25, A-5
 - RIGHT (3270 key), A-10
 - RIGHT DB (3270 key), A-11
- ## S
- Screen
 - redrawing, 1-24
 - size, 1-2
 - Send command, 4-22
 - SESS (3270 key), 2-2, 2-24, 2-30, 3-8, A-9
 - Session IDs
 - remapping, 2-10
 - Session screen, 2-12
 - Session selection keys, A-9
 - STAT HELP (3270 key), A-7
 - STAT TOG (3270 key), 2-2, 2-16, 6-6, A-7, A-8
 - Status line, 2-12, 2-16, 5-18, B-1
 - stop*3270, 2-11
 - Style file, 2-8, 2-13, 2-27, 5-2, 6-4
 - creating, 5-9
 - extension, 5-10
 - in the configuration file, 5-8
 - opening, 5-6
 - saving, 5-9
 - saving under a different name, 5-10
 - search path, 5-2
 - specifying on command line, 2-8, D-4
 - SYS REQ (3270 key), A-5
- ## T
- TAB (3270 key), A-11
 - Technologies and guidelines, xiv
 - Common User Access (CUA), 1-1
 - Systems Application Architecture (SAA), xiv
 - Common Communications Support (CCS), xv
 - Common User Access (CUA), xiv
 - Systems Network Architecture (SNA), xiv
 - TEMPEXIT (3270 key), 2-2, 2-13, 2-14, 2-20, 2-21, 2-24, 2-26, 2-30, 3-15, 4-16, A-6
 - Temporary exit, 2-4, D-3
 - Terminal requirements, 1-2
 - Terminfo database, 1-2
 - Transferring files, 3-15, 4-1
 - Translation table, 2-8

Index

custom translation tables, 5-37

Type-ahead buffer, 2-5, 2-24

U

UP (3270 key), A-10

User interface, 1-1

terminal requirements, 1-2

V

VM/CMS host

file transfer, 4-12

see also File transfer

3270 LINKix

HLLAPI Programmer's Guide

June 1994

RESTRICTED RIGHTS

Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (C)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013. Cleo Communications.

LIMITED RIGHTS

Limited rights shall be effective indefinitely and are not subject to expiration as set forth in paragraph (3) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013.

Cleo Communications, 3796 Plaza Drive, Ann Arbor MI 48108
(A Division of Interface Systems, Inc.)



June 10, 1994

3270 LINKix HLLAPI Programmer's Guide

June 1994

Cleo Communications reserves the right to, without notice, modify or revise all or part of this document and/or change product features or specifications and shall not be responsible for any loss, cost or damage, including consequential damage, caused by reliance on these materials.

This document may not be reproduced, stored in a retrieval system or transmitted, in whole or in part, in any form or by any means (electronic, mechanical, photocopied or otherwise) without the prior written permission of Cleo Communications.

© 1993 Cleo Communications – All rights reserved.

Portions:

© 1989–1992 Data Connection Limited

© 1989, 1990 Digital Communications Associates, Inc.

© 1989, 1990 Microsoft Corporation

Table of Contents

1. Introducing HLLAPI	1 - 2
What is HLLAPI?	1 - 2
How HLLAPI Works	1 - 3
Compatibility	1 - 3
Required Knowledge	1 - 4
Intended Audience	1 - 4
Prerequisite Knowledge	1 - 4
How a HLLAPI Application Works	1 - 5
Typical Uses of HLLAPI	1 - 6
Automating Startup of Host 3270 Applications	1 - 6
Simplifying Host 3270 Applications	1 - 6
Accessing Host Applications Designed for Larger Screens	1 - 6
Documentation Conventions	1 - 7
 2. Programming in the 3270 Environment	
Using the 3270 Presentation Space	2 - 2
The Presentation Space	2 - 2
Screen Buffer	2 - 2
Converting Row and Column Coordinates	2 - 7
Fields in the Presentation Space	2 - 8
Working with Unformatted Presentation Spaces	2 - 8
Working with Formatted Presentation Spaces	2 - 8
Field Attributes	2 - 9
Locating Fields	2-11
Auto-skip Fields	2-11
Detecting Modified Fields	2-11
Extended Attributes	2-13
Extended Field Attributes	2-13
Extended Character Attributes	2-14
Relationship of Extended Attributes to Field Attributes	2-15
Base Colors	2-17

Contents

Attribute Bytes Returned to HLLAPI Applications	2-18
Cursor Movement within HLLAPI Functions	2-19
3. Writing Transaction Programs	
General Guidelines	3 - 2
Programming Host Transactions	3 - 4
Sending Data to the Host	3 - 5
Logging onto a Host System	3 - 5
Waiting for Host Responses	3 - 6
Single-session UNIX Environment	3 - 7
Multi-session UNIX Environment	3 - 7
Synchronizing with the Host Program	3 - 8
Using HLLAPI Functions in C Programs	3-10
Declarations	3-10
Data String Size	3-11
Initializing Function Parameters	3-12
Requesting a Function	3-12
Accessing Values from Returned Data Strings	3-14
Bit Numbering Convention	3-14
Byte Numbering Convention	3-14
Accessing Two-byte Numeric Values	3-14
Requesting Multiple HLLAPI Functions	3-15
Writing Portable Applications	3-16
Multiple Processes	3-17
Use of signals	3-17
Fast Path Checklist	3-18
4. Testing and Troubleshooting	
Compiling and Linking a HLLAPI Application	4-2
Running a HLLAPI Application	4-3
Tracing HLLAPI Applications	4-4
Tracing an Application	4-5
SNATRC	4-5
SNAMSG	4-6

SNATRUNC	4 - 6
SNACTL	4 - 7
Controlling Tracing from within an Application	4 - 7
HLLAPI Trace File Format	4 - 8
Process ID	4 - 8
Parameter Data	4 - 8
Message Data	4 - 9
Sample Trace Output	4 - 9

5. HLLAPI Functions

Function Information	5 - 2
Symbolic Function Name and Function Number	5 - 2
Prior Functions	5 - 2
Request Parameters	5 - 2
Returned Parameters	5 - 3
Affected by Set Session Parameters?	5 - 4
Usage and Restrictions	5 - 4
Additional Information	5 - 4
Alphabetical Listing	5 - 5
Functional Listing	5 - 7
Important Terms and Concepts	5-10
Connect Presentation Space	5-12
Convert Position or RowCol	5-15
Copy Field to String	5-18
Copy OIA	5-21
Copy Presentation Space	5-25
Copy Presentation Space to String	5-28
Copy String to Field	5-31
Copy String to Presentation Space	5-34
Disconnect Presentation Space	5-37
Display Attribute	5-39
Display Off	5-42
Display On	5-45
Display Text	5-47

Contents

Find Field Length	5-50
Find Field Position	5-53
Get Key	5-55
Notify Event	5-59
Pause	5-62
Post Intercept Status	5-64
Query Cursor Location	5-66
Query Field Attribute	5-67
Query Host Update	5-69
Query Session Status	5-72
Query Sessions	5-74
Query System	5-77
Receive File	5-80
Release	5-84
Reserve	5-86
Reset System	5-88
Search Field	5-90
Search Presentation Space	5-93
Send File	5-96
Send Key	5-100
Send Multiple	5-105
Set Cursor	5-108
Set Session Parameters	5-110
Start Host Notification	5-120
Start Keystroke Intercept	5-123
Stop Host Notification	5-125
Stop Keystroke Intercept	5-127
Storage Manager	5-129
Wait	5-133

A. OIA Image Information

B. Sample Program

C. Header File: hapi_c.h

2. Programming in the 3270 Environment

This chapter explains features of the 3270 environment which you need to consider when developing HLLAPI application programs.

The following topics are discussed in this chapter:

- ◆ Using the 3270 presentation space
- ◆ Fields in the 3270 presentation space
- ◆ Extended attributes
- ◆ Base screen colors

Using the 3270 Presentation Space

Your application program and the host communicate through the 3270 presentation space (display). This section describes the presentation space and explains how to access information in the presentation space. Operator Information Area (OIA)

.i.presentation spaceOperator Information Area (OIA)

The Presentation Space

Four display screen sizes are supported: models 2,3,4 and 5.

Model	—Screen Size—	
	Rows	Columns
2	24	80
3	32	80
4	43	80
5	27	132

Depending on the support provided by your UNIX computer and the terminal you are using, you may not be able to view all of these screen models. However, you can still access a 3270 session using HLLAPI even though the presentation space for that session cannot be displayed on the screen.

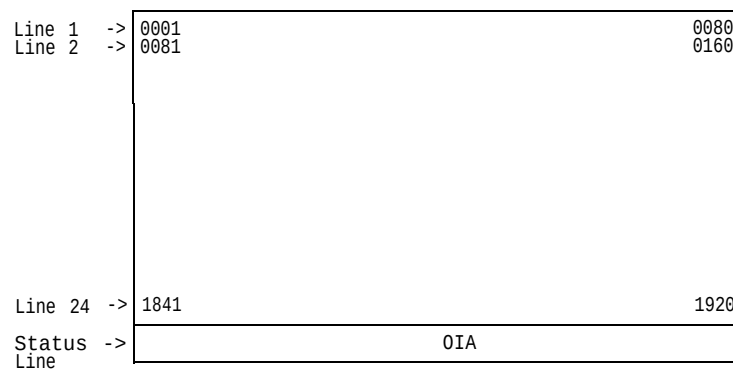
The row totals shown for each screen model do not include the status line (also called the operator information area). The status line is a line at the bottom of each screen that shows the status of the session and information concerning the presentation space. The status line is not considered part of the presentation space.

Screen Buffer

Information displayed in the presentation space is stored in a screen buffer. Using HLLAPI functions, your application program can read from and write to the screen buffer. To access data in the screen buffer, you must supply the position of the presentation space containing that data.

The addressing scheme is the same for each screen model. Presentation spacescreen addresses start at position 1 and end with the maximum number of presentation space positions for the screen model. For example, Model 2 screens have 1,920 positions; Model 3 screens have 2,560 positions; Model 4 screens have 3,440 positions; and Model 5 screens have 3,564 positions.

The following diagram shows the presentation space positions for a Model 2 screen.



The tables that follow list the beginning and ending positions for each line of the various 3270 screen models:

- ◆ Model 2 (24 x 80)
- ◆ Model 2 (32 x 80)
- ◆ Model 3 (43 x 80)
- ◆ Model 5 (27 x 132)

Programming in the 3270 Environment

Model 2

Line	Starting addresses		Ending addresses	
	Hex	Decimal	Hex	Decimal

1	0001	0001	0050	0080
2	0051	0081	00A0	0160
3	00A1	0161	00F0	0240
4	00F1	0241	0140	0320
5	0141	0321	0190	0400
6	0191	0401	01E0	0480
7	01E1	0481	0230	0560
8	0231	0561	0280	0640
9	0281	0641	02D0	0720
10	02D1	0721	0320	0800
11	0321	0801	0370	0880
12	0371	0881	03C0	0960
13	03C1	0961	0410	1040
14	0411	1041	0460	1120
15	0461	1121	04B0	1200
16	04B1	1201	0500	1280
17	0501	1281	0550	1360
18	0551	1361	05A0	1440
19	05A1	1441	05F0	1520
20	05F1	1521	0640	1600
21	0641	1601	0690	1680
22	0691	1681	06E0	1760
23	06E1	1761	0730	1840
24	0731	1841	0780	1920

Model 3

The first 24 lines of a Model 3 screen (32 x 80) are the same as for a Model 2 (24 x 80). The remaining lines are listed below.

Line	Starting addresses		Ending addresses	
	Hex	Decimal	Hex	Decimal
25	0781	1921	07D0	2000
26	07D1	2001	0820	2080
27	0821	2081	0870	2160
28	0871	2161	08C0	2240
29	08C1	2241	0910	2320
30	0911	2321	0960	2400
31	0961	2401	09B0	2480
32	09B1	2481	0A00	2560

Model 4

The first 32 lines of a Model 4 screen (43 x 80) are the same as for a Model 3 (32 x 80) screen. The remaining lines are listed below.

Line	Starting addresses		Ending addresses	
	Hex	Decimal	Hex	Decimal
33	0A01	2561	0A50	2640
34	0A51	2641	0AA0	2720
35	0AA1	2721	0AF0	2800
36	0AF1	2801	0B40	2880
37	0B41	2881	0B90	2960
38	0B91	2961	0BE0	3040
39	0BE1	3041	0C30	3120
40	0C31	3121	0C80	3200
41	0C81	3201	0CB0	3280
42	0CD1	3281	0D20	3360
43	0D21	3361	0D70	3440

Programming in the 3270 Environment

Model 5

Even though this model has 27 rows and 132 columns, the status line contains the same amount of information (80 bytes) as the status line for other terminal models.

Line	Starting addresses		Ending addresses	
	Hex	Decimal	Hex	Decimal
1	0001	0001	0084	0132
2	0085	0133	0108	0264
3	0109	0265	018C	0396
4	018D	0397	0210	0528
5	0211	0529	0294	0660
6	0295	0661	0318	0792
7	0319	0793	039C	0924
8	039D	0925	0420	1056
9	0421	1057	04A4	1188
10	04A5	1189	0528	1320
11	0529	1321	05AC	1452
12	05AD	1453	0630	1584
13	0631	1585	06B4	1716
14	06B5	1717	0738	1848
15	0739	1849	07BC	1980
16	07BD	1981	0840	2112
17	0841	2113	08C4	2244
18	08C5	2245	0948	2376
19	0949	2377	09CC	2508
20	09CD	2509	0A50	2640
21	0A51	2641	0AD4	2772
22	0AD5	2773	0B58	2904
23	0B59	2905	0BDC	3036
24	0BDD	3037	0C60	3168
25	0C61	3169	0CE4	3300
26	0CE5	3301	0D68	3432
27	0D69	3433	0DEC	3564

Converting Row and Column Coordinates

HLLAPI provides the Functions `CONVERT POSITION` or `ROWCOL` function to convert row and column coordinates on any 3270 screen size into a presentation space position. You can also use this function to convert a presentation space position into row and column coordinates.

Note: If you are using a Model 5 terminal, the host can toggle to an 80-column screen without notice. Have your application check the column mode to determine the column width set by the host.

Fields in the Presentation Space

Your application can work with unformatted and formatted presentation spacefields. What the host writes into a given presentation space dictates whether it's unformatted or formatted. Your HLLAPI application cannot control this aspect of the host program, but can only react to it. This section explains how to work with both types and how field attributes are used to define fields in formatted presentation spaces.

Working with Unformatted Presentation Spaces

When working with unformatteds, the HLLAPI functions see the presentation space as one large field that starts at the first location in the space and ends at the last location. The absence of field attributes indicates an unformatted presentation space.

Because an unformatted presentation space has no individually defined fields, your application can use HLLAPI functions to write information to any location in the presentation space. However, when writing an application that communicates with a host application program, you must ensure that your program is writing data to the location being read by the host application program. Since some host programs are designed to read information from specific locations in an unformatted presentation space, while others read the entire presentation space, you must determine how the host application reads unformatted presentation spaces in order to correctly design and write your application.

Typically, the host application will expect data to be written starting at the current cursor position.

Working with Formatted Presentation Spaces

Formatted s have one or more fields that contain specific types of data. These fields present special problems for HLLAPI application programs. Applications that send keystrokes to formatted presentation spaces need to consider the following conditions:

- ◆ Prompts for each field
- ◆ Characteristics of each field
- ◆ Keystrokes required to complete an entry

Failure to take these conditions into account can cause your application to send bad data to the host. For example, when sending data to a field, your program should determine if data already exists in the target field. If data exists, your application should clear the field before sending any new data. This ensures that the existing data is not appended to the data sent by your application. To facilitate working with formatted presentation spaces, the Functions `SET SESSION PARAMETERES` function allows you to return field attributes to your program. See the `SET SESSION PARAMETERS` discussion in chapter 3, *Writing Transaction Programs*, for more information.

In addition, even small changes in a host program's screen layout can cause your application to fail. Make sure there is close coordination between the local and host sites when the host application program changes, and design your application programs so that you can easily modify them to conform to changes in the host program.

For example, if your application expects a particular prompt, your program will probably fail if even a single character of the prompt from the host program is changed. In this case, you must correct your application to match the host application program.

Field Attributes

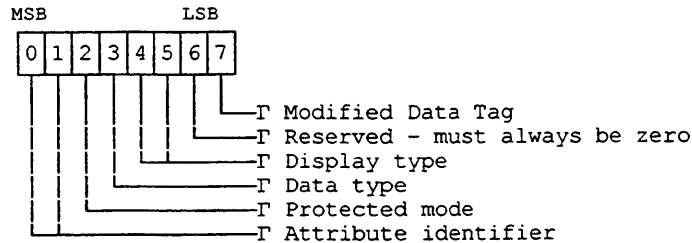
Every field in a formatted has a corresponding fields attribute byte to its immediate left. This byte of information defines the type of information that can be entered in that field by an operator or application program. Field attributes define the following types of information for each field in the presentation space.

- ◆ Beginning location of a field
- ◆ Whether the field is protected or unprotected
- ◆ Type of data allowed in the field (alphanumeric, numeric or both)
- ◆ Appearance of the field (displayed, not displayed or intensified)
- ◆ Whether the field can be detected by a light pen
- ◆ Whether the field has been modified since it was put on the presentation space

Note: Bit 0 is the most significant bit and bit 7 is the least significant bit.

Programming in the 3270 Environment

The following diagram shows the field attribute byte and the information defined for each field in the presentation space.



The following table explains the possible values for each bit in the attribute byte.

Name	Bit(s)	Values
Attribute identifier	0-1	always 11
Protected mode	2	0- unprotected 1- protected
Data type	3	0- alphanumeric 1- numeric
Display type	4-5	00 - normal display, non-detectable 01- normal display, detectable 10 - bright display, detectable 11- non-display, non-detectable
Reserved	6	always 0
Modified data tag	7	0- unmodified field 1- modified field

Field attribute bytes can be distinguished from normal displayable ASCII characters by the fact that the two most significant bits (bits 0 and 1) are always set. The attribute byte defines the start of a new field; that field continues until the next attribute byte. The screen displays attribute bytes as blanks, but they can be read by your program. Use the FunctionsQUERY FIELD ATTRIBUTE function to examine any specific field attribute value.

Locating Fields

The 3270 emulation program locates the beginning of in the presentation space through the field attribute byte. When searching for a field, the 3270 emulation program searches until it finds a byte with a value greater than or equal to 0xC0 (two most significant bits set). This byte signifies the beginning of the next field. All data between the field attribute byte and the next attribute byte (which indicates the beginning of the next field) is part of the field. Searching wraps from the end of one row to the beginning of the next row and from the lower-right corner (24,80) to the upper-left corner (1, 1).

The FunctionsFIND FIELD POSITION function can be used to help find fields in the presentation space.

Auto-skip Fields

When an application sends keystrokes to an unprotected field in a formatted presentation space using the FunctionsSEND KEY or FunctionsCOPY STRING TO FIELD function, the cursor automatically skips to the beginning of the next available field after data is entered in the last character space of the current field.

The only exception is when the attribute byte of the next available field is set for protected mode *and* numeric data type. Protected numeric fields are called auto-skip fields, as the SEND KEY and COPY STRING TO FIELD functions bypass them for the next unprotected field.

When using the SEND KEY and COPY STRING TO FIELD functions you should keep these auto-skip fields in mind, because HLLAPI cannot enter keystrokes into them. You do not, however, need to consider auto-skip fields when using other HLLAPI functions, such as FunctionsCOPY STRING TO PRESENTATION SPACE, because they do not operate in this manner.

Detecting Modified Fields

When the information in a field in the screen buffer is modified by an operator or an application program, the fieldsModified Data Tag (MDT) is set to 1. The MDT is bit 7 in that field's attribute byte. The 3270 emulation program uses this bit to determine which fields to send to the host when your application program or an operator sends an to the 3270 emulation program. When keys are sent to the 3270 emulation program, the program sets the MDT for the field. When strings are copied to the presentation space, HLLAPI automatically sets all necessary MDT bits.

Programming in the 3270 Environment

A host's typical reaction to receiving your AID key will be to clear all the MDT bits when it sends you its reaction/results.

Use the FunctionsQUERY FIELD ATTRIBUTE function to examine any specific field attribute value.

Extended Attributes

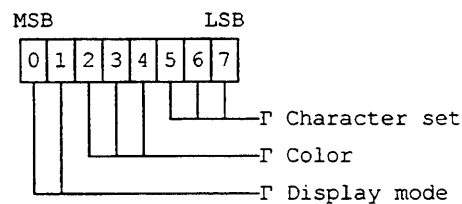
The field attribute byte cannot store all of the information about the fields and characters in the presentation space. Two other types of information, called extended field attributes (EFAs) and extended character attributes (ECAs), are also kept for each field and character in the presentation space if the 3270 emulation program and host application program are configured to support extended attributes. This information is kept in the extended attribute buffer. This section explains EFAs and ECAs and how they work together and with the field attributes to define the characteristics of fields in the presentation space.

Extended Field Attributes

Extended field attributes define characteristics that apply to the entire field unless extended character attributes override them. Extended field attributes control the following attributes:

- ◆ Highlighting
- ◆ Color
- ◆ Character set

Each field has an extended attribute byte (EAB) that defines the extended attributes for that field. The following diagram shows the extended field attribute byte and identifies the bits in the field.



Programming in the 3270 Environment

The following table explains each bit in the byte and the valid values for the bits.

Name	Bit(s)	values	Meanings
Display mode	0-1	00	Normal mode
		01	Blink character
		10	Reverse video character
		11	Underline character
Color	2-4	000	Default to base color
		001	Blue
		010	Red
		011	Pink
		100	Green
		101	Turquoise
		110	Yellow
		111	White
Character set	5-7	000	Base character set
		001	Reserved
		010	Reserved
		011	Reserved
		100	Reserved
		101	Reserved
		110	Reserved
		111	Reserved

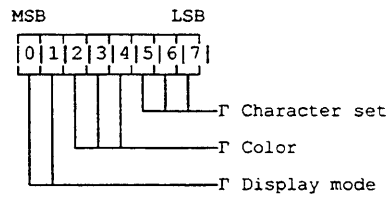
Extended Character Attributes

Extended character attributes control the characteristics of individual characters in a field. ECAs control the following characteristics:

- ◆ Highlighting
- ◆ Color
- ◆ Character set

Each character in a formatted presentation space has an associated EAB that defines the extended attributes for that character.

The following diagram shows the extended character attribute byte.

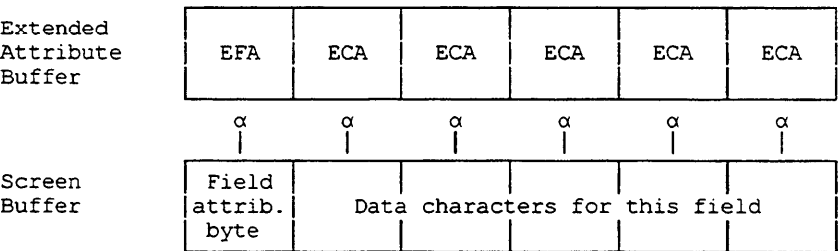


The following table explains each bit in the extended character attribute byte.

Name	Bit(s)	Values	Meanings
Display mode	0-1	00	Defaults to most recent EFA
		01	Blink character
		10	Reverse video character
		11	Underline character
Color	2-4	000	Defaults to most recent EFA
		001	Blue
		010	Red
		011	Pink
		100	Green
		101	Turquoise
		110	Yellow
		111	White
Character set	5-7	000	Defaults to most recent EFA
		001	Reserved
		010	Reserved
		011	Reserved
		100	Reserved
		101	Reserved
		110	Reserved
		111	Reserved

Relationship of Extended Attributes to Field Attributes

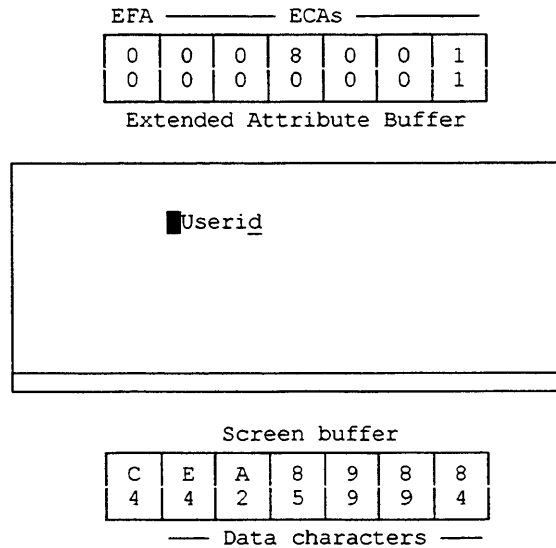
Extended attributes and field attributes work in conjunction to define the characteristics of a field and its characters. The following diagram shows this concept.



Note: Fields defined as non-display fields in the field attribute byte are never displayed on the screen, regardless of the extended attributes.

There is a hierarchy in the way the 3270 emulation program interprets and handles the combination of field attributes and extended attributes for fields in the presentation space. EFAs apply to the whole field; ECAs take precedence over EFAs, but apply only to a specific character.

The following diagram demonstrates how this principle affects the presentation space.



The diagram shows a 6-character field in a presentation space. The field attribute character at the beginning of the field defines the field as an unprotected, alphanumeric, normal display field that has not been modified since the last time the controller polled the 3270 emulation program.

The first and second characters (U and s) do not have ECAs defined. Therefore, they use the EFA which is set to default values. The ECA for the third character (e) causes it to be displayed as reverse video (this is not shown in the diagram). Since the fourth and fifth characters (r and i) do not have ECAs defined, they use the EFA; the ECA for the last character (d) defines it as underlined, so it is underlined.

Base Colors

Extended attributes define the color for or in the presentation space. However, field attributes are used with the color options supported by your 3270 emulation program to determine the base color for a screen. If extended attributes are not defined for a field, the color of the field is determined by its field attribute.

The following table shows how the color option and the field attributes determine the base color. This base color corresponds to one of the four base colors listed in the color mappings dialog of the 3270 customization program; the actual color displayed on the screen depends on the color mappings you are using for that session.

Programming in the 3270 Environment

Attribute Field attributes	Color options		
	bits	Mono	Color
Unprotected, normal intensity	0x0x	Green	Green
Unprotected, intensified	0x10	Green	Red
Protected, normal intensity	1x0x	Green	Blue
Protected, intensified	1x10	Green	White

Attribute Bytes Returned to HLLAPI Applications

If you use the EAB option (without the XLATE option below) on FunctionsSET SESSION PARAMETERS to return extended attribute bytes to your HLLAPI program, each character copied from the presentation space is followed by its extended character attribute (ECA); this will have a zero value if an ECA is not defined for that character.

If you also use the XLATE option to translate attribute bytes to CGA format each character is followed by an attribute byte in CGA format which indicates the appearance of that character on the screen. If no ECA is defined for the character, HLLAPI determines the appearance of the character following the hierarchy described in this section (the field's extended field attribute if no ECA is defined for this character, or the field attribute byte and base colors if no EFA is defined for this field), generating a corresponding attribute byte in CGA format. CGA format attribute bytes, therefore, give a full description of the appearance of each character on the screen, but do not necessarily correspond to the ECAs for those characters.

Cursor Movement within HLLAPI Functions

As keystrokes are typed on a real 3270 terminal, the cursor automatically advances to the next column. This type of advancement is called tracking.

This cursor tracking is duplicated in only one of the HLLAPI functions that place data in presentation spaces (FunctionsSEND KEY). Cursor tracking is *not* performed by the FunctionsCOPY STRING TO FIELD or FunctionsCOPY STRING TO PRESENTATION SPACE functions—the cursor remains at its original position/offset after invoking either of the above copy functions. When using either one of these functions, you must use the FunctionsSET CURSOR function to perform any required cursor positioning before supplying an Attention Identifier (AID) key to the FunctionsSEND KEY function.

3. Writing Transaction Programs

This chapter provides information on developing application programs using CLEO's HLLAPI programming interface.

Included in this chapter are the following:

- ◆ General guidelines for designing application programs
- ◆ Techniques for programming host transactions
- ◆ Using HLLAPI functions in C programs
- ◆ Writing portable applications
- ◆ Fast path checklist

General Guidelines

For any application program that uses HLLAPI to send transactions to a host computer and receive responses from the host, there are certain steps that the program needs to execute. They areas follows:

1. Use the FunctionsRESET SYSTEM function at program start to establish a common HLLAPI foundation. Any pre-existing modes/states/etc. will be discarded.
2. Use the FunctionsSET SESSION PARAMETERS function to set the HLLAPI session parameters to appropriate values; this can be done before connecting to a session or during the session. See SET SESSION PARAMETERSin Chapter 5 for more information.

These parameters determine how your program interacts with HLLAPI and the emulation program. Using the right options can greatly simplify a program. For example, the default end-of-text copy option is STRLEN (string lengths must be passed explicitly); from C programs, it is generally easier to use the alternative STREOT and EOT=0x00 (end of a string is indicated by a binary zero).

3. Connect to a host session and check the status of the session using the FunctionsCONNECT PRESENTATION SPACE function. For the program to write to the presentation space, the session must not be busy (waiting for a host response) or locked (no further input is being accepted, following an error by the 3270 user).
4. If the session is busy or locked, your program should issue a FunctionsWAIT function request to wait for the session to become available, switch to another session or display an appropriate error before terminating. You should not specify the LWAIT (wait forever) set session parameter when using WAIT in this manner.
5. Once a program verifies that the session is available for input, you should lock the session from operator input by using the FunctionsRESERVE function or the functions FunctionsSTART KEYSTROKE INTERCEPT, FunctionsGET KEY POSITION and FunctionsSEND KEY to limit the operator's interaction with the presentation space. Designing a program to distinguish operator input from host responses can be very difficult.

6. Avoid dependence *fixed* presentation space positions when locating input and output fields. Instead, use positions that are *relative* to nearby prompts or headings — HLLAPI search functions can be used to help. While this technique may require more initial coding, it will help insulate your application from minor screen layout changes made in the host application.
7. When performing host transactions, be sure to give the host sufficient time to respond before you continue. If the host and application programs get out of synch, re-synching can be very difficult. You can check where you are in the application by searching the presentation space for application dependent character strings displayed by the host.
8. When working with line-by-line host applications, remember that program behavior often changes on the last line of the screen; for example, an extra ENTER keystroke may be required. You can use the FunctionsQUERY CURSOR LOCATION function to find where you are on the screen.
9. After all host transactions are completed for the connected session, you will need to release and disconnect the session before terminating the program. Use the FunctionsRELEASE and FunctionsDISCONNECT PRESENTATION SPACE functions to perform these actions.

In addition, some hosts may insist (for security reasons) that you logout/logoff before disconnecting.

Programming Host Transactions

To process host transactions, an application program may automate automating manual operations in the following ways:

- Use the FunctionsSEARCH PRESENTATION SPACE function to find any host messages or prompts in the presentation space.
- Use the FunctionsCOPY STRING TO FIELD, FunctionsCOPY STRING TO PRESENTATION SPACE, and FunctionsSEND KEY functions to send information to the host. See **Sending data to the host program**, later in this section, for more information.
- Use the FunctionsWAIT or FunctionsPAUSE function to give the host time to respond. See **Waiting for host responses**, later in this section, for more information.
- Use the FunctionsSEARCH PRESENTATION SPACE function to determine how the host has responded.
- Use the FunctionsCOPY PRESENTATION SPACE or FunctionsCOPY PRESENTATION SPACE TO STRING function to retrieve the host response.

In each of these steps, it is important that the application program always check the function's return code to see if subsequent steps should be avoided.

Also, it is important to note that some host applications do not keep a presentation space locked, so the program may need to loop on the WAIT function while waiting for a response from the host.

Your HLLAPI program should check the return code for all function calls before continuing.

In general, if the return code is 0, the function completed successfully and the program should continue. If the return code is 9 (system error), your HLLAPI program should call a subroutine that requests the FunctionsQUERY SYSTEM function to examine your program logic. You should have the information generated by QUERY SYSTEM available when contacting CLEO Technical Support.

Sending Data to the Host

The FunctionsCOPY STRING TO FIELD and FunctionsCOPY STRING TO PRESENTATION SPACE functions can be used to copy data to the 3270 screen buffer. These functions transfer data to the screen buffer more efficiently than the FunctionsSEND KEY function.

However, these functions simply write data into the screen buffer without requiring the 3270 emulation program to take any action. In order to tell the 3270 emulation program to process the information you have placed in the screen buffer, your application must use the SEND KEY function to send an (ENTER, PA1, PF1 and so on).

Logging onto a Host System

A HLLAPI application can be written that supplies the necessary keystrokes to perform a logon sequence to a 3270 host system. This logon sequence must take a number of things into consideration in order to operate correctly. For instance, the host can draw the 3270 screen in either of two ways: formatted or unformatted. Before you have logged on, the screen will normally be unformatted. Once you have logged on, the host will generally draw a formatted screen; this means that the screen is divided into several areas known as fields. Fields may differ in color and intensity; they may also be restricted so that you cannot enter data in certain fields, or can enter only certain types of data-

After a 3270 session connects to a host system and your HLLAPI application has issued a logon request, a number of events must take place before any further actions can be carried out by the application. The events that occur at the host include changes in the status line, the keyboard status (locked/unlocked) and the contents of the screen (presentation space).

When a 3270 session connects to a host system, the typical host session begins in an SSCP-LU session. After your application issues a logon request, the screen will clear, the keyboard will lock and unlock (and many times it will lock again briefly), the status line will indicate a change from SSCP-LU to LU-LU status and, finally, the host's screen will change (the subsequent screen may or may not require additional actions on your part). Obviously, during a logon sequence it's critical to monitor the actions of the host making sure that all of these events have occurred before you proceed with any further actions.

Waiting for Host Responses

Probably no other area of a HLLAPI program generates more questions or difficulty than the area responsible for waiting on the host's response. Not only does your application require logic to handle every conceivable host response, but it must also be capable of detecting that the host *has* responded and when it has *finished* responding.

Numerous methods exist for dealing with these needs. The traditional method is to loop until the word WAIT has been removed from the status line. When a terminal operator or application program sends a command to the host computer, the terminal assumes a waiting state until the command completes. While the host computer is processing the command, the word WAIT appears on the status line; it remains on the status line until the command is finished processing. When the word WAIT has been removed from the status line, the host is ready to process another command.

While your HLLAPI program can use a function in an idle loop to determine when the session is finished, due to the multi-tasking nature of UNIX you should not write a "waiting-for-host" loop that causes continuous cpu consumption. For example, do not place HLLAPI into NWAIT (No Wait) mode and then proceed to tightly loop on the FunctionsWAIT function. Although doing so would not produce any errors, it would rob excessive cpu cycles from other UNIX procedures without any effective gain in performance.

The examples that follow explain how you might wait for the host to respond by using the WAIT function; other methods may be used, such as using the FunctionsQUERY HOST UPDATE function to notify the program of updates to the status line. The complexity of the method that's right for you will be influenced by such factors as the number of sessions being managed and whether other work must be accomplished while waiting for the host response.

In the example methods that follow, when we instruct you to send an (such as **ENTER** or **PF3**) we assume you've already updated the desired presentation space with any necessary text (via any of the functions: FunctionsCOPY STRING TO FIELD, FunctionsCOPY STRING TO PRESENTATION SPACE or FunctionsSEND KEY). You must then use the FunctionsSEND KEY function to emulate the desired AID key and initiate the transmission to the host.

Please examine the examples carefully, and recognize that they are not to be considered the only way to accomplish the desired goal.

Single-session UNIX Environment

The example that follows conforms to the guidelines previously outlined and, therefore, may be used for single-session UNIX applications:

- 1 . Send an AID key to the host.
- 2 . Issue a FunctionsWAIT request. (This assumes the default wait-style of TWAIT has not been overridden via the FunctionsSET SESSION PARAMETERS function).
- 3 . Execution of your application will be suspended until the host has responded *completely* or until approximately one minute elapses, whichever occurs first (*completely* means that the host has sent all of the data it intends to, in addition to unlocking the keyboard). Note that some host applications unlock the keyboard *before* sending their entire response. Using the WAIT function will prevent miscues by your application on partial screens when the host unlocks the keyboard prematurely.
- 4 . The resulting return code will indicate which of two events occurred:

If the return code contains the value HARC_SUCCESS, this indicates that the host has responded *completely* and the session is available for input (the keyboard is unlocked). You should examine the session image for the host's response and then construct your next host transmission.

If the return code contains the value HARC_TIMEOUT, this indicates that the host has yet to *completely* respond during the one-minute that the WAIT function was invoked. At this point it would be wise to use the FunctionsCOPY OIA function to determine if the connection to the host is still intact. If so, you may elect to reissue the WAIT function. Otherwise you must consider the situation as an exception condition and treat it according to the requirements of your application.

Any other return code value from the WAIT function should be considered an exception-condition, possibly requiring application-specific reaction(s).

If the one minute timeout duration is too long for your specific needs, consider using the FunctionsPAUSE function after establishing IPAUSE mode via Functions SET SESSION PARAMETERS.

Multi-session UNIX Environment

UNIX programmers must decide whether to create a single process that deals with multiple sessions or multiple processes (possibly clones of each other) that each deal with a single (but different) session. The following strategy assumes a single process dealing with multiple sessions.

1. Use the FunctionsSET SESSION PARAMETERS function to place HLLAPI into NWAIT (No WAIT) mode. This need only be done once at the start of the application.
2. Send an for the desired presentation space.
3. Loop on both the FunctionsWAIT function (targeted toward the presentation space you are waiting on by prefixing the WAIT with a FunctionsCONNECT PRESENTATION SPCAE) and any other processing to be done (possibly dealing with other presentation space sessions). Some applications lend themselves to an architecture that loops in round-robin fashion across all your presentation sessions looking for host responses to deal with. Call the UNIX sleep() function after each pass across all of the sessions in the loop. The appropriate duration is somewhat application specific, but would rarely exceed five seconds and should approach one second as the quantity of outstanding host responses increases and/or at high baud rates.
4. When a complete host response is detected in Step 3, deal with it as required by your application.

Alternate methods exist in HLLAPI to help deal with this situation. Please read the descriptions concerning FunctionsSET SESSION PARAMETERS' IPAUSE mode and the FunctionsQUERY HOST UPDATE function.

Regardless of the method you use to detect host responses, be sure it includes a call to the WAIT function (in NWAIT mode) to determine if the detected update is the *complete* response from the host.

Synchronizing with the Host Program

Many things can affect the time application design synchronizing with the host computer takes to execute a command. The amount of time taken to clear the WAIT condition can vary between a few hundredths of a second to several seconds depending on the configuration, system load and so on. For this reason, it is

advisable for your program to ensure that the host has responded (using a method such as the one previously outlined) before attempting any further host transactions.

To ensure that the host has responded, your application could monitor a specific region of the screen buffer for a certain mainframe application message which would be expected following the command from your program. When the message has appeared and the keyboard is unlocked (the order in which these occur is dependent on the host system), your application can continue with the next host transaction.

If this is impossible and no other synchronization method can be found, the only automatic way to check for completion of the command may be to increase the period your application waits for the host to respond, even though this will degrade performance.

Using HLLAPI Functions in C Programs

This section explains how to declare HLLAPI structures, constants and parameters in your C program, how to initialize the parameters supplied to each function and how to call the function.

Declarations

HLLAPI application programs should declare or use the following data types:

- ◆ Variables
- ◆ Constants
- ◆ Function constants

The structures and function constants can be defined automatically by putting the `hapi_c.h` include file in your source module(s):

```
#include <hapi_c.h>
```

When you call a HLLAPI function, four HLLAPI arguments are passed. These arguments consist of three integers (passed indirectly using pointers) and a character string. They should be declared in the following form (the argument names are arbitrary):

```
int  api_func;  
char api_string [DATA_SIZE] ;  
int  api_len;  
int  api_ret;
```

The arguments may be global to your program or local to a function. The length of the character string for the second parameter, shown as `DATA_SIZE` in the above example, varies depending on the particular HLLAPI function being used; see the next section, **Data String Size**, and consult the function description to determine the appropriate value to use for the size.

The four arguments may be given any names within the program, but must be passed to HLLAPI in the correct order. The meaning of each argument varies according to the particular HLLAPI function being called; typical usage is given below.

Argument 1 (integer) is used for the application design passing the function code of the HLLAPI function being called.

Argument 2 (character string) is application design passing/returning a string of text or control information supplied to HLLAPI or an empty string into which HLLAPI will return data.

Argument 3 (integer) is often used to specify the length of the string which is being supplied as Argument 2, but may be used to supply or return other numeric values.

Argument 4 (integer) is often used to supply or return a application design passing/returning a presentation space position offset (the position on the 3270 display where data was found or where an action is to be performed). It is also used for return codes from HLLAPI indicating whether or not a HLLAPI function was called successfully, or to supply or return other numeric values.

Data String Size

The data string supplied as the second argument to a HLLAPI function may be any size; for HLLAPI functions that return data to the program, you should ensure that it is large enough to hold the maximum amount of data which may be returned. These functions are FunctionsCOPY FIELD TO STRING, FunctionsCOPY OIA, FunctionsCOPY PRESENTATION SPACE, FunctionsCOPY PRESENTATION SPACE TO STRING, FunctionsQUERY SESSIONS and FunctionsQUERY SYSTEM.

For example, COPY PRESENTATION SPACE requires a data string at least as large as the presentation space to be copied. This depends on the screen model of the 3270 emulation session:

Model 2 sessions	1920 characters
Model 3 sessions	2560 characters
Model 4 sessions	3440 characters
Model 5 sessions	3564 characters

Note: The data string sizes for COPY FIELD TO STRING, COPY PRESENTATION SPACE and COPY PRESENTATION SPACE TO STRING must be doubled if the EAB parameter is set in SET SESSION PARAMETERS, since this parameter instructs HLLAPI to copy the extended attribute byte for each character as well as the character itself.

Initializing Function Parameters

Before requesting a HLLAPI function, a program must initialize the variables used as application designinitializing function parameters with appropriate values.

All HLLAPI function calls require the first parameter to be set to a valid function code number. One or more of the three remaining parameters may also need to be initialized, depending on which function is being called. The description of each function in chapter 5, **HLLAPI Functions**, explains which parameters a given function must have initialized.

The following example code shows how a program can load these values prior to requesting the HLLAPI FunctionsCOPY STRING TO PRESENTATION SPACE function:

```
api_func = HA_COPY_STR_TO_PS;  
api_str = "list files * exec";  
api_len = strlen (api_str) ;  
api_retc = 1761;
```

This example performs the following:

1. Sets the function code for COPY STRING TO PRESENTATION SPACE.
2. Sets up the string to be copied - "listfiles * exec".
3. Sets the length parameter to the length of the string.
4. Sets the fourth parameter to 1761. This is the position in the presentation space where you want the function to place the copied data.

In this example initializing the length, `api_len`, is required only if the session parameter STRLEN (the default) is enabled. (See SET SESSION PARAMETERS in chapter 5, **HLLAPI Functions**, for more information on session parameters.)

Requesting a Function

Once you have initialized the function parameters, the program is ready to transfer control to the HLLAPI library.

To provide compatibility with other HLLAPI implementations, there are three function names which can be used to call HLLAPI: `hllapi ( )`, `hllc ( )` and `clim ( )`. You are advised to use the `hllapi ( )` function call in new applications; the other two are provided for compatibility with DOS HLLAPI applications.

The following code demonstrates the function call using `hllapi ( )`; `hllapi ( )` may be replaced by `hllc ( )` or `clim ( )` using the same syntax.

```
hllapi (&api_func, send_string, &api_len, &api_ret) ;
```

The four arguments are passed to HLLAPI by address.

Check the return code to see if the function completed successfully. If, for some reason, an invalid HLLAPI function code is specified in the HLLAPI function request, the `HARC_UNSUPPORTED` return code is returned.

The following sample code shows how you might handle the return code after calling HLLAPI with function code 15 (the FunctionsCOPY STRING TO PRESENTATION SPACE function code.) When designing return code handling for a function, refer to its description to determine which return code values need to be handled.

```
switch (api_ret)
{
    case HARC_SUCCESS:           /* All data was copied */
    case HARC_TRUNCATION:       /* * Some data was copied */
        break;

    case HARC_NO_CONNECT_ION:   /* Not connected */
        connect_error ( ) ;
        break;

    case HARC_INVALID_PS_POS:   /* PS position invalid */
    case HARC_BAD_PARM:         /* * String length bad */
        parameter_error (api_ret) ;
        break;

    case HARC_LOCKED:           /* * PS inhibited or bad data values */
        state_or_data_error (api_str) ;
        break;

    case HARC_SYSTEM_ERROR:     /* * System error */
        do_query_system ( ) ;   /* * call Query System function */
                                /* * and display returned data */
        terminate ;             /* * Cleanup & terminate application */
        break;

    default:                    /* unknown error */
        display_error (api_ret) ;
        terminate ;             /* Cleanup & terminate application */
        break;
}
```

Writing Transaction Programs

In this example the functions within the switch statement's body are functions that an application could have defined to handle various conditions. Their form and content are entirely dependent on the application's design and implementation.

Accessing Values from Returned Data Strings

Several HLLAPI functions (for example FunctionsQUERY SESSIONS) return formatted data strings which contain logical and numeric values as well as characters. When accessing data from these strings, you should be aware of the following:

Bit Numbering Convention

In this manual, the most significant bit of a byte is numbered bit 0 and the least significant bit is bit 7. This allows cross-referencing with IBM documentation, such as the 3274 Control Unit Description, where this convention is used.

Byte Numbering Convention

The bytes in a returned data string are numbered so that the first byte is byte 1, in order to retain compatibility with the descriptions of these strings in IBM HLLAPI documentation. Because the C programming language numbers the first byte of a string as byte 0, you need to allow for this in your program. For example, the description of the returned data string for the FunctionsQUERY SESSIONS function states that the session type is returned in byte 10; to access this byte, your program would need to access `data_string[9]`.

Accessing Two-byte Numeric Values

Where these formatted data strings contain two-byte binary numbers, HLLAPI always stores these as the least significant byte of the number followed by the most significant byte. To extract the numeric value from the data string, your program should multiply the value in the second position by 256 and then add the value in the first position. For example, if the description of a data string states that the presentation space size is in bytes 4 and 5 (equivalent to `data_string[3]` and `data_string[4]`), the numeric value is given as follows:

```
ps_size = ( (unsigned char) mystring[4] * 256 ) +  
           (unsigned char) mystring[3] ;
```


Don't try to access such numeric values by using a pointer to the first byte and casting it to an integer pointer. Even if this is the correct byte ordering for your UNIX computer, it may cause an alignment fault on some processors.

Requesting Multiple HLLAPI Functions

CLEO's HLLAPI implementation includes a "Super-HLLAPI" function that allows you to send up to 20 HLLAPI function calls at one time from an application. In order to use this function, SEND MULTIPLE, you declare and initialize an array of HLLAPI function parameters using the HLL_PARAMETERS structure (defined in *hapi_c.h*), and instead of placing a pointer to a character string in *api_str* you use a pointer to the array of HLLAPI function parameters.

This process is displayed in the source code example below which implements two HLLAPI verbs using a single function call:

```
HLL_PARAMETERS HllapiVerbs [2] ;

HllapiVerbs [0]. func_number = HA_SET_CURSOR;
HllapiVerbs [0]. string      = ;
HllapiVerbs [0]. data_length = 0;
HllapiVerbs [0]. pspos       = 726;

HllapiVerbs [1]. func_number = HA_SEND_KEY;
HllapiVerbs [1]. string      = "Hello World" ;
HllapiVerbs [1]. data_length = strlen (HllapiVerbs [1].string) ;
HllapiVerbs [1]. pspos       = 0;

api_func = HA_SEND_MULTIPLE ;
api_str = (char *) HllapiVerbs;
api_len = 1;
api_retcode = 0;
hllapi (&api_func, api_str, &api_len, &api_retcode) ;
if (api_retcode != HARC_SUCCESS)
{
    /* Handle any errors */
}
}
```

The *return_code* for this function indicates global success or failure; either all of the HLLAPI functions executed successfully or one of them failed (remaining functions in the SEND MULTIPLE call are not processed when an error code is returned).

For complete details on the parameter values, as well as usage and restrictions, see SEND MULTIPLE in chapter 5 of this manual.

Writing Portable Applications

The CLEO implementation of HLLAPI is designed to be compatible with the OS/2 HLLAPI implementation provided by the DCA/Microsoft Comm Server and Comm Workstation and with IBM OS/2 Extended Edition EHLLAPI.

However, the following points should be noted:

- ◆ The formats of parameter strings on the FunctionsSEND FILE and FunctionsRECEIVE FILE functions will differ between UNIX and OS/2, because of differences in syntax for the different file systems.
- ◆ For functions which return formatted data strings, such as FunctionsQUERY SYSTEM, other implementations may return values other than those listed in this manual, or make use of fields which are shown as reserved in this manual.
- ◆ The header files provided with other implementations contain structure declarations for use with functions which return formatted data strings. These structures are not supported because compiler differences mean that they cannot be guaranteed to work on all UNIX systems. Applications written for other environments which use these structures will have to be rewritten for UNIX, so that they access bytes at a given offset in the string rather than using the structures.

The following guidelines are provided for writing HLLAPI applications so that they will be portable to other environments:

- ◆ Include the header file *hapi_c.h* without any pathname prefix. Use the `-I /usr/include/linkix` option on the compilation line to locate the file. This allows HLLAPI applications to be used in an environment with a different file system.
- ◆ When using the FunctionsSEND FILE and FunctionsRECEIVE FILE functions, the local files should be in the directory from which you run the HLLAPI application, so that you do not need to specify a directory path for them.
- ◆ Use the symbolic constant names for parameter values and return codes, not the numeric values shown in the file *hapi_c.h*; this ensures that the correct value will be used regardless of the way these values are stored in memory. (See *hapi_c.h* for more information.)
- ◆ When testing values in returned data strings, include a check for values other than those listed here (for example using a `default` case in a switch statement) and provide appropriate diagnostics.

Multiple Processes

You need to be aware of the following issues when designing HLLAPI applications to use multiple processes.

If a HLLAPI application issues the **fork** system call while connected to a presentation space, the parent process remains connected to the presentation space. The child process may connect to other presentation spaces not used by the parent process, but it will not be able to connect to the parent's presentation space unless the parent process explicitly disconnects from it. Child processes should perform a FunctionsRESET SYSTEM function before connecting to any presentation spaces.

Use of signals

To detect work arriving from other CLEO components, the HLLAPI library makes use of the following signals:

- ◆ SIGALRM
- ◆ SIGUSR1
- ◆ SIGUSR2

While a HLLAPI function is being processed, these signals are used by the HLLAPI library and are not available to the application.

HLLAPI uses **alarm** during the processing of the following functions:

- ◆ Pause
- ◆ Receive File
- ◆ Send File
- ◆ Wait

Applications should avoid issuing **alarm** calls prior to issuing these functions, because the operation of the function maybe disrupted if an alarm is received during function processing.

Applications must not issue any further HLLAPI function calls from within a signal catcher.

Fast Path Checklist

This Fast Path checklist offers a high-level overview of how to design a HLLAPI application. Each step is either required programming or a suggested procedure (indicated by*) to help assure a smooth-running HLLAPI application.

1. * Begin by outlining the manual steps your functions will automate. Note any screen text the user would look for, each keystroke entered and the cursor position; note also any differences in behavior at the end of the screen.
2. Ensure that you have a 3270 emulation session with a session ID that matches the session identifier used by your HLLAPI program. (If you need details, see the *3270 User's Guide*.)
3. Declare the parameters.
4. * Define application error messages. Do this up front and keep a list to maintain application consistency.
5. Initialize the function parameters. Be sure to verify data string length.
6. Call HLLAPI, passing all parameters in the correct order.
7. Check the return codes for each function call.
8. Compile and link the application.

4. Testing and Troubleshooting

This chapter provides information on testing and troubleshooting application programs that have been written using the HLLAPI programming interface.

- ◆ Compiling and linking HLLAPI applications
- ◆ Running HLLAPI applications
- ◆ Tracing HLLAPI calls
- ◆ Solving common HLLAPI problems

Compiling and Linking a HLLAPI Application

Before you can test a HLLAPI application, the C program source code must be properly compiled and linked. The following is a sample command that uses the `cc` compiler to compile and link a HLLAPI program:

```
cc -I/usr/include/ linkix -o program program.c -lcom
```

The `-I` option indicates the include path for the header file `hapi_c.h`, in this case `/usr/include/linkix`.

The `-o` option gives the name of the output executable file, in this case *program*.

program.c is the name of the C program source file.

The `-l` (ell) option indicates the name of the library to be used; in the example above, the HLLAPI library file `/usr/lib/libcom.a`.

Compiling Under AIX

If you'll be testing our HLLAPI sample programs under AIX 3.2 or later, you should use one of the following samples as a guide:

```
cc -qlanglvl=ansi -I/usr/include/linkix -o program program.c -lcom
```

or

```
xlc -I/usr/include/linkix -o program program.c -lcom
```

Using the `-q` option with `cc` tells the compiler to set the language level to "ansi" during compilation (`cc`'s default language level is "extended"). This option is required when compiling our HLLAPI sample programs with `cc`.

Alternatively, you can use `xlc` which uses "ansi" as the default language level.

Running a HLLAPI Application

To running a HLLAPI application, you need a 3270 display session with a session ID that matches each HLLAPI session identifier specified in the application. If necessary, modify your 3270 style file to include these session IDs or start 3270 emulation with the -h option. For detailed information on 3270 emulation, see the *3270 User's Guide*.

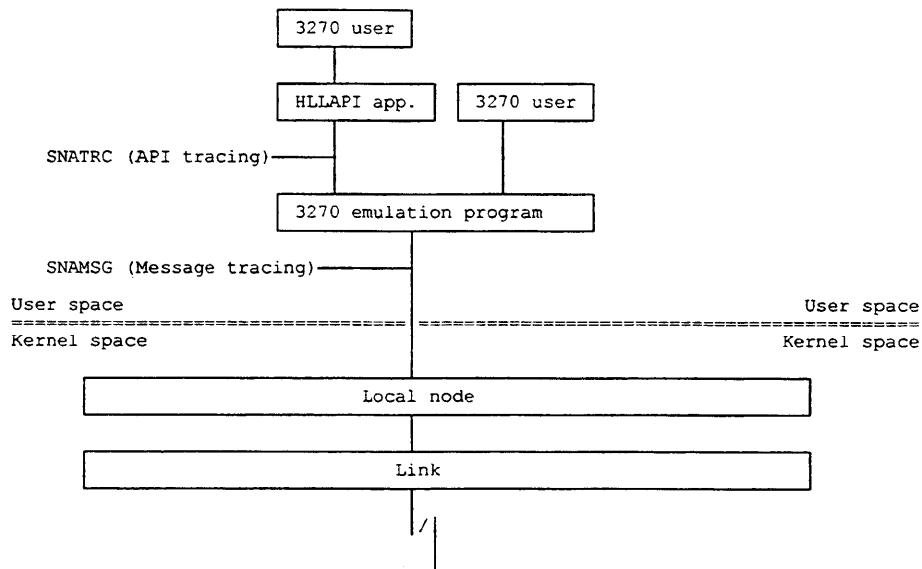
To run the application:

1. Start the 3270 emulation program with a style file which includes a session with the required session ID or with the appropriate -h option added.
2. Enable the 3270 session and ensure that it reaches SSCP status 0 that is, waiting for login at the host.
3. Perform any initial steps to get the session into the state expected by the HLLAPI application; for example, if the application does not perform an automated login to the host, then you need to do this yourself.
4. Use the TEMPEXT keystroke to move to a temporary exit shell, or login at another terminal with the same login ID as you used to run the 3270 emulation program.
5. Start the HLLAPI application by entering the executable file name you specified when compiling and linking the application. The HLLAPI application must run with the same effective user ID as the 3270 emulation program.

Tracing HLLAPI Applications

When HLLAPI applications don't operate correctly, you can use the trace facilities to help find the problem.

The following diagram shows the overall structure of the product and where tracing can be activated by the SNATRC and SNAMSG environment variables.



The SNATRC trace tracks all of the parameters supplied to the HLLAPI functions by a transaction program and all the parameters returned by the functions. The SNAMSG trace allows HLLAPI application data to be examined by tracing the 3270 emulation program under which the HLLAPI application is being used. This trace keeps track of all messages sent between the remote system and 3270.

Tracing an Application

To trace a HLLAPI transaction program, you must specify the file(s) to which the trace information will be stored and any limits on the data you wish to see. These choices are controlled with environment variables. If you're not familiar with environment variables, see the appropriate section of your UNIX documentation.

SNATRC

This environment variable specifies the trace file(s) that will be used for storing API trace data and whether or not tracing is active when 3270 emulation is started.

Its syntax is:

```
file1 [ : [file2] [:l]
```

file1 is the name of the file to which API trace data will be written. A second file name, *file2*, is optional. If two file names are specified, you must use a colon to separate them. With two files specified, up to 250,000 bytes of trace data will be stored in the first file. When the first file is filled, trace information is written to the second file. When the second file contains 250,000 bytes, data is redirected to the first file, overwriting the previous 250,000 bytes of trace information.

Using two files, tracing can be active for an extended period of time and only 500,000 bytes of disk space will be used. If you specify only one file name, there is no limit on the size of the trace file.

Warning: Trace data must always be directed to one or more text files. *Do not specify the name of a device (such as /dev/tty), or a print spooler, as a trace filename.* Doing so may cause corruption of vital UNIX system files.

A second colon can be specified to indicate that tracing should be active when 3270 emulation is started. If you specify only one file name, two colons are required to make tracing active when 3270 emulation is started. If tracing is not active when 3270 emulation is started, it can be activated from within a HLLAPI application by using Functionthe TRON option of the SET SESSION PARAMETERS function.

Testing and Troubleshooting

Examples

SNATRC=file1.trc:file2.trc

Tracing to two files, inactive at start of 3270 emulation.

SNATRC=file.trc::

Tracing to one file, active at start of 3270 emulation.

SNAMSG

This environment variable controls the tracing of message data between the remote system and the 3270 emulation program, as well as the internal control messages that are passed between components.

Its syntax is:

file1 : [file2] { : | :A }

The syntax for specifying files with this environment variable is similar to that of SNATRC. You can specify two files and no more than 500,000 bytes of disk space will be used for storing trace data, or you can specify one file and trace data will be captured without regard for trace file size. In addition to the trace file names, you must specify one of the following end parameters:

Trace all messages between the host and UNIX computer, but no internal control messages

:A Trace all messages (host/local messages and internal control messages)

Unlike SNATRC, there is no way to turn this tracing feature on or off within a HLLAPI application; it can only be started with the SNAMSG environment variable. Since all HLLAPI applications function through an active 3270 session, tracing 3270 messages may be helpful in resolving problems with a HLLAPI application.

SNATRUNC

This environment variable lets you limit the number of bytes in each entry that's written to the trace file, and it applies to all trace types. By default, the length of each internal message is limited to a maximum of 256 KB (essentially unlimited).

Set this variable to a decimal number, which represents the number of bytes to allow in each trace file entry (a typical size might be somewhere in the range of 512 to 1024). If a trace entry contains more characters than the value of SNATRUNC, the excess characters are truncated. For example, setting SNATRUNC to 512 limits tracing to 512 bytes of data per entry.

SNACTL

This environment variable can be used to suspend an application program's control over tracing. Normally, HLLAPI applications can use the TRON/TROFF options of the Functions SET SESSION PARAMETERS function to turn API tracing to the SNATRC file(s) on or off. When SNACTL is set to any non-blank string, trace control commands that are issued by an application are ignored. If tracing is on, it remains on; if tracing is off, it remains off. The return code from the SET SESSION PARAMETERS function still indicates successful completion of the function.

To cancel SNACTL, set it to a blank character.

Controlling Tracing from within an Application

CLEO's HLLAPI interface provides the ability to control tracing from within an application. This allows you to trace a specific section of a program, without having to trace the whole application. Tracing can be activated at the start of a section (where a problem is being encountered) and deactivated at the end of it.

To use this facility, the SNATRC environment variable must be set before the application program is started. Tracing is then controlled using the TRON/TROFF options of the Functions SET SESSION PARAMETERS function. The function reference in the next chapter contains more information.

Note that the SNACTL environment variable can override the ability to control tracing from within application programs. If SNACTL is set before the application program is started, any tracing control requests from within the HLLAPI application (TRON/TROFF) are ignored. If tracing is off and SNACTL is set to any non-blank string, you can prevent tracing in an application which normally uses it (TRON is ignored). In a similar manner, if tracing is on and SNACTL is set you can force tracing on an entire application for which portions are normally turned off (TROFF is ignored). In both cases, the HLLAPI function still sends back a return code that indicates the trace request completed successfully.

HLLAPI Trace File Format

This section presents an example of and notes on interpreting it.

Process ID

The 5-digit process ID of the component being traced appears at the start of each line. The process ID is followed by an indicator of the type of component being traced (EHLL indicates HLLAPI).

Parameter Data

Each HLLAPI function is traced when it is requested and again when the function returns a response. Each time a function is requested the following data is traced:

- ◆ the function name (in CAPS)
- ◆ the fact that the function is being requested
- ◆ the function's number (in parentheses)
- ◆ the value in the data_length parameter
- ◆ the value in the ps_position parameter
- ◆ the data_string parameter, if it is used by this function

When the function responds, the following data is traced:

- ◆ the function name (in CAPS)
- ◆ the fact that the function is returning a response
- ◆ the function's number (in parentheses)
- ◆ the value in the data_length parameter
- ◆ the value in the return_code parameter
- ◆ the data_string parameter, if it is used by this function

Message Data

Message data (the contents pointed to by the `data_string` parameter) is shown in both hexadecimal and character formats. The hexadecimal data is followed by the equivalent ASCII or EBCDIC characters. Note that the **tracemessages** translation to ASCII or EBCDIC is done character by character that is, if the hex value represents a printable ASCII character, it is interpreted as ASCII, otherwise it is interpreted as EBCDIC. Some EBCDIC values which are also valid ASCII values may therefore be interpreted incorrectly. For example, the value 0x40 represents a blank character in EBCDIC, but is also the @ character in ASCII; therefore it will be interpreted as @ even when it occurs within a string of EBCDIC characters.

Sample Trace Output

```

150.95 EHLL -----19:37:57.00
150.95 EHLL RESET SYSTEM request (21) , length 0, PS Position 0
150.95 EHLL -----19:37:57.00
150.95 EHLL RESET SYSTEM response, length 0, return code 0
150.95 EHLL -----19:37:57.00
150.95 EHLL QUERY SESSIONS request (10), length 64, P.S Position 0
150.95 EHLL -----19:37:57.00
150.95 EHLL QUERY SESSIONS response, length 2, return code 2
150.95 EHLL ---- Data at address 4092a0 ----
150.95 EHLL 31312020 20202020 2050D007 45534553 <11 P..ESES>
150.95 EHLL 53303120 20480000 <s01 H..
150.95 EHLL -----19:37:57.00
150.95 EHLL CONNECT PSPACE request (I), length 1, PS POSitiOn 0
150.95 EHLL ---- Data at address 4092a0 ----
150.95 EHLL 45 <E
150.95 EHLL -----19:37:58.00;
150.95 EHLL CONNECT PSPACE response(I), length 1, return code 0
150.95 EHLL ---- Data at address 4092a0 ----
150.95 EHLL 45 <E
150.95 EHLL -----19:37:58.00;
150.95 EHLL SET SESSION PAF.ANETERS request (9), length 30, PS position 0
150.95 EHLL ---- Data at address 4092a0 ----
150.95 EHLL 53524348 46524F4D 20535243 48465257 <SRCHFRM SRCHFRW>
150.95 EHLL 44205354 524C454E 20454F54 3D24 <D STRLEN EOT=$>
150.95 EHLL -----19:37:58.00
150.95 EHLL SET SESSION PARAMETERS response(9), length 30, return code 0
150.95 EHLL -----19:37:58.00
150.95 EHLL WAIT request (4), length 0, PS position 0
150.95 EHLL -----19:37:59.00
150.95 EHLL WAIT response(4), length 0, return code 0
150.95 EHLL -----19:37:59.00
150.95 EHLL SEND KEY request (3), length 42, PS position 0
150.95 EHLL ---- Data at address 4092a0 ----
150.95 EHLL 6C6F676F 6E206170 706C6964 3D74736F <logon applid=tso>
150.95 EHLL 2C6C6F67 6D6F6465 3D633234 2C646174 <.logmode=c24 ,dat>
150.95 EHLL 613D7864 636C3034 4045 <a=xdc104@E
150.95 EHLL -----19:38:08.00;
150.95 EHLL SEND KEY response(3), length 42, return code 0
150.95 EHLL -----19:38:08.00
150.95 EHLL WAIT request (4), length 0, PS position 0
150.95 EHLL -----19:38:11.00
150.95 EHLL WAIT response(4), length 0, return code 0
150.95 EHLL -----19:38:11.00
150.95 EHLL SEARCH FIELD request (30), length 8, PS Position 23
150.95 EHLL ---- Data at address 4092a0 ----
150.95 EHLL 50415353 574F5244 < PASSWORD
150.95 EHLL -----19:38:11.00;
150.95 EHLL SEARCH FIELD response, length 0, return code 24

```


5. HLLAPI Functions

This chapter defines the HLLAPI functions. Each definition contains the following information:

- ◆ Purpose
- ◆ Prior Functions
- ◆ Request Parameters
- ◆ Returned Parameters
- ◆ Whether this function is affected by the SET SESSION PARAMETERS options
- ◆ Usage and Restrictions

Note on Data String Formats

Several HLLAPI functions use a character string parameter for passing request and return data. For compatibility with IBM documentation, the descriptions in this chapter refer to the first position of the string as 1. Because C numbers the first position of string arrays such as this 0, you will need to take this difference into account when accessing individual positions in the string from your C application. See **Using HLLAPI Functions in C Programs**, in chapter 3, for more information.

Function Information

This section explains the information that's provided in each function description in this chapter.

Symbolic Function Name and Function Number

Each HLLAPI function has a symbolic name and a unique function number. The symbolic names are defined in the header file *hapi_c.h*, along with their corresponding function number. The function numbers (shown in decimal format throughout this manual) are used by the HLLAPI software. While both the symbolic name and the function number are shown under the **Request Parameters** for each function, you should always use the symbolic name rather than the number when calling the function from your application to ensure portability between operating systems. If, for some reason, an invalid HLLAPI function code is specified in the HLLAPI function request, the HARC_UNSUPPORTED return code is returned.

A list of HLLAPI functions with symbolic names and function numbers can be found under the **Alphabetical Listing** that appears later.

Prior Functions

Some functions cannot execute correctly unless one or more other functions have already been called. This section lists all functions that must be called before the function being described should be called.

Request Parameters

When requesting a function from HLLAPI, your application must pass four parameters. The names you use for these parameters doesn't matter, but your program must pass the connect types in the correct order:

1st parameter:	pointer to an integer
2nd parameter:	pointer to a character string or to an individual character
3rd parameter:	pointer to an integer
4th parameter :	pointer to an integer

The parameters are passed by means of pointers, since HLLAPI often returns values in the same parameters.

The names given to the parameters in the function descriptions vary according to the particular function being described, since the same parameter may be used for different purposes by different functions. These names are not variable names, but are descriptions of the parameters.

Parameter names that are often used in the function descriptions are:

- ◆ func_number
- ◆ data_string
- ◆ data_length
- ◆ ps_position or return_code

Integer parameters which are not used by the function should contain a pointer to zero and character parameters should contain a pointer to a null string.

Returned Parameters

Parameters returned to your application by the functions are:

- ◆ func_number
- ◆ data_string
- ◆ data_length
- ◆ return_code

If a parameter is not used by a function, its value remains unchanged.

Most functions place a return code in the fourth parameter (e.g., HARC_SUCCESS if the function completed successfully, HARC_SYSTEM_ERROR if a system error was encountered, HARC_UNSUPPORTED if an invalid function code is specified and so on). The possible return codes for each function are listed, along with an explanation of the probable condition causing the return code. For portability

HLLAPI Functions

between operating systems, you should always use the mnemonic constant (name) rather than the numeric value of the return code.

Affected by SET SESSION PARAMETERS?

Many functions are affected by options of the FunctionsSET SESSION PARAMETERS function. This section indicates which options, if any, affect this function. For complete information on the options, see the listing for SET SESSION PARAMETERS.

Usage and Restrictions

Any special considerations are explained in this section, along with explanations of how to use the function with other HLLAPI functions in your application.

Additional Information

Additional information about the function is also provided, when necessary.

Alphabetical Listing

The following table lists the HLLAPI functions in alphabetical order, giving the symbolic name and the number for each function.

Function	Symbolic name	Number
CONNECT PRESENTATION SPACE	HA_CONNECT_PS	1
CONVERT POSITION OR ROWCOL	HA_CONVERT_POS_ROW_COL	99
COPY FIELD TO STRING	HA_COPY_FIELD_TO_STR	34
COPY OIA	HA_COPY_OIA	13
COPY PRESENTATION SPACE	HA_coPY_Ps	5
COPY PRESENTATION SPACE TO STRING	HA_COPY_PS_TO_STR	8
COPY STRING TO FIELD	HA_COPY_STR_TO_FIELD	33
COPY STRING TO PRESENTATION SPACE	HA_COPY_STR_TO_PS	15
DISCONNECT PRESENTATION SPACED	HA_DISCONNECT_PS	2
DISPLAY ATTRIBUTE	HA_DISPLAY_A'TTR113 UTE	194
DISPLAY OFF	HA_DISPLAY_OFF	197
DISPLAY ON	HA_DISPLAY_ON	196
DISPLAY TEXT	HA_DISPLAY_TEXT	195
FIND FIELD LENGTH	HA_FIND_FIELD_LEN	32
FIND FIELD POSITION	HA_FIND_FIELD_POS	31
GET KEY	HA_GET_KEY	51
NOTIFY EVENT	HA_NOTIFY_EVENT	190
PAUSE	HA_PAUSE	18
POST INTERCEPT STATUS	HA_POST_INTERCEPT_STATUS	52
QUERY CURSOR LOCTION	HAQUERY_CURSOR_LOC	7
QUERY FIELD ATTRIBUTE	HA_QUERY_FIELD_A'TTR	14

HLLAPI Functions

Function	Symbolic name	Number
QUERY HOST UPDATE	HA_QUERY_HOST_UPDATE	24
QUERY SESSION STATUS	HA_QUERY_SESSION_STATUS	22
QUERY SESSION	HA_QUERY_SESSIONS	10
QUERY SYSTEM	HA_QUERY_SYSTEM	20
RECEIVE FILE	HA_RECEIVE_FILE	91
RELEASE	HA_RELEASE	12
RESERVE	HA_RESERVE	11
RESET SYSTEM	HA_RESET_SYSTEM	21
SEARCH FIELD	HA_SEARCH_FIELD	30
SEARCH PRESENTATION SPACE	HA_SEARCH_PS	6
SEND FILE	HA_SEND_FILE	90
SEND KEY	HA_SENDKEY	3
SEND MULTIPLE	HA_SENT_MULTIPLE	300
SET CURSORS	HA_SET_CURSOR	40
SET SESSION PARAMETERS	HA_SET_SESSION_PARMS	9
START HOST NOTIFICATION	HA_START_HOST_NOTIFY	23
START KEYSTROKE INTERCEPT	HA_START_KEY_INTERCEPT	50
STOP HOST NOTIFICATION	HA_STOP_HOST_NOTTFY	25
STOP KEYSTROKE INTERCEPT	HA_STOP_KEY_INTERCEPT	53
STORAGE MANAGER	HA_STORAGE_MGR	17
WAIT	HA_WAIT	4

Functional Listing

The following tables list the HLLAPI functions grouped by functional category.

Initialize/Reset Functions	Symbolic name	Number
CONNECT PRESENTATION SPACE	HA_CONNECT_PS	1
DISCONNECT PRESENTATION SPACE	HA_DISCONNECT_PS	2
QUERY SESSION STATUS	HA_QUERY_SESSION_STATUS	22
QUERY SESSION	HA_QUERY_SESSIONS	10
RESET SYSTEM	HA_RESET_SYSTEM	21
SET SESSION PARAMETERS	HA_SET_SESSION_PARMs	9

Interaction Functions	Symbolic name	Number
RELEASE	HA_RELEASE	12
RESERVE	HA_RESERVE	11

Extraction Functions	Symbolic name	Number
COPY OIA	HA_COPY_OIA	13
COPY FIELD TO STRING	HA_COPY_FIELD_TO_STR	34
COPY PRESENTATION SPACE	HA_COPY_PS	5
COPY PRESENTATION SPACE TO STRING	HA_COPY_PS_TO_STR	8

HLLAPI Functions

Orientation Functions	Symbolic name	Number
CONVERT POSITION OR ROWCOL	HA_CONVERT_POS_ROW_COL	99
FIND FIELD LENGTH	HA_FIND_FIELD_LEN	32
FIND FIELD POSITION	HA_FIND_FIELD_POS	31
QUERY CURSOR LOCATION	HA_QUERY_CURSOR_LOC	7
QUERY FIELD ATTRIBUTE	HA_QUERY_FIELD_ATTR	14
SEARCH FIELD	HA_SEARCH_FIELD	30
SEARCH PRESENTATION SPACE	HA_SEARCH_PS	6

Monitoring Functions	Symbolic name	Number
NOTIFY EVENT	HA_NOTIFY_EVENT	190
PAUSE	HA_PAUSE	18
QUERY HOST UPDATE	HA_QUERY_HOST_UPDATE	24
START HOST NOTIFICATION	HA_START_HOST_NOTIFY	23
STOP HOST NOTIFICATION	HA_STOP_HOST_NOTIFY	25
WAIT	HA_WAIT	4

Sending Functions	Symbolic name	Number
COPY STRING TO FIELD	HA_COPY_STR_TO_FIELD	33
COPY STRING TO PRESENTATION SPACE	HA_COPY_STR_TO_PS	15
SEND MULTIPLE	HA_SEND_MULTIPLE	300
SEND KEYS	HA_SENDKEY	3
SET CURSOR	HA_SET_CURSOR	40

HLLAPI Functions

Storage Management Function	Symbolic name	Number
STORAGE MANAGER	HA_STORAGE_MGR	17

File Send/Receive Functions	Symbolic name	Number
RECEIVE FILE	HA_RECEIVE_FILE	91
SEND FILE	HA_SEND_FILE	90

Keystroke Intercept Functions	Symbolic name	Number
GET KEY	HA_GET_KEY	51
POST INTERCEPT STATUS	HA_POST_INTERCEPT_STATUS	52
START KEYSTROKE INTERCEPT	HA_START_KEY_INTERCEPT	50
STOP KEYSTROKE INTERCEPT	HA_STOP_KEY_INTERCEPT	53

Display Functions	Symbolic name	Number
DISPLAY ATTRIBUTES	HA_DISPLAY_ATTRIBUTE	194
DISPLAY OFF	HA_DISPLAY_OFF	197
DISPLAY ON	HA_DISPLAY_ON	196
DISPLAY TEXT	HA_DISPLAY_TEXT	195

Debugging Function	Symbolic name	Number
QUERY SYSTEM	HA_QUERY_SYSTEM	20

Important Terms and Concepts

AID key

Attention identification key. A 3270 control key, such as PF1, CLEAR or ENTER, that causes data to be sent from a device to the mainframe.

Attribute Byte

On a 3270 display, a byte that specifies either the characteristics of a field (such as numeric or protected) or the properties of a character or field (such as blinking or reverse video).

Auto-skip

A characteristic of a 3270 protected numeric field that causes the cursor to skip automatically to the next unprotected field when the current field is completed.

Code Page

A table that defines the mapping between graphical characters and 8-bit numbers in the range 0-255. Code pages are typically based on the EBCDIC or ASCII character sets and vary slightly from country to country.

Display Session

A 3270 emulation session between a local computer and a host. An LU type 2 session that emulates a 3278 or 3279 display. Also called host display session.

EAB

Extended attribute byte. A byte whose bits define the extended attributes of a field.

ECA

Extended character attribute. An EAB that defines highlighting, color and the character set for individual characters.

EFA

Extended field attribute. An EAB that defines highlighting, color and the character set for a field unless overridden by an ECA.

Enabled Session

The state in which a 3270 emulation session is associated with an LU. If the host connection is active, the session may be used.

Extended Attributes

Visual features provided by a terminal: blinking, reverse video, underline and, on a color monitor, seven-color character display.

IND\$FILE

IBM's file transfer program. It allows files to be transferred between a local computer and the host (in three environments: CICS, VM/CMS and MVS/TSO).

Least Significant Bit

The lowest bit of an 8-bit binary number; corresponds to 2^0 (2 to the power zero).

Most Significant Bit

The highest bit of an 8-bit binary number; corresponds to 2^7 (2 to the power 7).

Operator Information Area (OIA) or Status Line

The bottom line on a 3270 emulation display. It is reserved for system messages and information about the current 3270 session.

Protected Field

A field on a 3270 screen or in a presentation space into which no data can be entered by the operator. The field is restricted by the value in its attribute byte.

Presentation Space (PS) identifier

A one-character name that identifies a presentation space; corresponds to the session ID (see below).

Session ID

This is the one-character session ID defined in the 3270 style file. Session IDs may range from "A" to "Z" or from 0x01 to 0x255 (depending on whether or not you are using the extended ID addressing capabilities). See chapter 2 (**3270 Emulation**) and chapter 5 (**Customizing 3270 Emulation**) of the *3270 User's Guide* for details.

Unprotected Field

A field in a 3270 screen or presentation space into which an operator can enter data.

Connect Presentation Space

This function connects your HLLAPI application to a specific δ that space becomes the current presentation space and all communication with the host is performed through it.

Prior Functions

None.

Request Parameters

fimc.number	HA_CONNECT_PS (1)
data_string	Session ID.
data_length	Not used.
ps_position	Not used.

Returned Parameters

func_number	1
data_string	Not used.
data_length	Not used.
return_code	HARC_SUCCESS –Function was successful; presentation space is unlocked and ready for input. HARC_INVALID_PS –Specified an invalid session ID, or the session with this ID is not active. HARC_BUSY -Connect was successful, but presentation space is busy.

HARC_LOCKED -Connect was successful, but presentation space is locked. Input is inhibited.

HARC_SYSTEM_ERROR -System error occurred.

HARC_UNAVAILABLE -The presentation space is unavailable. It is already in use.

Affected by SETT SESSION PARAMETERS?

The following options affect this function:

EXTID/NOEXTID

If the EXTID option is set, any number between 0x01 and 0xFF can be used to specify the session ID; NOEXTID causes session IDs to be limited to any single letter between A and Z.

Usage and Restrictions

- ◆ You can connect to multiple presentation spaces; the most recently connected presentation space is the current presentation space. You can return to another previously connected presentation space by issuing another connect to it.
- ◆ Once you are connected to a presentation space, no other applications can be connected to it, until you issue either FunctionsDISCONNECT PRESENTATION SPace (while connected to the presentation space) or FunctionsRESET SYSTEM (which disconnects all connected presentation spaces).
- ◆ If you are connected to a presentation space, and you issue a connect for another presentation space, the first presentation space remains “owned” by the HLLAPI application (whether or not the second connect is successful). In order to access the first presentation space, you have to issue another connect to it. In order to free the first presentation space for use by other applications, you have to issue another connect to it and then disconnector issue RESET SYSTEM (which frees all connected presentation spaces).

HLLAPI Functions: CONNECT PRESENTATION SPACE

- ◆ Your program must call this function before requesting any of the following:

Function	Number
FunctionsuCOPY FIELD TO STRING	34
FunctionsCOPY OIA	13
FunctionsCOPY PRESENTATION SPACE	5
FunctionsCOPY PRESENTATION SPACE TO STRING	8
FunctionsCOPY STRING TO FIELD	33
FunctionsCOPY STRING TO PRESENTATION SPACE	15
FunctionsDISCONNECT PRESENTATION SPACE	2
FunctionsDISPLAY ATTRIBUTE	194
FunctionsDISPLAY OFF	197
FunctionsDISPLAY ON	196
FunctionsDISPLAY TEXT	195
FunctionsFIND FIELD LENGTH	32
FunctionsFIND FIELD POSITION	31
FunctionsQUERY CURSOR LOCATION	7
FunctionsQUERY FIELD ATTRIBUTE	14
FunctionsRELEASE	12
FunctionsRESERVE	11
FunctionsSEARCH FIELD	30
FunctionsSEARCH PRESENTATION SPACE	6
FunctionsSEND KEY	3
FunctionsSEND MULTIPLE	300
FunctionsSET CURSOR	40
FunctionsWAIT	4

CONVERT POSITION OR ROWCOL

This function performs one of the following actions, depending on which request parameter is passed:

- ◆ Converts the presentation space position passed in `ps_position` into row and column coordinates.
- ◆ Converts the presentation spacerow and column coordinates passed in `row_number` and `column` into a presentation space position.

This function takes the screen model and the size of the corresponding presentation space into account during the conversion.

Prior Functions

None.

Request Parameters

func_number HA_CONVERT_POS_ROW_COL (99)

data_string A two-character string containing:

Position	Description
1	Session ID.
2	Option code character: P convert position to row and column coordinates R convert row and column to position

row_number If you are converting a presentation space position into row and column coordinates, this parameter should contain a null value. If you are converting row and column coordinates into a presentation space position, this parameter contains the row number.

HLLAPI Functions: CONVERT POSITION OR ROWCOL

**ps_position
or column** If you are converting a presentation space to row and column coordinates, this is the presentation space position. If you are converting row and column coordinates into a presentation space position, this is the column number.

Returned Parameters

func_number 99

data_string Not Used.

row_number If you are converting to row and column coordinates, this parameter contains the row position. A zero indicates the row number is invalid for the presentation space.

status_code This parameter contains a status code, rather than a standard return code, that indicates whether the function was successful and, if so, the presentation space position or column.

HARC99_INVALIDJNP -Invalid presentation space position or column was specified.

HARC99_INVALID_PS -Your application specified an invalid session ID or a system error occurred.

HARC99_INVALID_CONV_OPT -The 2nd character in the data_string parameter was not P or R.

(any number > 0) -The presentation space position or column.

Affected by SET SESSION PARAMETERS?

No.

Usage and Restrictions

- ◆ This function does not move the cursor to the specified presentation space position. (Use the FunctionsSET CURSOR function to do this.)

HLLAPI Functions: CONVERT POSITION OR ROWCOL

- ◆ Since this function returns a status code instead of a standard return code, a common error-handling routine might interpret the code incorrectly. You may need to develop a special routine for this function.
- ◆ The upper-left corner (row 1, column 1) of the presentation space is position 1; the bottom right corner is the maximum presentation space value.

COPY FIELD TO STRING

This function copies all the characters in a specified field in the current presentation space to a data string in your application. Copied characters are translated to before they are stored in the data string.

Prior Functions

CONNECT PRESENTATION SPACE –Your application must be connected to a valid presentation space before calling this function.

Request Parameters

func_number HA_COPY_FIELD_TO_STR (34)

data_string An empty data string `õ` holds the information copied from the field. This string must be at least as long as the length specified in the `data_length` parameter.

data_length Number of bytes you want to copy from the specified field. If the EAB option is set in SET SESSION PARAMETERS, this value must be twice the number of character positions in the field (each character is followed by its EAB).

ps_position Position of the field in the presentation space you want to copy. The value you supply here may be any position within the target field, including the position of the attribute byte, but you should note that the copy will always begin at the first position of the field. The range for this number depends on the type of display you are using.

Returned Parameters

func_number 34

data_string Information copied from the beginning of the field identified by `ps_position`.

data_length	Not used.
return_code	HARC_SUCCESS –Function was successful; field was copied to your application. HARC_NO_CONNECTION –Application is not connected to a valid presentation space. HARC_BAD_PARM -A supplied parameter was invalid. HARC_TRUNCATION –Field in the presentation space is not the same length as the data_length supplied by your application (it is either longer or shorter). HARC_INVALID_PS_POS –Your application specified an invalid presentation space position. HARC_SYSTEM_ERROR –System error occurred. HARC_STR_NOT_FOUND_UNFM_PS –String or field was not found, or host presentation space was unformatted.

Affected by SET SESSION PARAMETERS?

The following options affect this function:

EAB/NOEAB

EAB copies each character from the presentation space followed by its extended attribute byte; NOEAB does not copy extended attribute bytes.

XLATE/NOXLATE

If the EAB option is set, XLATE causes the copied extended attribute bytes to be translated into CGA format; NOXLATE causes extended attribute bytes to be copied without translation. If the EAB option is not set, XLATE/NOXLATE is ignored.

Usage and Restrictions

- ◆ This function copies the field beginning with the byte immediately following the attribute information for the field.

HLLAPI Functions: COPY FIELD TO STRING

- ◆ This function can copy from protected and unprotected fields.
- ◆ This function stops copying when it encounters one of the following conditions:
 - ◆ End of the field is encountered
 - ◆ Number of characters specified in `data_length` are copied
 - ◆ End of the presentation space is encountered
- ◆ Use the Functions `FIND FIELD POSITIONS` function to find the position of the field in the presentation space you want to copy.
- ◆ Use the Functions `FIND FIELD LENGTH` function to find the length of the field you want to copy.
- ◆ If the EAB option has been set in, the length of the target string must be at least twice the length of the field, to allow for the EAB characters as well as the data characters. If extended attributes are not enabled for the 3270 emulation program, they will be returned as nulls (0X00).
- ◆ If the specified field extends beyond the end of the presentation space, the copy wraps to the start of the presentation space.
- ◆ This function does not add a null character to the end of the copied string. If you want to use the copied string directly in your program, use a string which is at least one byte longer than the data to be copied, and set the whole string to nulls (with `memset`) before passing it as a parameter to this function; this will ensure you have a correctly null-terminated string on return. However, note that the `data_length` parameter should be the number of bytes to be copied, not the length of the supplied string.
- ◆ If the value of `data_length` is greater than the size of the `data_string`, your application's data area can become corrupted and produce unpredictable results.

COPY OIA

This function returns the contents of the Operator Information Area (OIA) for the current presentation space to your application. The OIA corresponds to the status line information on an IBM 3270 series terminal; it performs the same function as the 3270 emulation program's status line, but presents the information differently.

Prior Functions

CONNECT PRESENTATION SPACE -Your program must be connected to a valid presentation space before calling this function.

Request Parameters

func.number	EIA_COPY_OIA(13)
data_string	An empty data string, at least 103 bytes long ò holds the OIA information.
data_length	The length of the data string; this must be at least 103.
ps_position	Not used.

Returned Parameters

func_number	13
data_string	If the function was successful (return_code HARC_SUCCESS, HARC_BUSY or HARC_LOCKED), this parameter contains the OIA (status line) information from the current presentation space. The format of the OIA string is explained under Additional Information .
data_length	Not used.

HLLAPI Functions: COPY OIA

return_code	HARC_SUCCESS –Function was successful; OIA information was returned to your application. Host presentation space was unlocked. HARC_NO_CONNECTION -Application is not connected to a valid presentation space. HARC_BAD_PARM –An invalid string length was passed. OIA information was not returned. HARC_BUSY –OIA information was returned to your program, but the host presentation space was busy. HARC_LOCKED -OIA information was returned to your program, but the keyboard is locked. HARC_SYSTEM_ERROR -System error occurred.

Affected by SET SESSION PARAMETERS?

No.

Additional Information

The following table shows the format of the 103 byte OIA string.

Byte(s)	Description
1	OIA format byte for the PC
2-81	OIA image group; a representation of the 3270 emulation program's status line
82-103	OIA Bit groups

O/A Format Byte (Byte 1)

This is set to 1 to indicate that the returned information is in the format of a 3270 OIA. (Some HLLAPI implementations use a different value to indicate data in 5250 OIA format.)

O/A Image Group (Bytes 2-81)

Bytes 2-81 of the returned data string contain an image of the OIA information for the current session, as it would appear on a 3278 or 3279 terminal screen. Note that the OIA information is always 80 bytes, even on a Model 5 screen.

The OIA information is returned to your application as hexadecimal values corresponding to symbols in the 3274 character generator ROM. Appendix A, **OIA Image Information**, explains the meaning of the hex values in the data string.

OIA Image Bit Groups (Bytes 82-103)

Bytes 82-103 of the returned data string contain bit groups that show the current settings of the indicators in the OIA for the specified session. Bit 0 is the most significant bit and bit 7 is the least significant bit.

Byte 82: On-line and Screen Ownership (Group 1)

Indicates the owner of the current session.

Bit	Description
0,1	Reserved
2	SSCP-LU session owns screen
3	LU-LU session owns screen
4	On-line
5	Subsystem ready
6,7	Reserved

Bytes 83-87: Groups 2 to 6

Reserved and not used by the function.

HLLAPI Functions: COPY OIA

Byte 88: Insert (Group 7)

Defines whether the current presentation space is in insert mode.

Bit	Description
0	Insert mode
1-7	Reserved

Bytes 89-93: Input Inhibited (Group 8)

Describes why input is inhibited in the presentation space.

Byte	Bit	Description
89	0-2	Reserved
	3	Communications check
	4	Program check
	5-7	Reserved
90	0	Reserved
	1	Terminal wait
	2	Reserved
	3	Minus function
	4	Too much entered
	5-7	Reserved
91	0-7	Reserved
92	0,1	Reserved
	2	System wait
	3-7	Reserved
93	0-7	Reserved

Bytes 94-103: Groups 9 to 18

Reserved.

COPY PRESENTATION SPACE

This function copies the entire contents of the to a string in your application. Characters are translated to values before being stored in the data string.

Prior Functions

CONNECT PRESENTATION SPACE –Your application must be connected to a valid presentation space before calling this function.

Request Parameters

func_number HA_COPY_PS (5)

data_string An empty data string, large enough to hold all the information in the current presentation space. Presentation space sizes for each display model can be found in chapter 2, **Programming in the 3270 Environment**. If the EAB option is set in SET SESSION PARAMETERS, the string must be at least twice the size of the presentation space to allow for the EAB characters.

data_length Not used.

ps_position Not used.

Returned Parameters

func_number 5

data_string Contents of the current presentation space.

data_length Not used.

HLLAPI Functions: COPY PRESENTATION SPACE

return_code HARC_SUCCESS -Function was successful; presentation space is active, keyboard is unlocked and contents of host presentation space were copied to your application.

HARC_NO_CONNECTION –Application is not connected to a valid presentation space.

HARC_BUSY -Copy was successful, but presentation space is waiting for host response.

HARC_LOCKED -Copy was successful, but keyboard is locked.

HARC_SYSTEM_ERROR –System error occurred.

Affected by SET SESSION PARAMETERS?

The following options affect this function:

ATTRB/NOATTRB

ATTRB causes field attribute bytes to be included unchanged in the returned string;

NOATTRB causes field attribute bytes to be returned as blanks.

EAB/NOEAB

EAB copies each character from the presentation space followed by its extended attribute byte; NOEAB does not copy attribute bytes.

XLATE/NOXLATE

If the EAB option is set, XLATE causes the copied extended attribute bytes to be translated into CGA format; NOXLATE causes extended attribute bytes to be copied without translation.

Usage and Restrictions

- ◆ Use this function to copy the entire presentation space. Use FunctionsCOPY PRESENTATION SPACE TO STRING when you need to copy only a portion of the presentation space.
- ◆ The predefine data string in your program must be at least the size of the presentation space from which the data is copied. If you use this function with

different 3270 screen models, you can use FunctionsQUERY SESSIONS to find the size of the current.

- ◆ If the FunctionsSet Session ParametersEAB option has been set in SET SESSION PARAMETERS, the length of the target string must be at least twice the length of the presentation space, to allow for the EAB bytes. If extended attributes are not enabled for the 3270 emulation program, they will appear in the copied string as nulls (0X00).

COPY PRESENTATION SPACE TO STRING

This function copies the contents of all or part of the current into a data string defined in your application. Copied characters are translated to values before being stored in the data string.

Prior Functions

CONNECT PRESENTATION SPACE –Your application must be connected to a valid presentation space before calling this function.

Request Parameters

func_number HA_COPY_PS_TO_STR (8)

data_string Empty data string, large enough to hold the data that will be copied from the presentation space. Presentation space sizes for each display model can be found in chapter 2, **Programming in the 3270 Environment**. If the EAB option is set in SET SESSION PARAMETERS, the string must be twice as long as the number of character positions you want to copy, to allow for the EAB characters.

data_length Number of bytes of data to copy from the presentation space (including the extended attribute bytes if the EAB option is set).

ps_position Position in the presentation space where you want the function to begin copying.

Returned Parameters

func_number 8

data_string String copied from the presentation space.

data_length Not used.

return_code HARC_SUCCESS –Function was successful; session is active and keyboard is unlocked. Contents of host presentation space were copied to your program.

 HARC_NO_CONNECTION –Your application is not connected to a valid presentation space.

 HARC_BAD_PARM -The data_length parameter was invalid (either zero or greater than the presentation space size). No data was copied.

 HARC_BUSY -Copy was successful, but presentation space is waiting for host response.

 HARC_LOCKED -Copy was successful, but keyboard is locked.

 HARC_INVALID_PS_POS –Your program specified an invalid presentation space position.

 HARC_SYSTEM_ERROR –System error occurred.

Affected by SET SESSION PARAMETERS?

The following options affect this function:

ATTRB/NOATTRB

ATTRB causes field attribute bytes to be included unchanged in the returned string; NOATTRB causes field attribute bytes to be returned as blanks.

EAB/NOEAB

EAB copies each character from the presentation space followed by its extended attribute byte; NOEAB does not copy attribute bytes.

XLATE/NOXLATE

If the EAB option is set, XLATE causes the copied extended attribute bytes to be translated into CGA format; NOXLATE causes extended attribute bytes to be copied without translation.

Usage and Restrictions

- ◆ Use this function to copy a portion of the presentation space. Use the COPY PRESENTATION SPACE function to copy the entire presentation space.
- ◆ If this function encounters the end of the presentation space while copying data, the copying stops and the return code is set to HARC_SUCCESS. If you need to copy from a field which wraps around the end of the presentation space, use the FunctionsCOPY FIELD TO STRING function instead.
- ◆ This function copies extended attributes if you have specified the EAB option of the SET SESSION PARAMETERS function. The length of the target string must be at least twice the length of the part of the presentation space being copied, to allow for the EAB characters. If extended attributes are not enabled for the 3270 emulation program, they will appear in the copied string as nulls (0x00).
- ◆ The predefined data string in your application must be large enough to hold all of the information copied from the presentation space.
- ◆ This function automatically translates the characters in the host presentation space into ASCII.
- ◆ If the FunctionsSet Session ParametersXLATE option is set, this function translates attribute bytes to CGA terminal format.

COPY STRING TO FIELD

This function copies a string of characters from your application to a specified in the current presentation space. The characters are translated from ASCII to EBCDIC before they are added to the presentation space.

Prior Functions

CONNECT PRESENTATION SPACE –Your application must be connected to a valid presentation space before calling this function.

Request Parameters

func_number HA_COPY_STR_TO_FIELD (33)

data_string Data string to copy. If you are using the STREOT option in the function, this string must contain an end-of-text marker.

data_length Length of the data string, if you are using the STRLEN option of the function. If you are using the STREOT option, this parameter is ignored.

If you are using the EAB option, this length must include the attribute bytes as well as the characters.

ps_position Position of the field in the current presentation space where the data string is to be placed. The value maybe any position within the target field. The copy always begins at the first position of the field.

Returned Parameters

func_number 33

data_string Not used.

data_length Not used.

HLLAPI Functions: COPY STRING TO FIELD

return_code	HARC_SUCCESS -Function was successful; string was copied to the presentation space.
	HARC_NO_CONNECTION –Application is not connected to a valid presentation space.
	HARC_BAD_PARM –String length of zero was specified.
	HARC_LOCKED –Target field was protected or inhibited, or illegal data (e.g., field attribute) was sent to the presentation space.
	HARC_TRUNCATION -String was copied, but data was truncated.
	HARC_INVALID_PS_POS -Invalid presentation space position was sent.
	HARC_SYSTEM_ERROR –System error occurred.
	HARC_STR_NOT_FOUND_UNFM_PS -String or field was not found, or host presentation space was unformatted.

Affected by SET SESSION PARAMETERS?

The following options affect this function:

EAB/NOEAB

If the EAB option is set, HLLAPI assumes that each printable character in the supplied string is followed by its extended attribute byte. The characters are copied to the presentation space, and the attribute bytes are ignored. If the NOEAB option is set, HLLAPI assumes that the string does not contain attribute bytes, and copies all characters from the string to the presentation space.

STRLEN/STREOT

If STRLEN is set, the string length is passed explicitly in the data_length parameter; STREOT indicates that the supplied string is terminated by an EOT character.

EOT

If the STREOT parameter is set, EOT specifies the character used as an EOT character. If EOT IS not specified, the default EOT character is a binary zero (0x00).

Usage and Restrictions

- ◆ This function copies the string to the specified field in a formatted presentation space. You cannot copy the string to an unformatted presentation space; use COPY STRING TO PRESENTATION SPACE instead.
- ◆ The string is copied to the start of the field, even if the specified PS position is not the first position in the field.
- ◆ Copying stops when one of the following is encountered:
 - ◆ End of the field
 - ◆ End-of-text character, if using FunctionsSet Session ParametersSTREOT option
 - ◆ Number of characters specified in data_length are copied, if using FunctionsSet Session ParametersSTRLEN option
- ◆ Wrapping occurs if the field spans the screen boundary.
- ◆ To include either of the 3270 keys 3270 keysDUP or 3270 keysFMARK (Field Mark) in the supplied string, include the decimal value 28 (DUP) or 30 (FMARK). Each of these 3270 keys inserts a special character into a presentation space field. Some host applications use these characters for special functions; check the documentation for your host application to determine whether these keys can be used and what their functions are.
- ◆ This function does not change the location of the cursor.
- ◆ This function can only be used to enter printable characters, not keystrokes such as ENTER. You should ensure that each ASCII character in the supplied string has a printable EBCDIC equivalent. If a character is supplied which cannot be displayed on the 3270 screen (for example []), copying will stop and the function will return HARC_LOCKED. After using this function to copy printable characters to the PS, use the SEND KEY function to send ENTER or an AID key to the host.

COPY STRING TO PRESENTATION SPACE

This function copies an string from your application to a specific location in the current presentation space. The characters are translated from ASCII to EBCDIC before they are added to the presentation space.

Prior Functions

CONNECT PRESENTATION SPACE –Your application must be connected to a valid presentation space before calling this function.

Request Parameters

func_number	HA_COPY_STR_TO_PS (15)
data_string	ASCII string to copy. If the STREOT option is set in, this string must be terminated by an FunctionsSet Session ParametersEOT character. See Usage and Restrictions for details of sending the DUP and Field Mark characters using this function.
data_length	Length of the data string being copied to the presentation space. The maximum value for this parameter depends on the size of the presentation space for the screen model you are using. Presentation space sizes for each display model can be found in chapter 2, Programming in the 3270 Environment. This parameter is ignored if STREOT is set. If you are using the EAB option, this length must include the attribute bytes as well as the characters.
ps_position	Position in the presentation space where you want the function to place the copied data. Presentation space sizes for each display model can be found in chapter 2, Programming in the 3270 Environment .

Returned Parameters

func_number 15

HLLAPI Functions: COPY STRING TO PRESENTATION SPACE

data_string	Not used.
data_length	Not used.
return_code	HARC_SUCCESS -Function was successful; at least part of the string was copied to the presentation space. (If copying to a field-formatted presentation space, data may have been truncated at the end of the field.) HARC_NO_CONNECTION –Application is not connected to a valid presentation space. HARC_BAD_PARM –String length of zero was specified. HARC_LOCKED –Illegal data (such as a field attribute byte) was sent to the presentation space or the presentation space is protected or inhibited. None of the string was copied. HARC_TRUNCATION –String was copied, but data was truncated at the end of the screen. HARC_INVALID_PS_POS –Invalid presentation space position was passed. HARC_SYSTEM_ERROR -System error occurred.

Affected by SET SESSION PARAMETERS?

The following options affect this function:

EAB/NOEAB

If the EAB option is set, HLLAPI assumes that each printable character in the supplied string, is followed by its extended attribute byte. The characters are copied to the presentation space, and the attribute bytes are ignored. If the NOEAB option is set, HLLAPI assumes that the string does not contain attribute bytes, and copies all characters from the string to the presentation space.

STRLEN/STREOT

If STRLEN is set the string length is passed explicitly in the data_length parameter; STREOT indicates that the supplied string is terminated by an EOT character.

HLLAPI Functions: COPY STRING TO PRESENTATION SPACE

EOT

If the STREOT parameter is set, EOT specifies the character used as an EOT character. If EOT is not specified, the default EOT character is a binary zero (0x00).

Usage and Restrictions

- ◆ If the data string being copied extends beyond the end of the current presentation space, data will be truncated, and return code HARC_TRUNCATION will be returned.
- ◆ If the data string being copied extends beyond the end of the field to which it is being copied in a field-formatted presentation space, data will be truncated, but return code HARC_SUCCESS will still be returned.
- ◆ The string size cannot exceed the presentation space size. Presentation space sizes for each display model can be found in chapter 2, **Programming in the 3270 Environment**.
- ◆ The copy continues until the string size or an EOT is reached, or until end of field or end of the presentation space is reached.
- ◆ This function can only be used to enter printable characters, not keystrokes such as ENTER. You should ensure that each ASCII character in the supplied string has a printable EBCDIC equivalent. If a character is supplied which cannot be displayed on the 3270 screen (for example [), copying will stop and the function will return HARC_LOCKED. After using this function to copy printable characters to the PS, use the SEND KEY function to send ENTER or an AID key to the host.
- ◆ This function does not change the location of the cursor.
- ◆ You can use the FunctionsQUERY CURSOR LOCATION function to determine the position of the cursor in the presentation space and then use that value as the presentation space position when this function is requested.
- ◆ To include either of the 3270 keys 3270 keysDUP or 3270 keysFMARK (Field Mark) in the supplied string, include the decimal value 28 (DUP) or 30 (FMARK). Each of these 3270 keys inserts a special character into a presentation space field. Some host applications use these characters for special functions; check the documentation for your host application to determine whether these keys can be used and what their functions are.

DISCONNECT PRESENTATION SPACE

This function disconnects your application from the current. It does not end your session with the host. If you want to log off, or if you want the host to be in a certain state after this function is requested, send the relevant string of keystrokes to the host before issuing the disconnect request.

Prior Functions

CONNECT PRESENTATION SPACE -Your application must be connected to an active presentation space before calling this function.

Request Parameters

func_number HA_DISCONNECT_PS (2)

data_string Not used.

data.length Not used.

ps_position Not used.

Returned Parameters

func_number 2

data_string Not used.

data_length Not used.

return_code HARC_SUCCESS –Function was successful; your application was disconnected from the presentation space.

HARC_NO_CONNECTION –Application is not connected to a valid session.

HARC_SYSTEM_ERROR –System error occurred.

Affected by SET SESSION PARAMETERS?

No.

Usage and Restrictions

- ◆ This function disconnects the application from the current presentation space. It does not reset the session parameters to the defaults. To reset session parameters, use the FunctionsRESET SYSTEM function.
- ◆ After the application disconnects from the presentation space, functions that require a connected presentation space return an error.
- ◆ If the host screen display was suppressed (either by the “Host screens suppressed at startup” option in the 3270 customization, or by the Display Off function), Disconnect Presentation Space does not restore the screen display. It can only be restored if the application (or another application) connects to the PS and issues Display On. To leave the host screen displayed when the application disconnects from the PS, use Display On before Disconnect Presentation Space.
- ◆ If multiple presentation spaces are active, disconnecting from one presentation space does not return the application to a previously connected presentation space. To access a previously connected presentation space you must issue another connect to it. (See the **Usage and Restrictions** section for CONNECT PRESENTATION SPACE.)
- ◆ If operator input was previously disabled by a FunctionsRESERVE function request, this function releases the session and enables operator input to the session.
- ◆ HLLAPI performs a disconnect automatically when the application terminates.

DISPLAY ATTRIBUTE

This function specifies the text colors (foreground and background) to be used when writing text to the screen. This allows a HLLAPI application that is managing its own screen display (using the DISPLAY OFF and DISPLAY TEXT functions) to control the appearance of the text it writes to the screen using DISPLAY TEXT.

The screen colors that you can use with this function are the same as the colors that you can choose in the 3270 emulation program's Choose Colors for Session dialog box to represent host colors.

Prior Functions

CONNECT PRESENTATION SPACE –Your application must be connected to an active presentation space before calling this function.

DISPLAY OFF -Your application must have set the emulator into non-display mode before calling this function. (This can also be accomplished with the “Host screens suppressed at startup” session parameter, specified in the Customize Display Parameters dialog ò see the *3270 User's Guide* for details.)

Request Parameters

func_number HA_DISPLAY_ATTRIBUTE (194)

data_string A two-byte string in the following format:

Byte	Description
1	Foreground color
2	Background color

HLLAPI Functions: DISPLAY ATTRIBUTE

Each byte must be set to one of the following values. Note that the background and foreground colors must not be the same.

HLL_BLACK

HLL_BLUE

HLL_RED

HLL_PINK

HLL_GREEN

HLL_TURQ (turquoise)

HLL_YELLOW

HLL_WHITE

The following values indicate intense colors, and are valid only for the foreground color:

HLL_INT_BLUE

HLL_INT_RED

HLL_INT_PINK

HLL_INT_GREEN

HLL_INT_TURQ

HLL_INT_YELLOW

HLL_INT_WHITE

If the 3270 emulation program does not use color, HLL_BLACK is displayed as black, all intense colors are displayed as white, and all non-intense colors are displayed as gray.

data_length Not used.

ps_position Not used.

Returned Parameters

func_number 194

data_string Not used.

data_length Not used.

return_code HARC_SUCCESS -Function was successful; attributes are set for subsequent display of text using the DISPLAY TEXT function.

HARC_NO_CONNECTION -Application is not connected to a valid session.

HARC_BAD_PARM –Application specified an invalid value in data_string.

HARC_SYSTEM_ERROR –System error occurred.

HARC_DISPLAY_MODE –Emulator is in display host screen mode.

Affected by SET SESSION PARAMETERS?

No.

Usage and Restrictions

- ◆ This function only affects data written to the screen using DISPLAY TEXT. It does not affect data copied to the host screen.
- ◆ When DISPLAY OFF is issued, or when the session is started with the host screen display suppressed, the initial colors are non-intense white on black (displayed as gray on black if the 3270 emulation program does not use color).
- ◆ This function affects only the display for the current presentation space. If you connect to more than one presentation space, the screen colors used for DISPLAY TEXT are the most recent colors for that display, regardless of any DISPLAY ATTRIBUTE calls for other presentation space displays.
- ◆ The colors set by the DISPLAY ATTRIBUTE function remains in effect for all subsequent DISPLAY TEXT calls for the current presentation space, until one of the following occurs:
 - ◆ The application uses DISPLAY ATTRIBUTE again to specify a different color.
 - ◆ The application uses DISPLAY ON to restore the host screen, which disables DISPLAY ATTRIBUTES and DISPLAY TEXT.
- ◆ Note that setting both the foreground and background to non-intense colors is valid if the 3270 emulation program uses color, but not if it does not use colors (all non-intense colors are displayed as gray, so the background and foreground colors would be the same). To ensure that the HLLAPI program can run with a 3270 emulation program that does not use color, avoid setting both foreground and background to non-intense colors.

DISPLAY OFF

This function clears the current screen and suppresses all further display of host screen data until a DISPLAY ON is issued. This allows a HLLAPI application to provide its own screen display (using the DISPLAY TEXT and DISPLAY ATTRIBUTE functions) instead of displaying the host screen. All other HLLAPI functions (such as copying data to and from the host screen, or waiting for host updates) can still be used, but the updated host screen is not displayed until the HLLAPI application uses the DISPLAY ON function (which restores display of the host screen).

Prior Functions

CONNECT PRESENTATION SPACE –Your application must be connected to an active presentation space before calling this function.

Request Parameters

func_number	HA_DISPLAY_OFF (197)
data_string	Not used.
data_length	Not used.
ps_position	Not used.

Returned Parameters

func_number	197
data_string	Not used.
data_length	Not used.
return_code	HARC_SUCCESS –Function was successful. The emulator is now in non-display mode and can use the DISPLAY TEXT and DISPLAY ATTRIBUTE functions to control the screen display.

HARC_NO_CONNECTION –Application is not connected to a valid session.

HARC_SYSTEM_ERROR –System error occurred.

HARC_DISPLAY_MODE –Emulator is already in non-display mode. The application previously set the emulator into non-display mode or the “Host screens suppressed at startup” session parameter in the Customize Display Parameters dialog was checked ò see the *3270 User’s Guide* for details.

Affected by SET SESSION PARAMETERS?

No.

Usage and Restrictions

- ◆ When the emulator is in non-display mode, the application may use the DISPLAY ATTRIBUTE and DISPLAY TEXT functions to control its own screen display. To redisplay the host screen, use the DISPLAY ON function.
- ◆ When this function is initially called, the display attribute is set to the default value. To use a different attribute, use the DISPLAY ATTRIBUTE function.
- ◆ This function can be used in combination with START KEYSTROKE INTERCEPT which enables the application to intercept user-entered keystrokes.
- ◆ Data copied to the host screen using COPY STRING TO FIELD, COPY STRING TO PS or SEND KEY, as well as all data entered by the user that is not intercepted by the HLLAPI application, is written to the host screen even though it is not displayed.
- ◆ The DISCONNECT PRESENTATION SPACE and RESET SYSTEM functions both disconnect the application from the presentation space, but do not restore the host screen to the display. The host screen can only be restored if the application (or another application) connects to the presentation space and issues DISPLAY ON. To leave the host screen displayed when the application disconnects from the presentation space, use DISPLAY ON before DISCONNECT PRESENTATION SPACE or RESET SYSTEM.

HLLAPI Functions: DISPLAY OFF

- ◆ There is also a style file option that allows a session to start in non-display mode. See **Customizing Display Sessions** in chapter 5 of the *3270 User's Guide* for details.

DISPLAY ON

This function redisplay the current host screen and enables the display of all subsequent host screen changes by the emulator.

Prior Functions

CONNECT PRESENTATION SPACE –Your application must be connected to an active presentation space before calling this function.

DISPLAY OFF –Your application must have set the emulator into non-display mode before calling this function. (This can also be accomplished with the “Host screens suppressed at startup” session parameter, specified in the Customize Display Parameters dialog ò see the *3270 User’s Guide* for details.)

Request Parameters

func_number HA_DISPLAY_ON (196)

data_string Not used.

data_length Not used.

ps_position Not used.

Returned Parameters

func_number 196

data_string Not used.

data_length Not used.

return_code HARC_SUCCESS –Function was successful; host screen restored and display mode is on.

HARC_NO_CONNECTION –Application is not connected to a valid session.

HLLAPI Functions: DISPLAY ON

HARC_SYSTEM_ERROR –System error occurred.

HARC_DISPLAY_MODE –Emulator is already in display mode.

Affected by SET SESSION PARAMETERS?

No.

Usage and Restrictions

- When in this mode, the DISPLAY ATTRIBUTE and DISPLAY TEXT functions cannot be used. The return code HARC_DISPLAY_MODE is returned if you use these functions when display mode is on.
- This function redisplay the host screen, including all updates made to it while the host screen display was suppressed.
- The DISCONNECT PRESENTATION SPACE and RESET SYSTEM functions do not restore the host screen display. You should use this function before calling DISCONNECT PRESENTATION SPACE or RESET SYSTEM if you need to restore the host screen.

DISPLAY TEXT

This function displays a text string at a specified position on the UNIX terminal's screen. This function allows the application to write to the screen when the host screen is not displayed.

Prior Functions

CONNECT PRESENTATION SPACE -Your application must be connected to an active presentation space before calling this function.

DISPLAY OFF -Your application must have set the emulator into non-display mode before calling this function. (This can also be accomplished with the "Host screens suppressed at startup" session parameter, specified in the Customize Display Parameters dialog ò see the *3270 User's Guide* for details.)

Request Parameters

func_number	HA_DISPLAY_TEXT (195)
data_string	Text string to display. If you are using the STREOT option in the function, this string must contain an end-of-text marker.
data_length	Length of the text string, if you are using the STRLEN option of the function. If you are using the STREOT option, this parameter is ignored.
ps_position	Position on the screen where you want the function to place the text string. This screen address is determined in the same manner as an address for the presentation space.

Returned Parameters

func_number	195
data_string	Not used.
data_length	Not used.

HLLAPI Functions: DISPLAY TEXT

return_code	HARC_SUCCESS -Function was successful; text string was displayed on screen.
	HARC_NO_CONNECTION –Application is not connected to a valid session.
	HARC_TRUNCATION –Text string was copied, but data was truncated at the end of the screen.
	HARC_SYSTEM_ERROR –System error occurred.
	HARC_DISPLAY_MODE –Emulator is currently in display host screen mode.
	HARC_INVALID_PS_POS –Application specified an invalid screen position.

Affected by SET SESSION PARAMETERS?

The following options affect this function:

STRLEN/STREOT

If STRLEN is set, the string length is passed explicitly in the data_lenght parameter; STREOT indicates that the supplied string is terminated by an EOT character.

EOT

If the STREOT parameter is set, EOT specifies the character used as an EOT character. If EOT is not specified, the default EOT character is a binary zero (0x00).

Usage and Restrictions

- ◆ The color of the text displayed on the screen can be modified with the DISPLAY ATTRIBUTE function before using DISPLAY TEXT.
- ◆ If the text string extends beyond the end of the screen, data will be truncated and the return code HARC_TRUNCATION will be returned.
- ◆ The string size cannot exceed the presentation space size. Presentation space sizes for each display model can be found in chapter 2, **Programming in the 3270 Environment**.

- ◆ Text is displayed to the screen until the string size or an EOT is reached, or until the end of the screen is reached.
- ◆ After this function has completed, the cursor is left at the position immediately following the last character copied. If the text string ended at the end of the screen, the cursor is left at position 1 (the first position of the first line).
- ◆ This function can also be used to change the location of the cursor without writing any text to the screen. To do so, specify the cursor position in `ps_position` and pass a zero length string as follows:
 - ◆ Set the `data_length` parameter to zero if the `STRLEN` parameter has been specified in `SET SESSION PARAMETERS`.
 - ◆ End the `data_string` parameter with the EOT character specified in `SET SESSION PARAMETERS` if using the `STREOT` setting.
- ◆ This function does not write text to the host screen. To do this, use `COPY STRING TO FIELD`, `COPY STRING TO PRESENTATION SPACE` or `SEND KEY`.

FIND FIELD LENGTH

This function returns the length of a target in the current presentation space.

Prior Functions

CONNECT PRESENTATION SPACE -Your application must be connected to an active presentation space before calling this function.

Request Parameters

func_number HA_FIND_FIELD_LEN (32)

data_string A two character code that specifies which field you want the function to find the length of. The values are:

Code	Description
_ _ or T_	This field
P_	Previous field; either protected or unprotected
PP	Previous protected field
PU	Previous unprotected field
N_	Next field; either protected or unprotected
NP	Next protected field
NU	Next unprotected field
Note:	_ represents a space.

data_length Not used

ps_position Position of the field within the presentation space. The value may be any position within the target field.

Returned Parameters

func_number	32
data_string	Not used
data_length	The length of the specified field in the presentation space. If the return code is HARC_STR_NOT_FOUND_UNFM_PS, then a zero length indicates an unformatted presentation space; if the return code is HARC_FIELD_LEN_ZERO, then a zero length indicates a zero length field.
return_code	HARC_SUCCESS –Field length was found. HARC_NO_CONNECTION –The HLLAPI application is not connected to a valid presentation space. HARC_BAD_PARM –An incorrect parameter was specified. HARC_INVALID_PS_POS –Application specified an invalid presentation space position. HARC_SYSTEM_ERROR –System error occurred. HARC_STR_NOT_FOUND_UNFM_PS –String or field was not found, or presentation space was unformatted. HARC_FIELD_LEN_ZERO –Field length is zero bytes.

Affected by SET SESSION PARAMETERS?

No.

Usage and Restrictions

- ◆ This function returns the length of protected and unprotected fields, but only in a field-formatted host presentation space. This function stops counting characters in the field when it encounters an attribute byte.

HLLAPI Functions: FIND FIELD LENGTH

- ◆ If a field wraps the screen boundary, the length of the entire wrapped field is returned.
- ◆ A zero length may indicate either an unformatted presentation space or a field length of zero (two adjacent field attribute bytes in the presentation space). The return code (`HARC_STR_NOT_FOUND_UNFM_PS` or `HARC_FIELD_LEN_ZERO`) will indicate which of the two cases it is.
- ◆ If you use a common error-handling routine for all HLLAPI functions, you should take note of the return code `HARC_FIELD_LEN_ZERO` (field length was zero bytes), which does not necessarily indicate an error.

FIND FIELD POSITION

This function returns the first position of a in the current presentation space.

Prior Functions

CONNECT PRESENTATION SPACE –Your application must be connected to an active presentation space before calling this function.

Request Parameters

func_number HA_FIND_FIELD_POS (31)

data_string A two character code specifying the field you want the function to find. The following table lists the possible values.

Format	Description
_ _ or T_	This field
P_	Previous field; either protected or unprotected
PP	Previous protected field
PU	Previous unprotected field
N_	Next field; either protected or unprotected
NP	Next protected field
NU	Next unprotected field
Note:	_ represents a space.

data_length Not used

ps_position The position within the presentation space where you want the function to begin its search. The value maybe any position within the target field, including the field attribute byte.

HLLAPI Functions: FIND FIELD POSITION

Returned Parameters

func_number	31
data_string	Not used
data_length	The starting position of the field found in the host presentation space. If HARC_STR_NOT_FOUND_UNFM_PS is the return code, then a zero in this parameter indicates the presentation space was unformatted.
return_code	HARC_SUCCESS –Function was successful; field was found. HARC_NO_CONNECTION –Application is not connected to a valid session. HARC_BAD_PARM –Incorrect parameter was passed. HARC_INVALID_PS_POS –Invalid presentation space position was passed. HARC_SYSTEM_ERROR –System error occurred. HARC_STR_NOT_FOUND_UNFM_PS –String or field was not found, or presentation space was unformatted.

Affected by SET SESSION PARAMETERS?

No.

Usage and Restrictions

- ◆ This function can find the position of a field only in a formatted .
- ◆ The presentation space position returned by this function is the position immediately following the fields attribute byte;. If the field has zero length, this will in fact be the position of the field attribute byte of the following field.

GET KEY

This function retrieves a keystroke entered by a user on any session. You can then accept, process or reject the using the FunctionsPOST INTERCEPT STATUS function.

Prior Functions

START KEYSTROKE INTERCEPT –A keystroke intercept must be started before calling this function.

Request Parameters

func_number	HA_GET_KEY (51)
data_string	An 8-byte string. The first byte indicates the session ID of the presentation space from which to retrieve a keystroke. The remaining bytes are used for return data.
data_length	Not used.
ps_position	Not used.

Returned Parameters

func_number	51
data_string	Data string that identifies the presentation space and type of keystroke returned ð the format is explained under Additional Information .
data_length	Not used.
return_code	HARC_SUCCESS –Function was successful. HARC_NO_CONNECTION –Application is not connected to a valid presentation space. HARC_LOCKED –The keyboard was locked. This can occur if

HLLAPI Functions: GET KEY

you specify AID only under FunctionsSTART KEYSTROKE INTERCEPT, so HLLAPI tries to write non-AID keys to the presentation space. If the non-AID keys fail (e.g., because the user tried to enter data in a protected field), the keyboard will be locked, inhibiting AID keys.

HARC_NO_PRIOR_START –The START KEYSTROKE INTERCEPT function was not called prior to this function for the specified presentation space

HARC_SYSTEM_ERROR –System error occurred.

HARC_NO_KEYS_AVAIL -Requested keystrokes are not available on the input queue.

HARC_QUEUE_OVERFLOW –Keystroke overflow - some keystrokes lost.

Affected by SET SESSION PARAMETERS?

The following options affect this function:

ESC

ESC=*n* specifies the escape character used in the presentation of keyboard mnemonics. If ESC is not specified, the default ESC character is @.

EXTID/NOEXTID

If the EXTID option is set, any number between 0x01 and 0xFF can be used to specify the session ID and a value of 0 (zero) indicates the current session; NOEXTID causes session IDs to be limited to any single letter between A and Z and the current session can be identified with a blank or null.

LWAIT/NWAIT/TWAIT

For TWAIT or LWAIT, control of the process is not returned to your HLLAPI application program until a keystroke (normal or AID) is intercepted.

For NWAIT, the function returns immediately whether or not keystrokes are queued that match the option (normal or AID) specified for START KEYSTROKE INTERCEPT. The return code is set to HARC_NO_KEYS_AVAIL if the requested keystrokes are not available.

Usage and Restrictions

- After using this function to obtain keystrokes, use the FunctionsSEND KEY function to pass these keystrokes and any others that your HLLAPI application needs to the target presentation space.

Additional Information

The following table explains the data string format returned by this function.

Position	Description
1	Session ID.
2	Option code character: A ASCII M 2-byte mnemonic S 4-byte mnemonic
3-4	If position 2 is: A byte 3 is the ASCII character; byte 4 is 0x00 M bytes 3 and 4 represent a two-byte keyboard mnemonic as defined in the SEND KEY description S bytes 3 and 4 represent the first two characters of a four-byte keyboard mnemonic as defined in the SEND KEY description
5-6	If position 2 is S, bytes 5 and 6 represent the last two characters of a four-byte mnemonic. If position 2 is A or M, bytes 5 and 6 are 0x00.
7-8	These bytes are always 0x00.

HLLAPI Functions: GET KEY

Notes:

- ◆ The HLLAPI escape character's value is usually @, unless option ESC has redefined it.
- ◆ Mnemonics are either 2 bytes or 4 bytes long. Refer to the description of SEND KEY for more information on mnemonics.

Examples:

The following examples of returned data strings assume the default HLLAPI escape character @ is being used:

AS@A@R	Your HLLAPI program is interacting with presentation space A, keystrokes are being returned as four-byte mnemonics, and the keystroke being returned is Device Cancel.
BAr	The presentation space is B, returning keystrokes will be ASCII (A), and a lowercase r is the letter being returned. (Bytes 4-8 are 0x00.)
CM@6	The presentation space is C, keystrokes are returned as two-byte mnemonics, and PF6 is the keystroke being returned. (Bytes 5-8 are 0x00.)

NOTIFY EVENT

This function allows multiple sessions to communicate their status using a single message queue instead of START HOST NOTIFY, STOP HOST NOTIFY and QUERY HOST NOTIFY.

Prior Functions

SET SESSION PARAMETERS –Your application must set the option XNOTIFY before calling this function.

Request Parameters

func_number HA_NOTIFY_EVENT (190)

data_string A 10-character string formatted as follows:

Position	Description

1	Session ID.

2	Notification option code:

P	Notify when PS is updated
O	Notify when OIA is updated
B	Notify when PS or OIA is updated
T	Notify when keyboard is unlocked and ready for input
0x00	No notification

3-6	Message type to be used when messages are added to the queue (a positive integer of your choosing that's used by the receiving process for selecting messages). The application should use memcpy to copy the message type.

data_length Not used.

ps_position Message queue id for notification.

HLLAPI Functions: NOTIFY EVENT

Returned Parameters

func_number	190
data_string	Not used.
data_length	Not used.
return_code	HARC_SUCCESS –Function was successful. HARC_BAD_PARM –Parameter error. HARC_SYSTEM_ERROR –System error occurred. HARC_NVALID –The SET SESSION PARAMETERS function was not called to set XNOTIFY before calling this function. HARC_INVALID_PS –Your application specified an invalid session ID.

Affected by SET SESSION PARAMETERS?

The following option affects this function:

EXTID/NOEXTID

If the EXTID option is set, any number between 0x01 and 0xFF can be used to specify the session ID and a value of 0 (zero) indicates the current session; NOEXTID causes session IDs to be limited to any single letter between A and Z and the current session can be identified with a blank or null.

XNOTIFY/NOXNOTIFY

The XNOTIFY option must be set before this verb can be called; NOXNOTIFY (the default) prevents this verb from being used.

Usage and Restrictions

- *com3270* uses the message queue to inform the application that an event of the type requested has occurred on the session specified. The application can use the same message queue on calls with different session IDs to provide notification of

events across a range of sessions. The message type field allows the application to use the message queue of input other than from HLLAPI.

- ◆ *com3270* writes the session ID (a single character) and the event type (P, O or T) to the queue as well as the message type specified on the NOTIFY EVENT call. (Note that it is not possible for 3270 to return notification of updates to both Presentation Space and OIA. 3270 returns when the first event occurs.)
- ◆ One and only one event is notified per NOTIFY EVENT call.
- ◆ Extended notification may be deactivated by a null notification. The application may specify 0x00 as the second character of the data string. This causes any outstanding NOTIFY EVENT to be cancelled. A message is placed on the queue with a null event type to indicate that the previous NOTIFY EVENT has been cancelled, whether or not there is an outstanding event.
- ◆ If a NOTIFY EVENT requesting notification (rather than canceling notification) is issued when a NOTIFY EVENT is outstanding on this session, the following occur:
 - ◆ the first NOTIFY EVENT (if still pending) is cancelled, causing the null notification to be placed on the queue
 - ◆ the second NOTIFY EVENT becomes operative.
- ◆ The message queue is created and maintained by the application. It must be cleaned up when the application terminates.
- ◆ In the event of an error writing to the message queue, *com3270* will log an error (CE3BADQID = 1353) and switch off Extended Host Notification for this session only. (In such cases it is likely that the application has terminated abnormally.) Note that this is reported when *com3270* tries to write to the queue, not when NOTIFY EVENT is issued.
- ◆ The RESET SYSTEM call will cancel any outstanding Notify Event calls, placing a null event on the queue for each one, but will not clean up the queue.

PAUSE

This function causes your application to wait a specific amount of time for an event to occur.

Prior Functions

If you want to use the IPAUSE option (), you must issue SET SESSION PARAMETERS and FunctionsSTART HOST NOTIFICATION before you use this function; otherwise no prior functions are required.

Request Parameters

func_number HA_PAUSE (18)

data_string Not Used.

data_length Amount of time, in half-second increments, the application is to pause. For example, a setting of 120 causes a pause of 60 seconds.

ps_position Not used.

Returned Parameters

func_number 18

data_string Not used.

data_length Not used.

return_code HARC_SUCCESS –The pause duration has expired.
HARC_SYSTEM_ERROR –System error occurred.
HARC_HOST_UPDATE –A host session presentation space or OIA has been updated. (This return code can only be obtained if the option IPAUSE is set in SET SESSION PARAMETERS and START HOST NOTIFICATION has previously been issued.)

Affected by SET SESSION PARAMETERS?

The following option affects this function:

FPAUSE/IPAUSE

FPAUSE indicates that the program pauses for the entire length of time specified; IPAUSE indicates that the program pauses until the time specified has expired or until an event occurs in the host presentation space, whichever occurs first. To use IPAUSE, you must call the START HOST NOTIFICATION function before the PAUSE function.

Usage and Restrictions

- ◆ Do not use this function for the following types of tasks:
 - ◆ programs that are responsive to user events.
 - ◆ Long delays (that is, several hours).
 - ◆ Programs that require a high-resolution timer. The time interval created by this function is approximate (timings on UNIX are only accurate to 1 second). If you need greater accuracy, use another timing method.
- ◆ UNIX scheduling delays may mean that the application suspends for longer than the specified interval.
- ◆ Be careful when using long pauses even if you expect a host response to end the pause. If the host does not respond as expected, your application will pause the entire length of time specified.
- ◆ After the event has occurred on the host computer, use the FunctionsQUERY HOST UPDATE function to retrieve more information about the event. That function tells your application whether the presentation space or OIA was updated and the host session identification prior to the next use of the PAUSE function.
- ◆ Remember to set the IPAUSE session parameter when waiting for a host event. Once interrupted by a host event, the application must issue a QUERY HOST UPDATE to clear the host event. Otherwise, subsequent PAUSE will return immediately with a return code of HARC_HOST_UPDATE ð “the event has already occurred on the host.”

Because IPAUSE requires START HOST NOTIFY, you cannot use IPAUSE if the XNOTIFY session parameter has been set.

Post Intercept Status

This function informs HLLAPI whether a keystroke obtained from the FunctionsGET KEY function was accepted or rejected. his function can cause an audible signal (beep) to notify the operator that the keystrokes were rejected.

Prior Functions

FunctionsSTART KEYSTROKE INTERCEPT -A keystroke intercept must be started before calling this function.

Request Parameters

func_number HA_POST_INTERCEPT_STATUS (52)

data_string A two character string indicating which presentation space is being used and whether the keystroke was accepted or rejected. The possible values for the code are:

Position	Description
1	Session ID.
2	One of the following: A keystroke accepted R keystroke rejected

data_length Not used.

ps_position Not used.

Returned Parameters

func_number	52
data_string	Not used.
data_length	Not Used.
return_code	HARC_SUCCESS -Function was successful. HARC_INVALID_PS –An invalid presentation space was specified. HARC_BAD_PARM –An invalid option was specified. HARC_NO_PRIOR_START –START KEYSTROKE INTERCEPT was not called prior to this function. HARC_SYSTEM_ERROR –System error occurred.

Affected by SET SESSION PARAMETERES?

The following option affects this function:

EXTID/NOEXTID

If the EXTID option is set, any number between 0x01 and 0xFF can be used to specify the session ID and a value of 0 (zero) indicates the current session; NOEXTID causes session IDs to be limited to any single letter between A and Z and the current session can be identified with a blank or null.

Usage and Restrictions

- ◆ Rejecting a keystroke causes an audible beep. Accepting a keystroke does not.

QUERY CURSOR LOCATION

This function returns the position of the cursor in the current; you can use it to ensure the cursor is properly located before sending data.

Prior Functions

CONNECT PRESENTATION SPACE –Your application must be connected to an active presentation space before calling this function.

Request Parameters

func_number HA_QUERY_CURSOR_LOC (7)

data_string Not used.

data_length Not used.

ps_position Not used.

Returned Parameters

func_number 7

data_string Not used.

data_length The current cursor position.

return_code HARC_SUCCESS –Function was successful; position was found.

HARC_NO_CONNECTION –Application is not connected to a valid session.

HARC_SYSTEM_ERROR –System error occurred.

Affected by SET SESSION PARAMETERS?

No.

QUERY FIELD ATTRIBUTE

This function returns the field attribute byte of the specified in the current presentation space.

Prior Functions

CONNECT PRESENTATION SPACE –Your application must be connected to an active presentation space before calling this function.

Request Parameters

func_number HA_QUERY_FIELD_ATTR (14)

data_string Not used.

data_length Not used.

ps_position Any position in the target field in the current presentation space.

Returned Parameters

func_number 14

data_string Not used.

data_length Value of the field attribute byte for the specified field. This function does not translate the returned value. If the current presentation space is unformatted, this value is 0.

return_code HARC_SUCCESS –Query field attribute was successful.

HARC_NO_CONNECTION –Application is not connected to a valid session.

HARC_INVALID_PS_POS –Invalid presentation space position was passed.

HLLAPI Functions: QUERY FIELD ATTRIBUTE

HARC_SYSTEM_ERROR –System error occurred.

HARC_STR_NOT_FOUND_UNFM_PS –The presentation space was unformatted.

Affected by SET SESSION PARAMETERS?

No.

Usage and Restrictions

- ◆ Field attribute byte values are always 0xC0 or greater.
- ◆ This function returns only the field attribute byte; it does not take account of any extended field attribute or extended character attribute information that may override it.

Query Host Update

This function, in conjunction with the FunctionsSTART HOST NOTIFICATION function, allows your HLLAPI application to determine if the host or have been updated since the last request.

Prior Functions

START HOST NOTIFICATION –You must request notification of updates before calling this function.

Request Parameters

func_number HA_QUERY_HOST_UPDATE (24)

data_string Session ID.

data_length Not used.

ps_position Not used.

Returned Parameters

func_number 24

data_string Not used.

data_length Not used.

return_code HARC_SUCCESS -Function was successful; no updates since the last call.

HARC_INVALID –Function cannot be called if the XNOTIFY parameter is set.

HARC_INVALID_PS –Invalid presentation space was specified.

HLLAPI Functions: QUERY SESSIONS

HARC_NO_PRIOR_START -START HOST NOTIFICATION has not been requested for this presentation space.

HARC_SYSTEM_ERROR -System error occurred.

HARC_OIA_UPDATE -OIA was updated.

HARC_PS_UPDATE -Presentation space was updated.

HARC_PS_OR_OIA -OIA and presentation space were updated.

Affected by SET SESSION PARAMETERS?

The following options affect this function:

EXTID/NOEXTID

If the EXTID option is set any number between 0x01 and 0xFF can be used to specify the session ID and a value of 0 (zero) indicates the current session; NOEXTID causes session IDs to be limited to any single letter between A and Z and the current session can be identified with a blank or null.

XNOTIFY/NOXNOTIFY

If the XNOTIFY option is set, this verb cannot be called (see NOTIFY EVENT for information on using the message queue interface instead); NOXNOTIFY (the default) allows this verb to be used.

Usage and Restrictions

- ◆ Your application does not have to be connected to the presentation space which it wants to check for updates. To check a presentation space to which you are not currently connected, you specify the session ID in the request parameters.
- ◆ Up to six HLLAPI applications maybe notified of host updates on any one session.
- ◆ Return code HARC_PS_UPDATE, HARC_OIA_UPDATE or HARC_PS_OR_OIA is set when the screen is updated by any of the following:
 - ◆ The host
 - ◆ Information typed in from the keyboard

- ◆ The FunctionsCOPY STRING TO PRESENTATION SPACE, FunctionsCOPY STRING TO FIELD or FunctionsSEND KEY functions
- ◆ The FunctionsSET CURSOR function has no effect on this function; neither do keystrokes such as **TAB** and **HOME**, which move the cursor but do not affect the displayed data.
- ◆ Host updates which move the cursor but do not affect any displayed data will not be reported by this function.

QUERY SESSION STATUS

This function returns an 18 byte data string which contains information about the current and .

Prior Functions

None.

Request Parameters

func_number	HA_QUERY_SESSION_STATUS (22)
data_string	Session ID plus 17 extra bytes. The extra bytes are used to hold information returned to your application.
data_length	Length of the data string. This value must be at least 18.
ps_position	Not used.

Returned Parameters

func_number 22

data_string An 18-byte string containing:

Byte	Description
1	Session ID
2-9	Long name of the session
10	Session type. CLEO'S HLLAPI implementation returns D (for DFT host session type); other implementations may return different values.

Byte	Description
11	Session characteristics: bit 0=EAB: if 0 the session has base attributes if 1 the session has extended attributes bit 1-7=RESERVED:
12-13	Number of rows in the presentation space (binary number)
14-15	Number of columns in the presentation space (binary number)
16-17	Host code page number (expressed as a binary number)
18	Reserved
	See Using HLLAPI Functions in C Programs , in chapter 3, for details on extracting the two-byte numeric values from this string.
data_length	Not used.
return_code	HARC_SUCCESS – Function was successful. HARC_INVALID_PS – Invalid session ID was specified. HARC_BAD_PARM – Invalid string length was sent. HARC_SYSTEM_ERROR – System error occurred.

Affected by SET SESSION PARAMETERS?

The following option affects this function:

EXTID/NOEXTID

If the EXTID option is set, any number between 0x01 and 0xFF can be used to specify the session ID and a value of 0 (zero) indicates the current session;

NOEXTID causes session IDs to be limited to any single letter between A and Z and the current session can be identified with a blank or null.

Query Sessions

This function returns the number of sessions currently defined to the emulation program and a data string of session information.

Prior Functions

None.

Request Parameters

func_number	HA_QUERY_SESSIONS (10)
data_string	Empty data string used for return of session information. The string must be long enough to hold information on all active sessions. See data_length below for information on the string length required.
data_length	Length of the data string must be a multiple of 12 and should be at least 12 bytes for each active session plus 12 bytes. (The extra 12 bytes are used for the PC session string ð see Usage and Restrictions.)
ps_position	Not used.

Returned Parameters

func_number	10						
data_string	Information on all active sessions. The information for each session requires 12 bytes and is in the following format: <table><tr><th>Byte</th><th>Description</th></tr><tr><td>1</td><td>Session ID</td></tr><tr><td>2-9</td><td>Long name of the session</td></tr></table>	Byte	Description	1	Session ID	2-9	Long name of the session
Byte	Description						
1	Session ID						
2-9	Long name of the session						

Byte	Description
10	Session type: H = host P = P C See Usage and Restrictions for details of the PC string.
11-12	Host presentation space size (expressed as a binary number) See Using HLLAPI Functions in C Programs , in chapter 3, for details of extracting the two-byte numeric value of the presentation space size from this string.
data_length	Number of sessions (including the PC session – see Usage and Restrictions below).
return_code	HARC_SUCCESS - Function was successful; session information was returned to your application. HARC_BAD_PARM - String length is not a multiple of 12 or is too short to contain information on all active sessions. HARC_SYSTEM_ERROR – System error occurred.

Affected by SET SESSION PARAMETERS?

EXTID/NOEXTID

If the EXTID option is set, any number between 0X01 and 0xFF can be used to specify the session ID and a value of 0 (zero) indicates the current session; NOEXTID causes session IDs to be limited to any single letter between A and Z and the current session can be identified with a blank or null.

Usage and Restrictions

- ◆ Some other HLLAPI implementations provide an extra presentation space not associated with a host session (the PC session). This PC session is always included in the number returned in the data_length parameter, so the number is *always one more than the number of host sessions*.

HLLAPI Functions: QUERY SESSIONS

Also, the first 12 bytes of information returned in the `data_string` parameter contain the PC session string (identified with session type P). Information on the first *host* session begins at byte 13.

CLEO's HLLAPI implementation does not support the PC session, except as outlined above. The PC session information is included in the `data_string` so that the format is compatible with other implementations, but your HLLAPI application will not need to use it.

- ◆ This function returns information on all active 3270 sessions that are configured for the current user only ÷ it does not return system-wide information on all HLLAPI sessions.
- ◆ If the supplied string is not long enough for the number of active sessions, HLLAPI will return 12-byte strings for as many sessions as possible; the return code will be set to `HARC_BAD_PARM` to indicate that the information returned is not complete. The `data_length` parameter will be set to *the total number of sessions*, even if session information is not be returned on all of them.

QUERY SYSTEM

This function returns 35 bytes of system information. This data can be used by support personnel and may also assist you in identifying the cause of a HARC_SYSTEM.ERROR return code.

Prior Functions

None.

Request Parameters

func_number	HA_QUERY_SYSTEM (20)
data_string	A string of at least 35 bytes for the returned data.
data_lengthb	Not used.
ps_position	Not used.

Returned Parameters

func_number 20

data_string The format of the returned data string follows. See **Using HLLAPI Functions in C Programs**, in chapter 3, for details of extracting the two-byte numeric country code from this string. All other returned data consists of printable ASCII values.

Byte	Description

1	HLLAPI version number.

2-3	HLLAPI level number.

4-9	HLLAPI release date (month, day, year).

10-12	Reserved.

HLLAPI Functions: QUERY SESSIONS

Byte	Description
13	Hardware base: U = Unable to determine
14	X = Control program type.
15-19	Reserved.
20-23	Extended Error Code 1. Printable ASCII string representing the HLLAPI component ID and system error.
24-27	Extended Error Code 2. Printable ASCII string representing an internal fault code for the last internal system error.
28-29	Not used.
30-31	Country code for national language
32	Monitor code: U = Unknown
33-35	Reserved.
data_length	Not used.
return_code	HARC_SUCCESS - Function was successful; data string was returned. HARC_SYSTEM_ERROR – System error occurred.

Affected by SET SESSION PARAMETERS?

No.

Usage and Restrictions

- ◆ If you get return code HARC_SYSTEM_ERROR from any HLLAPI function call, issue a QUERY SYSTEM call to get information about the error code. If the 3270 emulation program is terminated while a HLLAPI application verb is waiting to complete, a code of HARC_SYSTEM_ERROR is returned. The extended error code 1 is set to 0700 to indicate this. Report error values other than this to the system administrator.
- ◆ Other implementations of HLLAPI may return different values for some of the bytes in the returned data string, or use some “reserved” or “not used” bytes.
- ◆ Your application can use this function to determine the session IDs and presentation space sizes of all currently available sessions, and then select an appropriate session to use. This avoids having to use a hardcoded session ID, which means that the application is not dependent on a particular session being available.

RECEIVE FILE

This function receives a file from a host. While the file is being received, the 3270 session is locked and other HLLAPI applications cannot connect to it. Although this function uses a 3270 display session, your application must not be connected to the presentation space for that session when you use this function.

Note that, although this function uses a 3270 display session, your application must not be connected to the presentation space for that session when you use this function.

Prior Functions

None.

Request Parameters

func_number HA_RECEIVE_FILE (91)

data_string Data string of parameters for the file transfer. This string has the same format as the command-line file transfer RECEIVE command:

CICS

[-h hexID] [path] filename [.ext] [id:]hostfilename
(options)

VM/CMS

[-h hexID] [path]filename[.ext] [id:]hostfilename
filetype [filemode] [(options)]

MVS/TSO

[-h hexID] [path] filename [.ext] [id:] 'datasetname
[(membername)]' [/password] [options]

See the *3270 User's Guide* for additional parameter information. The string should not exceed 128 characters and should be terminated with an EOT character if the STREOT option is specified in SET SESSION PARAMETERS.

Do not include the backslash(\) characters referred to in the *3270 User's Guide*. These are required for command-line file transfer only and are not valid in a HLLAPI application.

data_length	Actual length of the parameter string passed by the function if you are using the STRLEN option. If you are using the STREOT option, the length is ignored. The string length should not exceed 128 characters.
ps_position	Not used.

Returned Parameters

func_number	91
data_string	Not used.
data_length	Not used.
return_code	This parameter contains a HLLAPI return code, a file transfer return code or an operating system error code.

HLLAPI Return Codes

HARC_BAD_PARM – File was not received; data.string length was 0 or greater than 128 characters, or a parameter error occurred.

HARC_SYSTEM_ERROR - System error occurred.

File Transfer Return Codes

Return codes between 3 and 300 (except 9, system error) indicate file transfer messages (TRANSnnn). The most common file transfer return codes are listed below. (The *3270 User's Guide* contains complete information on file transfer messages.)

HARCFT_COMPLETE - File was transferred.

HARCFT_COMPLETE_REC_SEG – File was transferred with records segmented.

HLLAPI Functions: RECEIVE FILE

HARCFT_ALREADY_ACTIVE - Cannot link to host because file transfer is already active on this session; transfer canceled.

27

The return code 27 is used by different hosts to indicate different errors, which will be one of the following:

A timeout occurred while transferring the file (see TIMEOUT parameter of SET SESSION PARAMETERS function).

An error occurred in communications with the host. Retry.

The file transfer was terminated by the user.

If your host system generates INWnnnn messages instead of TRANSnnnn, HLLAPI does not fully distinguish between these messages. Any INW messages will be mapped either to TRANS098 (file transfer completed successfully) or to TRANS099 (file transfer failed). For more information, retry the file transfer from the 3270 emulation program's menu interface, where the INW message number and text can be displayed.

Operating System Error Codes

Return codes greater than 300 indicate operating system error codes. Subtract 300 from the return code to get the corresponding error code (HLLAPI places a 3 before each two digit value) and look up the error code value in the file *errno.h* (this file is specific to your particular UNIX computer).

Affected by SET SESSION PARAMETERS?

The following options affect this function (see **Usage and Restrictions**):

EOT

If the STREOT parameter is set, the EOT parameter specifies the character used as an EOT character. If EOT is not specified, the default EOT character is a binary zero (0x00).

QUIET/NOQUIET

QUIET indicates that file transfer messages are suppressed and are not written to standard output; NOQUIET enables the messages.

STRLEN/STREOT

STRLEN indicates that the string length is passed explicitly in the `data_length` parameter; STREOT indicates that the supplied string is terminated by an EOT character.

TIMEOUT

TIMEOUT=*n* indicates the length of time HLLAPI will wait for a response from the host. If there is no response in the specified time, the file transfer terminates. *n* indicates the number of 30-second periods for which HLLAPI will wait: 1 to 9 represent 30 seconds to 4.5 minutes, and the characters J to N represent 5 minutes to 7 minutes. HLLAPI generates a TRANS010 file transfer message (“the host has not responded”) every 30 seconds until the timeout period expires; these messages are written to standard output, unless you disable them using the QUIET option (see above) or the Q option in the string supplied to the file transfer.

TIMEOUT=0 indicates no timeout period; HLLAPI will wait indefinitely for the host to respond, unless you terminate the file transfer from the 3270 emulation program.

Usage and Restrictions

- The host must be ready to receive a command (for example, at the TSO READY prompt) when this function is used.
- Complete details on the parameters and options for file transfer, can be found in the *3270 User's Guide*.
- Test your file transfer command from the UNIX command line to ensure it works before embedding it in your application. (Don't forget the backslashes at the command line, and don't forget to remove them for the HLLAPI command.)
- The TIMEOUT and QUIET/NOQUIET options can be overridden by using the corresponding options in the parameter string supplied to this function.
- Before using this function, determine that your application has not issued FunctionsCONNECT PRESENTATION SPACE or FunctionsRESERVE. If connected to the host session, the function will fail with an error message indicating “host connection not established” or “session is in use.” Use the DISCONNECT PRESENTATION SPACE, RELEASE, or RESET SYSTEM function to disconnect and release the session.

RELEASE

This function unlocks the current host, locked with FunctionsRESERVE.

Prior Functions

CONNECT PRESENTATION SPACE – Your application must be connected to an active presentation space before calling this function.

RESERVE - The presentation space must have been locked with the RESERVE function, before you need to unlock with RELEASE.

Request Parameters

func_number HA_RELEASE (12)

data_string Not used.

data_length Not used.

ps_position Not used.

Returned Parameters

func_number 12

data_string Not used.

data_length Not used.

return_code HARC_SUCCESS – Function was successful; presentation space was released.

HARC_NO_CONNECTION – Application is not connected to a session.

HARC_SYSTEM_ERROR – System error occurred.

Affected by SET SESSION PARAMETERS?

No.

Usage and Restrictions

- ◆ Calling FunctionsDISCONNECT PRESENTATION SPACE or RESET SYSTEM, or exiting the application, releases a reserved session.

RESERVE

This function locks the current, preventing the terminal operator from entering data.

Prior Functions

CONNECT PRESENTATION SPACE - Your application must be connected to an active presentation space before calling this function.

Request Parameters

func_number HA_RESERVE (11)

data_string Not used.

data_length Not used.

ps_position Not used.

Returned Parameters

func_number 11

data_string Not used.

data_length Not used.

return_code HARC_SUCCESS – Function was successful; presentation space was reserved.

HARC_NO_CONNECTION – Application is not connected to a valid session.

HARC_SYSTEM_ERROR – System error occurred.

Affected by SET SESSION PARAMETERS?

No.

Usage and Restrictions

- ◆ Use this function to prevent the user from accessing the session to which your HLLAPI application is connected.
- ◆ Your HLLAPI application must be connected to a valid presentation space before using this function.
- ◆ Calling FunctionsDISCONNECT PRESENTATION SPACE or RESET SYSTEM, or exiting the application, releases a reserved session.

RESET SYSTEM

This function resets the default, stops host event notification, releases reserved host sessions and disconnects connected host presentation sessions.

Prior Functions

None.

Request Parameters

func_number HA_RESET_SYSTEM (21)

data_string Not used.

data_length Not used.

ps_position Not used.

Returned Parameters

func_number 21

data_string Not used.

data_length Not used.

return_code HARC_SUCCESS – Function was successful; system was reset.

HARC_SYSTEM_ERROR – System error occurred.

Affected by SET SESSION PARAMETERS?

No.

Usage and Restrictions

- ◆ This function does not free blocks of storage allocated by the STORAGE MANGER function. To do this, use the FREE ALL STORAGE subfunction.

- ◆ If the host screen display was suppressed (either by the “Host screens suppressed at startup” option in the 3270 customization, or by the DISPLAY OFF function), this function does not restore the screen display. It can only be restored if the application (or another application) reconnects to the PS and issues DISPLAY ON. To leave the host screen displayed, use DISPLAY ON before RESET SYSTEM.

SEARCH FIELD

This function searches the designated within the current presentation space for the occurrence of a specified string.

Prior Functions

CONNECT PRESENTATION SPACE – Your application must be connected to an active presentation space before calling this function.

Request Parameters

func_number	HA_SEARCH_FIELD (30)
data_string	String within the presentation space that you want this function to search for. If your program uses the FunctionsSet Session ParametersSTREOT option in the SET SESSION PARAMETERS function, the end-of-text marker must be the last character in the string.
data_length	Length of the data string if you're using the FunctionsSet Session ParametersSTRLEN option. If your program uses the STREOT option, this parameter is ignored.
ps_position	Designates the position of the field within the presentation space. This can be any position within the field.

Returned Parameters

func_number	30
data_string	Not used.
data_length	The presentation space position where the string was found; 0 if the string was not found.
return_code	HARC_SUCCESS – Function was successful; string was found.

HARC_NO_CONNECTION – Application is not connected to a valid presentation space.

HARC_BAD_PARM – Parameter string length was either zero or greater than the presentation space size.

HARC_INVALID_PS_POS – Invalid field position was sent to function.

HARC_SYSTEM_ERROR - System error occurred.

HARC_STR_NOT_FOUND_UNFM_PS – Function completed search; field or string was not found, or host presentation space was unformatted.

Affected by SET SESSION PARAMETERS?

The following options affect this function:

EOT

If the STREOT parameter (see below) is set, the EOT parameter specifies the character used as an EOT character. If EOT is not specified, the default EOT character is a binary zero (0x00).

SRCHALL/SRCHFROM

SRCHALL indicates that the entire field is searched; SRCHFROM indicates that the search begins at the specified presentation space position.

SRCHFRWD/SRCHBKWD

SRCHFRWD indicates that the field is searched forward, from the start or the specified PS position to the end; SRCHBKWD indicates that the field is searched backward, from end to the beginning or to the specified presentation space position.

STRLEN/STREOT

STRLEN indicates that the string length is passed explicitly in the data_length parameter; STREOT indicates that the supplied string is terminated by an EOT character.

Usage and Restrictions

- ◆ This function searches both protected and unprotected fields in a formatted host presentation space.
- ◆ This function searches until it encounters a field attribute byte in the presentation space.
- ◆ If the screen is unformatted, it will not be searched; the return code is set to HARC_STR_NOT_FOUND_UNFM_PS.
- ◆ If the specified field extends beyond the end of the presentation space, the search wraps to the start of the presentation space.
- ◆ The search string is case-sensitive.
- ◆ Different start and end positions are used with respect to the four search options.

Scope	Direction	Start position	End position
SRCHALL	SRCHBKWD	EOF	BOF
SRCHALL	SRCHFRWD	BOF	EOF
SRCHFROM	SRCHBKWD	EOF	FROMPOS
SRCHFROM	SRCHFRWD	FROMPOS	EOF

Key:

FROMPOS	From position specified in ps_position parameter
EOF	End of target field
BOF	Beginning of target field

SEARCH PRESENTATION SPACE

This function scans the current for a specified string.

Prior Functions

CONNECT PRESENTATION SPACE – Your application must be connected to an active presentation space before calling this function.

Request Parameters

func_number	HA_SEARCH_PS (6)
data_string	The string to search for. If your program is using the STREOT option in the SET SESSION PARAMETERS function, the last character in this string must be an end-of-text marker.
data_length	The length of the data string, if your program uses the STRLEN option in the SET SESSION PARAMETERS function. If your program uses the STREOT option, this parameter is ignored.
ps_position	The position where the function should begin searching for the string. (This option can be used only if the SRCHFROM option was set using the SET SESSION PARAMTERS function.)

Returned Parameters

func_number	6
data_string	Not used.
data_length	Specifies whether the function found the specified string. If the value returned in this parameter is 0, the string was not found. A value greater than 0 is the position where the string was found.
return_code	HARC_SUCCESS – Function was successful; string was found.

HLLAPI Function: SEARCH PRESENTATION SPACE

HARC_NO_CONNECTION – Application is not connected to a valid session.

HARC_BAD_PARM – Invalid parameters were passed.

HARC_INVALID_PS_POS - Invalid presentation space position was passed.

HARC_SYSTEM_ERROR – System error occurred.

HARC_STR_NOT_FOUND_UNFM_PS – Function completed its search, but string was not found.

Affected by SET SESSION PARAMETERS?

The following options affect this function:

EOT

If the STREOT parameter (see below) is set, the EOT parameter specifies the character used as an EOT character. If EOT is not specified, the default EOT character is a binary zero (0x00).

SRCHALL/SRCHFROM

SRCHALL indicates that the entire field is searched; SRCHFROM indicates that the search begins at the specified presentation space position.

SRCHFRWD/SRCHBKWD

SRCHFRWD indicates that the field is searched forward, from the start or the specified PS position to the end; SRCHBKWD indicates that the field is searched backward, from the end to the beginning or to the specified presentation space position.

STRLEN/STREOT

STRLEN indicates that the string length is passed explicitly in the data_length parameter; STREOT indicates that the supplied string is terminated by an EOT character.

Usage and Restrictions

- ◆ Use this function to confirm that specific data exists in the current presentation space. For example, if your application expects a specific prompt before sending data to the host, use this function to search for the prompt.
- ◆ Presentation space sizes for each display model can be found in chapter 2, **Programming in the 3270 Environment**.
- ◆ The search string is case-sensitive.
- ◆ Different start and end positions are used with respect to the four search options.

Scope	Direction	Start position	End position
SRCHALL	SRCHBKWD	EOS	BOS
SRCHALL	SRCHFRWD	BOS	EOS
SRCHFROM	SRCHBKWD	EOS	FROMPOS
SRCHFROM	SRCHFRWD	FROMPOS	EOS

Key:

EOS	End of screen
BOS	Beginning of screen
FROMPOS	From position specified in ps_position parameter

SEND FILE

This function sends a to a host. While the file is being sent, the session is locked and other HLLAPI applications cannot connect to it. Although this function uses a 3270 display session, your application must not be connected to the presentation space for that session when you use this function.

Note that, although this function uses a 3270 display session, your application must not be connected to the presentation space for that session when you use this function.

Prior Functions

None.

Request Parameters

func_number HA_SEND_FILE (90)

data_string Data string of parameters for the file transfer. This string has the same format as the command-line file transfer SEND command:

CICS

```
[ -h hexID] [path]filename[.ext] [id:]hostfilename  
(options [ ] comments]
```

VM/CMS

```
[ -h hexID] [path]filename[.ext] [id:]hostfilename  
filetype [ filemode] [(options]
```

MVS/TSO

```
[ -h hexID] [path] filename [ .ext] [id:] ' datasetname  
[(membername)] ' [/password] [options]
```

See the *3270 User's Guide* for additional parameter information. The string should not exceed 128 characters and should be terminated with an EOT character if the STREOT option is specified in SET SESSION PARAMETERS.

Do not include the backslash(\) characters referred to in the *3270 User's Guide*. These are required for command-line file transfer only and are not valid in a HLLAPI application.

data_length	Length of the data string passed by the function when using the STRLEN option. The data length should not exceed 128 characters. If you're using STREOT, the length is ignored.
ps_position	Not used.

Returned Parameters

func_number 90

data_string Not used.

data_length Not used.

return_code This parameter contains a HLLAPI return code, a file transfer return code or an operating system error code.

HLLAPI Return Codes

HARC_BAD_PARM – File was not sent; length was 0 or greater than 128 characters, or a parameter error occurred.

HARC_SYSTEM_ERROR - System error occurred.

File Transfer Return Codes

Return codes between 3 and 300 (except 9, system error) indicate file transfer messages (TRANSnnn). The most common file transfer return codes are:

HARCFT_COMPLETE - File was transferred.

HARCFT_COMPLETE_SEG_REC – File was transferred with records segmented.

HARCFT_ALREADY_ACTIVE – Cannot link to host because file transfer is already active on this session; transfer canceled (TRANS07 error).

HLLAPI Functions: SEND FILE

27

The return code 27 is used by different hosts to indicate different errors, which will be one of the following:

- A timeout occurred while transferring the file (see TIMEOUT parameter of SET SESSION PARAMETERS function).

- An error occurred in communications with the host. Retry.

- The file transfer was terminated by the user.

If your host system generates INWnnnn messages instead of TRANSnnnn, HLLAPI does not fully distinguish between these messages. Any INW messages will be mapped either to TRANS098 (file transfer completed successfully) or to TRANS099 (file transfer failed). For more information, retry the file transfer from the 3270 emulation program's menu interface, where the INW message number and text can be displayed.

Operating System Error Codes

Return codes greater than 300 indicate operating system error codes.

Subtract 300 from the return code to give the corresponding error code (HLLAPI places a 3 before each two digit value), then look up the error code in the file *erro.h* (this file is specific to your UNIX computer).

Affected by SSET SESSION PARAMETERS?

The following options affect this function (see **Usage and Restrictions**):

EOT

If the STREOT parameter is set, the EOT parameter specifies the character used as an EOT character. If EOT is not specified, the default EOT character is a binary zero (0x00).

QUIET/NOQUIET

QUIET indicates that file transfer messages are suppressed and are not written to standard output; NOQUIET enables the messages.

STRLEN/STREOT

STRLEN indicates that the string length is passed explicitly in the data_length parameter; STREOT indicates that the supplied string is terminated by an EOT character.

TIMEOUT

TIMEOUT=*n* indicates the length of time HLLAPI will wait for a response from the host. If there is no response in the specified time, the file transfer terminates. *n* indicates the number of 30-second periods for which HLLAPI will wait: 1 to 9 represent 30 seconds to 4.5 minutes, and the characters J to N represent 5 minutes to 7 minutes. HLLAPI generates a TRANS010 file transfer message (“the host has not responded”) every 30 seconds until the timeout period expires; these messages are written to standard output, unless you disable them using the QUIET option (see above) or the Q option in the string supplied to the file transfer.

TIMEOUT=0 indicates no timeout period; HLLAPI will wait indefinitely for the host to respond, unless you terminate the file transfer from the 3270 emulation program.

Usage and Restrictions

- ◆ The host must be ready to receive a command (for example, at the TSO READY prompt) when this function is used.
- ◆ For details of file transfer parameters and options, see the *3270 User's Guide*.
- ◆ Test your file transfer command from the UNIX command line to ensure it works before embedding it in your application. (Don't forget the backslashes at the command line, and don't forget to remove them for the HLLAPI command.)
- ◆ The TIMEOUT and QUIET/NOQUIET options specified on the SET SESSION PARAMETERS function can be overridden by using the corresponding options in the parameter string supplied to this function.
- ◆ Before using this function, determine that your application has not issued FunctionsCONNECT PRESENTATION SPACE or FunctionsRESERVE. If connected to the host session, the function will fail with an error message indicating “host connection not established” or “session is in use.”
- ◆ Use the FunctionsDISCONNECT PRESENTATION SPACE, FunctionsRELEASE or FunctionsRESET SYSTEM function to disconnect and release the session.

SEND KEY

This function sends one or more to the current presentation space: data keys (A, B, C, . ..) or attention identifier (AID) keys (ENTER or PA1).

Prior Functions

CONNECT PRESENTATION SPACE – Your application must be connected to an active presentation space before calling this function.

Request Parameters

func_number	HA_SENDKEY (3)
data_string	String of keystrokes to send to the host presentation space. This string can contain up to 255 characters (more than one character may be required to represent a keystroke). If your program is using the STREOT option in the SET SESSION PARAMETERS function, this string must end with an end-of-text (EOT) character.
data_length	Length of the data string, if the STRLEN option is being used in the SET SESSION PARAMETERS function. If the STREOT option is being used, this parameter is ignored.
ps_position	Not used.

Returned Parameters

func_number	3
data_string	Not used.
data_length	Not used.
return_code	HARC_SUCCESS – Function was successful; keystrokes sent. HARC_NO_CONNECTION – Application is not connected to a valid session.

HARC_BAD_PARM – Incorrect parameter was passed.

HARC_BUSY - Presentation space was busy; unable to send all keystrokes.

HARC_LOCKED – Input to the target presentation space was inhibited or rejected; unable to send all keystrokes.

HARC_SYSTEM_ERROR – System error occurred.

Affected by SET SESSION PARAMETERS?

The following options affect this function:

AUTORESET/NORESET

AUTORESET indicates that HLLAPI is to send the RESET key sequence before attempting to send keystrokes to the presentation space; NORESET indicates that the keystrokes are sent without first sending a RESET.

EOT

If the STREOT parameter is set, the EOT parameter specifies the character used as an EOT character. If EOT is not specified, the default EOT character is a binary zero (0x00).

ESC

ESC=*n* specifies the escape character used in the representation of keyboard mnemonics. If ESC is not specified, the default ESC character is a binary zero (0x00).

STRLEN/STREOT

STRLEN indicates that the string length is passed explicitly in the *data_length* parameter; STREOT indicates that the supplied string is terminated by an EOT character.

Usage and Restrictions

- ◆ Except for AID and a few other key mnemonics, your application cannot send keystrokes to protected fields. The following keys are AID keys: ENTER, CLEAR, SYSREQ, ATTN and all program function keys (PF1–PF24) and program access keys (PA1-PA3).
- ◆ Only numeric data can be sent to numeric fields.
- ◆ The maximum length of strings that you can pass to this function is 255 characters. To send more than 255 characters, use multiple calls.
- ◆ When performance is a consideration, use FunctionsCopy STRING TO PRESENTATION SPACE or FunctionsCOPY STRING TO FIELD to write entire strings to the presentation space. Then use this function (SEND KEY) to send an AID key to the host.
- ◆ Keystrokes are not sent to the presentation space when the session is busy (due to a FunctionsSet Session ParameterWAIT condition) or the keyboard is locked. Use the FunctionsWAIT or FunctionsPAUSE functions to determine the status of the keyboard before sending keystrokes to the presentation space. The WAIT indicator does not ordinarily appear until an AID key is sent to the presentation space.
- ◆ The following 3270 keys (represented by the EBCDIC values shown) have no equivalent key on your terminal. You can send these keystrokes to a host session by sending the appropriate ASCII value shown in the following table. Note that the value depends on whether you are using the international option on the 3270 emulation program (-i option).

For example, you can send the “cent” character by sending 0x9B (non-international) or 0xA2 (international).

3270 key	EBCDIC hex value	ASCII hex with -i/without -i
(solid vertical bar)	5E	B3/7C
S¢ (cent)	5B	9B/A2
(logical NOT)	5D	5E/AC

- ◆ You can only use this function to send a keystroke to a connected active session.
- ◆ Certain control keys must be generated using two or more ASCII characters. For specific information see the **Keyboard mnemonics** section.
- ◆ You can specify a new escape character (other than @) in SET SESSION PARAMETERS function with the FunctionSet Session Parameters ESC option.
- ◆ Do not embed AID keys within a list of keystrokes, although one maybe the last key in a list. Embedding an AID key in a string may cause the host to disregard all following keystrokes.

Additional Information

Keyboard Mnemonics

Special function keys on the 3270 keyboard are sent to the presentation space using 3270 keymnemonics. Mnemonics are preceded by the escape character. (The default is @, but it can be changed using the ESC option of the SET SESSION PARAMETERS function.) For example, to send PF6 to the presentation space, enter @6 as the string.

To send the escape character as a keystroke, include it twice. For example, if "&" is being used as the escape character, you can send the& character by specifying &&.

CLEO's HLLAPI implementation does not support the 3270 IDENT key (allows a 3270 user to specify a printer identifier). The SEND KEY function does accept the mnemonic @A@Pnn for IDENT nn (where nn is the printer identifier) without generating an error, but ignores it.

HLLAPI Functions: SEND KEY

The ASCII characters used to represent function keys are:

3270 key	Mnemonic	3270 key	Mnemonic
@	@@	Cursor Select	@A@J
Alternate Cursor	@\$	Cursor Up	@U
Attention	@A@Q	Delete Character	@D
Backspace	@<	Device Cancel	@A@R
Backtab	@B	Dup	@S@x
Backward Word	@A@z	Enter	@E
Clear	@C	Erase EOF	@F
Cursor Down	@V	Erase Input	@A@F
Cursor Left	@L	Field Mark	@S@y
Cursor Left Double	@A@L	Forward Word	@A@y
Cursor Right	@Z	Home	@0
Cursor Right Double	@A@Z	Insert Character	@I
PA1	@x	New Line	@N
PA2	@y	PF14	@e
PA3	@z	PF15	@f
PF1	@1	PF16	@g
PF2	@2	PF17	@h
PF3	@3	PF18	@i
PF4	@4	PF19	@j
PF5	@5	PF20	@k
PF6	@6	PF21	@l
PF7	@7	PF22	@m
PF8	@8	PF23	@n
PF9	@9	PF24	@o
PF10	@a	Print Screen	@P
PF11	@b	Reset	@R
PF12	@c	Sys Request	@A@H
PF13	@d	Tab	@T

SEND MULTIPLE

This function sends up to 20 HLLAPI function calls at one time from the application.

Prior Functions

None.

Request Parameters

func_number	HA_SEND_MULTIPLE (300)
data_string	A pointer to an array of multiple HLLAPI calls.
data_length	The number of HLLAPI functions in the call.
ps_position	Not used.

Returned Parameters

func_number	300
data_string	Not used.
data_length	If one of the HLLAPI functions fails, this parameter contains the number of the HLLAPI function that failed.
return_code	The code HARC_SUCCESS is returned if all of the HLLAPI functions executed successfully. If one of the HLLAPI functions failed, this parameter contains the return code from that HLLAPI function.

Affected by SET SESSION PARAMETERS?

No.

HLLAPI Functions: SEND MULTIPLE

Usage and Restrictions

- The return_code is for the complete function and indicates global failures. Each function in the array passed in the data_string will have its own return_code.
- A maximum of 20 functions are allowed per SEND MULTIPLE call.
- All data pointed to by individual HLLAPI calls must remain unchanged until all functions with the SEND MULTIPLE call are completed (or an error occurs).
- If all functions cannot be processed in one task switch due to lack of shared memory, additional task switches will be performed automatically until all functions have been processed.
- If a function returns an error code, remaining functions in the SEND MULTIPLE call will not be processed.
- If CONNECT PRESENTATION SPACE is one of the HLLAPI functions in SEND MULTIPLE, it must be the first function.
- The following HLLAPI functions are allowed in SEND MYLTIPLE:

Function	Symbolic name	Number
CONNECT PRESENTATION SPACE	HA_CONNECT_PS	1
COPY FIELD TO STRING	HA_COPY_FIELD_TO_STR	34
COPY OIA	HA_COPY_OIA	13
COPY PRESENTATION SPACE	HA_COPY_PS	5
COPY PRESENTATION SPACE TO STRING	HA_COPY_PS_TO_STR	8
COPY STRING TO FIELD	HA_COPY_STR_TO_FIELD	33
COPY STRING TO PRESENTATION SPACE	HA_COPY_STR_TO_PS	15
DISPLAY ATTRIBUTE	HA_DISPLAY_ATTRIBUTE	194
DISPLAY OFF	HA_DISPLAY_OFF	197
DISPLAY ON	HA_DISPLAY_ON	196
DISPLAY TEXT	HA_DISPLAY_TEXT	195
FIND FIELD LENGTH	HA_FIND_FIELD_LEN	32
FIND FIELD POSITION	HA_FIND_FIELD_POS	31
NOTIFY EVENT	HA_NOTIFY_EVENT	190
RELEASE	HA_RELEASE	12
RESERVE	HA_RESERVE	11

HLLAPI Functions: SEND MULTIPLE

Function	Symbolic name	Number
SEARCH FIELD	HA_SEARCH_FIELD	30
SEARCH PRESENTATION SPACE	HA_SEARCH_PS	6
SEND KEY	HA_SE.NDKEY	3
SET CURSOR	HA_SET_CURSOR	40
START HOST NOTIFICATION	HA_START_HOST_NOTIFY	23
STOP HOST NOTIFICATION	HA_STOP_HOST_NOTIFY	25

SET CURSOR

This function moves the cursor within the current.

Prior Functions

None.

Request Parameters

func_number	HA_SET_CURSOR (40)
data_string	Not used.
data_length	Not used.
ps_position	Position in the presentation space where the cursor is to be moved.

Returned Parameters

func_number	40
data_string	Not used.
data_length	Not used.
return_code	HARC_SUCCESS – Function was successful; cursor was moved. HARC_NO_CONNECTION – Application is not connected to a valid presentation space. HARC_BUSY – Host session is busy. HARC_INVALID_PS_POS – You specified a cursor location less than 1 or greater than the maximum presentation space size. HARC_SYSTEM_ERROR – System error occurred.

Affected by SET SESSION PARAMETERS?

No.

Usage and Restrictions

- ◆ The position to which the cursor is to be moved must be between 1 and the maximum number of positions for the presentation space. Presentation space sizes for each display model can be found in chapter 2, **Programming in the 3270 Environment**.

SET SESSION PARAMETERS

This function changes session parameters.

Prior Functions

None.

Request Parameters

func_number	HA_SET_SESSION_PARMS (9)
data_string	String of session parameters, separated by commas or blanks.
data_length	Length of the data string. <i>This parameter is required</i> as the STREOT option has no effect.
ps_position	Not used.

Returned Parameters

func_number	9
data_string	Not used.
data_length	Not Used.
return_code	HARC_SUCCESS – Function was successful; parameters were set to the specified values. HARC_BAD_PARM – Invalid string length was specified, or one or more invalid parameters were found in the string. HARC_SYSTEM_ERROR – System error occurred.

Usage and Restrictions

- ◆ This function affects the operation of many other HLLAPI functions. Make sure you use it correctly in your program.
- ◆ The length of the string *must* be passed in the `data_length` parameter.
- ◆ You need only specify the parameters to change; all parameters not included in the string remain at their previous values.
- ◆ If the return code is `HARC_BAD_PARM`, indicating that one or more parameters were invalid, no changes are made to any of the session options.
- ◆ Your application program should always reset the options to their default values (using `Reset System`) before terminating. This ensures that the next application program to use this presentation space will operate as expected.

Additional information

In the following sections, default options are indicated with an asterisk (*).

Attribute Copy Options

The `FunctionsSet Session ParametersATTRB` / `FunctionsSet Session ParametersNOATTRB` options specify whether or not information is copied as part of the strings that are copied to and from the presentation space.

ATTRB	Return field attributes to your application. Every field attribute byte in the copied string is returned untranslated (values are always greater than 0xC0).
*NOATTR	Return field attributes (and any other unknown values) to your application as blanks.

Functions Affected

`FunctionsCOPY PRESENTATION SPACE`

`FunctionsCOPY PRESENTATION SPACE TO STRING`

Connect Options

The FunctionsSet Session ParametersCONLOG / FunctionsSet Session ParametersCONPHYS options, which are used by other implementations of HLLAPI, are not required for CLEO's HLLAPI implementation.

Note: Field attribute bytes cannot be copied back to the host screen (or to the terminal's screen using the DISPLAY TEXT function). When copying strings from the host screen that will later be written back either to the host screen or to the terminal, you should either use NOATTRB or remove the field attribute bytes from the string before copying it to the screen.

Copy options

The FunctionsSet Session ParametersEAB / FunctionsSet Session ParametersNOEAB and FunctionsSet Session ParametersXLATE / FunctionsSet Session ParametersNOXLATE options specify whether or not extended attribute information is copied from the presentation space.

Note that extended attribute information cannot be copied to the presentation space; it can only be setup by the host. However, strings containing extended attribute bytes can be copied to the presentation space using this function; the extended attribute bytes are discarded.

EAB Copy presentation space data with extended attributes (each character is followed by its extended attribute byte). With this option you must allocate a buffer twice the size of the area to copy, because two characters are returned for every one that appears in the presentation space.

When copying to the presentation space, the supplied string is assumed to contain extended attribute bytes (as each character is copied from the string to the presentation space, the character following it is assumed to be an extended attribute byte and is discarded). The supplied string should be twice the length of the data you want to copy to the presentation space, because one character is copied to the presentation space for every two that appear in the supplied string.

***NOEAB** Extended attributes are not copied from the presentation space. When copying to the presentation space, all characters in the supplied string are taken to be data and are copied into the presentation space.

XLATE Translate EABs to CGA terminal format. If your screen hardware is CGA (Color Graphics Adapter) compatible, the data strings produced by this option can be sent directly to the terminal's screen, to produce a copy of the presentation space data complete with its host attributes.

***NOXLATE** Do not translate EABs.

Functions Affected

FunctionsCOPY FIELD TO STRING

FunctionsCOPY PRESENTATION SPACE

FunctionsCOPY PRESENTATION SPACE TO STRING

FunctionsCOPY STRING TO FIELD (EAB/NOEAB only)

FunctionsCOPY STRING TO PRESENTATION SPACE (EAB/NOEAB only)

Note: The **DISPLAY TEXT** function, which copies a text string to the terminal while the host screen display is suppressed, requires a text string without EAB characters. When copying strings from the host screen that will later be written to the terminal using **DISPLAY TEXT**, you should either use **NOEAB** or remove the EAB characters from the string before using **DISPLAY TEXT**.

End-of-Text Copy Options

The FunctionsSet Session ParametersSTRLEN / FunctionsSet Session

ParametersSTREOT options specify whether your application supplies the length of data strings being passed to the function or uses an end-of-text character. The FunctionsSet Session ParametersEOT option specifies the end-of-text character.

***STRLEN** The application provides an explicit length for all strings.

STREOT Strings end with an end-of-text character.

EOT=n Defines the end-of-text character. (This option is valid only with the STREOT option.) The default end-of-text character is an ASCII null, 0x00 (represented in C programs as "\0").

Functions Affected

FunctionsCOPY STRING TO FIELD

FunctionsCOPY STRING TO PRESENTATION SPACE

FunctionsDISPLAY TEXT

FunctionsRECEIVE FILE

FunctionsSEARCH FIELD

HLLAPI Functions: SET SESSION PARAMETERS

FunctionsSEARCH PRESENTATION SPACE
FunctionsSEND FILE
FunctionsSEND KEY

Escape Option

The FunctionsSet Session ParametersESC option lets you define the escape character used in strings of keystrokes sent to the host presentation space.

ESC=n Defines the escape character. Any ASCII character can be used as the escape character. The default is @.

Functions Affected
FunctionsGET KEY
FunctionsSEND KEY

Event Notification Options

The FunctionsSet Session ParametersXNOTIFY / FunctionsSet Session ParametersNOXNOTIFY options specify whether or not your application notifies of screen event changes via a single message queue or using standard event notification.

XNOTIFY The application will use NOTIFY EVENT to notify of screen events rather than START HOST NOTIFY, STOP HOST NOTIFY and QUERY HOST UPDATE.

***NOXNOTIFY** Use standard event notification with START HOST NOTIFY, STOP HOST NOTIFY and QUERY HOST UPDATE.

Functions Affected
FunctionsNOTIFY EVENT
FunctionsQUERY HOST UPDATE
FunctionsSTART HOST NOTIFICATION
FunctionsSTOP HOST NOTIFICATION

Pause Options

The FunctionsSet Session ParametersFPAUSE / FunctionsSet Session ParametersIPAUSE options determine how long your application will wait for an event to occur on the host when your application calls the PAUSE function.

- *FPAUSE** Pause for the full duration of time specified in the PAUSE function.
- IPAUSE** Pause until an event occurs in the host presentation space (or for the full duration of time specified in the Pause function, if no host presentation space event occurs in this time), and then return to the application. In order for this option to operate correctly, your application must call FunctionsSTART HOST NOTIFICATION before calling the FunctionsPAUSE function.

Functions Affected
FunctionsPAUSE

Quiet Options

The FunctionsSet Session ParametersQUIET / FunctionsSet Session ParametersNOQUIET options specify whether file transfer messages generated by the FunctionsSEND FILE and FunctionsRECEIVE FILE functions are sent to standard output.

Note that the option specified here can be overridden for a specific call to the SEND FILE or RECEIVE FILE function, by using the QUIET option in the parameter string supplied to that function. See the *3270 User's Guide* for more information on file transfer parameters.

- QUIET** All file transfer messages are suppressed and are not sent to standard output.
- *NOQUIET** All file transfer messages are sent to standard output.

Functions Affected
FunctionsRECEIVE FILE
FunctionsSEND FILE

Reset Options

The FunctionsSet Session ParametersAUTORESET / FunctionsSet Session ParametersNORESET options determine whether the FunctionsSEND KEY function attempts to reset all inhibited conditions on the host before sending the keystrokes.

HLLAPI Functions: SET SESSION PARAMETERS

***AUTORESET** Prefix keystrokes with a reset. The reset is sent once, before sending the first keystroke in the string, not before every keystroke.

NORESET Do not prefix keystrokes with a reset.

Functions Affected

FunctionsSEND KEY

Search Options

The FunctionsSet Session ParametersSRCHALL / FunctionsSet Session ParametersSRCHFROM and FunctionsSet Session ParametersSRCHFRWD / FunctionsSet Session ParametersSRCHBKWD options determine how functions search the or a in the presentation space for a specified string.

***SRCHALL** Search the entire presentation space or field. Your application does not have to supply a starting position for the search.

SRCHFROM Search the presentation space or field between the specified position and the end. The starting position for the search must be specified on the search function.

***SRCHFRWD** Search the presentation space or field in an ascending direction.

SRCHBKWD Search the presentation space or field in a descending direction.

Functions Affected

FunctionsSEARCH FIELD

FunctionsSEARCH PRESENTATION SPACE

Session ID Options

The FunctionsSet Session ParametersEXTID / FunctionsSet Session ParametersNOEXTID options specify how your application handles session IDs.

EXTID Any number between 0x01 and 0xFF can be used to specify the session ID and a value of 0 (zero) indicates the current session.

***NOEXTID** Causes session IDs to be limited to any single letter between A and Z and the current session can be identified with a blank or null.

Functions Affected

FunctionsCONNECT PRESENTATION SPACE
FunctionsGET KEY
FunctionsNOTIFY EVENT
FunctionsPOST INTERCEPT STATUS
FunctionsQUERY HOST UPDATE
FunctionsQUERY SESSION STATUS
FunctionsQUERY SESSIONS
FunctionsSTART HOST NOTIFICATION
FunctionsSTART KEYSTROKE INTERCEPT
FunctionsSTOP HOST NOTIFICATION
FunctionsSTOP KEYSTROKE INTERCEPT

Timeout Option

The FunctionsSet Session ParametersTIMEOUT option specifies the length of time to allow for completion of the FunctionsSEND FILE and FunctionsRECEIVE FILE functions. After the specified time, the is terminated.

Note that the option specified here can be explicitly overridden when calling the SEND FILE or RECEIVE FILE function.

TIMEOUT=n HLLAPI waits for n 30-second cycles, then terminates the file transfer if it has not yet completed. Replace the n with the number of cycles HLLAPI should wait:

1 = 30 seconds	8 = 4 minutes
2 = 1 minute	9 = 4.5 minutes
3 = 1.5 minutes	J = 5 minutes
4 = 2 minutes	K = 5.5 minutes
5 = 2.5 minutes	L = 6 minutes
6 = 3 minutes	M = 6.5 minutes
7 = 3.5 minutes	N = 7 minutes

HLLAPI generates a T'RANS010 timeout message every 30 seconds until the host responds or until the timeout period expires; these messages are sent to standard output, unless you suppress them by using the QUIET option on FunctionsSET SESSION PARAMETERS or the /0 option in the parameter string supplied to the FunctionsSEND FILE or FunctionsRECEIVE FILE function.

HLLAPI Functions: SET SESSION PARAMETERS

TIMEOUT=0 No timeout. The file transfer continues until it completes, or until it receives an interrupt signal. You can terminate it either by using the terminal's interrupt sequence (for example **CTRL + C**) or from the 3270 control interface. TRANS010 timeout messages are not generated.

Functions Affected

FunctionsSEND FILE

FunctionsRECEIVE FILE

Trace Options

The FunctionsSet Session ParametersTRON / FunctionsSet Session ParametersTROFF options control tracing of HLLAPI request parameters. Trace information is sent to the file(s) specified in the SNATRC environment variable, which must be set up before the HLLAPI application is started; if it is not set up, trace data is sent to standard output.

Note that tracing on a HLLAPI application can also be controlled using the CSV interface (the DEFINE_TRACE and COPY_TRACE_TO_FILE verbs). Refer to the *CSV Programmer's Guide* for details on these verbs.

TRON Turn trace facility on.

***TROFF** Turn trace facility off.

Functions Affected

TRON traces all HLLAPI function calls.

Wait Options

The FunctionsSet Session ParametersLWAIT / FunctionsSet Session ParametersNWAIT / FunctionsSet Session ParametersTWAIT options determine how long your application waits for a response from the host after the FunctionsWAIT or FunctionsGET KEY function is called.

***TWAIT** **WAIT:** wait up to 1 minute for the keyboard to become ready for input, then return.
 GET KEY: wait until a keystroke has been queued before returning.

LWAIT **WAIT:** wait for the keyboard to become ready, then return. Use this option with caution. Your application will not regain control of the

HLLAPI Functions: SET SESSIONS PARAMETERS

process if an error condition exists on the host that prevents the WAIT indicator from being cleared.

GET KEY: wait until a keystroke has been queued before returning.

NWAIT

WAIT: check the keyboard status and immediately return.

GET KEY: return immediately whether or not the requested keystrokes were available.

Functions Affected

FunctionsGET KEY

FunctionsWAIT

START HOST NOTIFICATION

This function allows your HLLAPI application to determine if the host and/or have been updated.

This function also allows the selected host session to end a pause started by the FunctionsPAUSE function if your application is using the FunctionsSet Session ParametersIPAUSE option in the SET SESSION PARAMETERS function.

Your application can then call the FunctionsQUERY HOST UPDATE function to determine specific information about the update.

Prior Functions

None.

Request Parameters

func_number HA_START_HOST_NOTIFY (23)

data_string A 6-byte string containing the following information:

Byte	Description

1	Session ID.

2	Request notification of updates to:

	P presentation space only

	O OIA (Operator Information Area) only

	B both the presentation space and the OIA

3-6	Reserved

data_length Not used.

ps_position Not used.

Returned Parameters

func_number	23
data_string	Not used.
data_length	Not used.
return_code	HARC_SUCCESS – Function was successful. HARCJINVALID - Function cannot be called if the XNOTTFY parameter is set. HARC_INVALID_PS – An invalid session ID was specified or, if you specified a blank or null session ID (for the current presentation space), you were not connected to a presentation space. HARC_BAD_PARM - Incorrect parameters were specified. HARC_SYSTEM_ERROR – System error occurred. This code will be returned if more than six HLLAPI applications try to start host notification on the same session.

Affected by SET SESSION PARAMETERS?

The following options affect this function:

EXTID/NOEXTID

If the EXTID option is set, any number between 0x01 and 0xFF can be used to specify the session ID and a value of 0 (zero) indicates the current session; NOEXTID causes session IDs to be limited to any single letter between A and Z and the current session can be identified with a blank or null.

XNOTIFY/NOXNOTIFY

If the XNOTIFY option is set, this verb cannot be called (see NOTIFY EVENT for information on using the message queue interface instead); NOXNOTIFY (the default) allows this verb to be used.

HLLAPI Functions: START HOST NOTIFICATION

Usage and Restrictions

- ◆ Multiple host sessions can be monitored for changes simultaneously by calling FunctionsSTART HOST NOTIFICATION once for each session. With the FunctionsSet Session ParametersIPAUSE option on, PAUSE can then be used to wait until any of these host sessions changes. FunctionsQUERY HOST UPDATE can be called for each session in turn to determine which one(s) had a change.
- ◆ Up to six HLLAPI applications may start host notification on the same session.

START KEYSTROKE INTERCEPT

This function intercepts entered by a terminal operator. After starting keystroke intercept, you can receive a keystroke using FunctionsGET KEY. You can then accept or reject the keystroke using FunctionsPOST INTERCEPT STATUS.

You can then enter the keystroke or replace it with another keystroke using FunctionsSEND KEY or use the keystroke to trigger another process.

Intercepted keystrokes are not displayed in the presentation space.

Prior Functions

None.

Request Parameters

func_number	HA_START_KEY_INTERCEPT (50)	
data_string	A six-character string indicating the session and which keystrokes should be intercepted.	
	Position	Description
	1	Session ID
	2	Option code character: D AID keystrokes only L all keystrokes
	3-6	Ignored.
data_length	Length of the intercept buffer used to hold the keystrokes. The length is variable, but HLLAPI will use a length of 256 bytes if you specify less.	
ps_position	Not used.	

HLLAPI Functions: START KEYSTROKE INTERCEPT

Returned Parameters

func_number 50

data_string Not used.

data_length Not used.

return_code HARC_SUCCESS – Function was successful.

HARC_INVALID_PS - Invalid presentation space was specified.

HARC_BAD_PARM – Invalid option was specified.

HARC_BUSY – Presentation space was busy; in use by another application.

HARC_SYSTEM_ERROR – System error occurred.

Affected by SET SESSION PARAMETERS?

The following option affects this function:

EXTID/NOEXTID

If the EXTID option is set, any number between 0x01 and 0xFF can be used to specify the session ID and a value of 0 (zero) indicates the current session; NOEXTID causes session IDs to be limited to any single letter between A and Z and the current session can be identified with a blank or null.

Usage and Restrictions

- ◆ If you use option code D (AID keystrokes only), only AID keystrokes are returned to the application. HLLAPI writes non-AID keys that it intercepts to the host presentation space for which they were originally intended.
- ◆ The FunctionsRESERVE function has no effect on this set of functions.
- ◆ Your application may start keystroke intercept on a session whether or not it is connected to that session; however, if another process has the session either connected or Functionskeystroke intercept started, the return code is HARC_BUSY.
- ◆ You can have keystroke intercept started for more than one session at a time.

STOP HOST NOTIFICATION

This function deactivates and disables the FunctionsQUERY HOST UPDATE function. It also prohibits host events in the specified host session from affecting the FunctionsPAUSE function.

Prior Functions

START HOST NOTIFICATION - You must request notification of updates before calling this function.

Request Parameters

func_number	HA_STOP_HOST_NOTIFY (25)
data_string	Session ID.
data_length	Not used.
ps_position	Not used.

Returned Parameters

func_number	25
data_string	Not used.
data_length	Not used.
return_code	HARC_SUCCESS – Stop notification was successful. HARC_INVALID – Function cannot be called if the XNOTIFY parameter is set. HARC_INVALID_PS – An invalid session ID was specified. HARC_NO_PRIOR_START – START HOST NOTIFICATION has not been requested for this presentation space.

HARC_SYSTEM_ERROR – System error occurred.

Affected by SET SESSION PARAMETERS?

The following options affect this function:

EXTID/NOEXTID

If the EXTID option is set, any number between 0x01 and 0xFF can be used to specify the session ID and a value of 0 (zero) indicates the current session; NOEXTID causes session IDs to be limited to any single letter between A and Z and the current session can be identified with a blank or null.

XNOTIFY/NOXNOTIFY

If the XNOTIFY option is set, this verb cannot be called (see NOTIFY EVENT for information on using the message queue interface instead); NOXNOTIFY (the default) allows this verb to be used.

Usage and Restrictions

- ◆ Your application does not have to be connected to a presentation space to check for updates. If you are not connected, you must specify the session ID in the request parameters; if you want to check for updates on the currently connected session, you do not need to specify the session ID.

STOP KEYSTROKE INTERCEPT

This function disables your application's ability to intercept. It should be used with the `FUNCTIONSSTART KEYSTROKE INTERCEPT` function.

Prior Functions

`START KEYSTROKE INTERCEPT` – A keystroke intercept must be started before calling this function.

Request Parameters

func_number	HA_STOP_KEY_INTERCEPT (53)
data_string	Session ID.
data_length	Not used.
ps_position	Not used.

Returned Parameters

func_number	53
data_string	Not used.
data_length	Not used.
return_code	HARC_SUCCESS – Function was successful; no more keystrokes will be intercepted. HARC_INVALID_PS – Invalid session ID was specified. HARC_NO_PRIOR_START – The <code>START KEYSTROKE INTERCEPT</code> function was not called prior to this function. HARC_SYSTEM_ERROR – System error occurred.

Affected by SET SESSION PARAMETERS?

The following option affects this function:

EXTID/NOEXTID

If the EXTID option is set, any number between 0x01 and 0xFF can be used to specify the session ID and a value of 0 (zero) indicates the current session; NOEXTID causes session IDs to be limited to any single letter between A and Z and the current session can be identified with a blank or null.

STORAGE MANAGER

This function allocates blocks of memory storage. It's provided for compatibility with other implementations of HLLAPI, particularly so that existing BASIC program maybe converted to C for use with CLEO'S HLLAPI implementation.

CLEO's HLLAPI implements this function using the malloc () and free () library functions. When writing new C applications, you should use these functions directly instead of using STORAGE MANAGER.

Prior Functions

None.

Request Parameters

func_number HA_STORAGE_MGR (17)

data_string Not used.

data_length (depends on the sub-function)

subfunction sub-function number

STORAGE MANAGER has four sub-functions that allow you to manipulate the storage buffer:

- ◆ GET STORAGE (01)
- ◆ FREE STORAGE (02)
- ◆ QUERY FREE STORAGE (03)
- ◆ FREE ALL STORAGE (04)

Each sub-function is specified through the fourth parameter (subfunction). Each sub-function also generates its own specific return parameters. Details on each sub-function follow.

Get Storage

GET STORAGE allocates a portion of storage to be used by a HLLAPI function. The third and fourth request parameters should contain:

data_length Size of requested storage area in bytes.

subfunction 1

Free Storage

FREE STORAGE frees the block of storage allocated by GET STORAGE. The third and fourth request parameters should contain:

data_length Storage Block ID (returned from GET STORAGE).

subfunction 2

Query Free Storage

QUERY FREE STORAGE returns the size of the largest available block of free storage. The third and fourth request parameters should contain:

data_length Not used.

subfunction 3

Free All Storage

FREE ALL STORAGE frees all previously allocated blocks of storage. The third and fourth request parameters should contain:

data_length Not used.

subfunction 4

Returned Parameters

Get Storage

func_number 17

data_string	If return_code is HARC_SUCCESS, this parameter contains the address of the storage block allocated. Otherwise it is not used.
data_length	If return_code is HARC_SUCCESS, this parameter gives the storage block ID. Otherwise it is not used.
return_code	HARC_SUCCESS – Requested storage has been allocated. HARC_STOREMAN_NO_MEM – Requested storage is more than amount available or the maximum number of blocks has already been allocated. HARC_SYSTEM_ERROR – System error occurred.

Free Storage

func_number 17

data_string	Not used.
data_length	Not used.
return_code	HARC_SUCCESS – Requested block has been freed. HARC_BAD_PARM – Invalid storage block ID. HARC_SYSTEM_ERROR – System error occurred.

When this subfunction has completed, the storage block ID is no longer valid.

Query Free Storage

func_number 17

HLLAPI Functions: STORAGE MANAGER

data_string	Not used.
data_length	Size of the largest available block of storage. Although this value may be required by a BASIC program, there is no need to test for it in a C program and no way of doing so; a number is returned which should be large enough for the requirements of a program originally written in BASIC.
return_code	HARC_SUCCESS – Free block was found. HARC_SYSTEM_ERROR - System error occurred.

Free All Storage

func_number 17

data_string	Not used.
data_length	The size of the largest available block of storage. Although this value may be required by a BASIC program, there is no need to test for it in a C program and no way of doing so; a number is returned which should be large enough for the requirements of a program originally written in BASIC.
return_code	HARC_SUCCESS - All blocks were freed. HARC_SYSTEM_ERROR – System error occurred.

When this subfunction has completed, none of the previously allocated storage block IDs are valid.

Usage and Restrictions

GET STORAGE:	Once a block of storage is allocated, it keeps its original size.
GET STORAGE:	The data_length is rounded up to a multiple of 16 before the block of storage is allocated.

WAIT

This function tests the status of the current, and optionally waits for the status to change. Use this function to give the host time to respond to requests from your application. Timeout options are provided to ensure that an application does not wait indefinitely for a status change (see SET SESSION PARAMETERS for more information).

Prior Functions

CONNECT PRESENTATION SPACE - Your application must be connected to an active presentation space before calling this function.

Request Parameters

func_number HA_WAIT (4)

data_string Not used.

data_length Not used.

ps_position Not used.

Returned Parameters

func_number 4

data_string Not used.

data_length Not used.

return_code HARC_SUCCESS – Keyboard is unlocked and ready for input.

HARC_NO_CONNECTION – Application is not connected to a valid session.

HARC_BUSY – Keyboard is busy; still in WAIT state.

HLLAPI Functions: WAIT

HARC_LOCKED – Keyboard is locked, input inhibited.

HARC_SYSTEM_ERROR – System error occurred.

Affected by SET SESSION PARAMETERS?

The FunctionsSet Session ParametersTWAIT, FunctionsSet Session ParametersLWAIT and FunctionsSet Session ParametersIWAIT options affect this function. TWAIT causes this function to wait up to one minute for the keyboard to become ready for input, then return to the application program. LWAIT causes the function to wait indefinitely for the keyboard to become ready before returning to the application program. NWAIT causes the function to check the keyboard status and immediately return to the application program without waiting.

Usage and Restrictions

- ◆ This function is particularly useful with FunctionsSEND KEY. Since keystrokes are not accepted in a session whose keyboard is busy or locked, use the FunctionsWAIT function (with the TWAIT or NWAIT option) to check the status of the keyboard before sending keystrokes.

A. OIA Image Information

This appendix explains the format of the Operator Information Area (OIA) data that's returned by the COPY OIA function (positions 2 to 81 of the returned data string). These positions can be accessed from C programs using `datastring[]` to `datastring[80]`, since C strings start at position 0 rather than 1. The following information numbers the positions from 1 to 80.

For each position(s) in the string, the following information is provided:

- ◆ Hexadecimal value returned by COPY OIA (this value corresponds to a symbol in the 3274 character generator ROM)
- ◆ Symbol corresponding to this value, as it appears on an IBM PC-3270 terminal's status line (see below)
- ◆ Corresponding indicator on the status line, if any
- ◆ Explanation of the value

The IBM PC-3270 symbols shown are an approximation to the 3274 symbols, as described in the 3274 Control Unit Description. Most are similar to the 3274 versions, except for which represents a “stick man” symbol (generally used to mean “operator”), and [] which represents a clock (used to mean “wait”).

Hexadecimal values shown in the table as hh represent numeric characters; the 3274 uses the values 0x20 - 0x29 to represent the characters 0-9, so that (for example) 123 is represented by the hex value 212223.

Positions not shown in the table are not used; they are set to the hex value F6 for positions 1-64, and null (00) for positions 65-80.

Position		IBM 3270 Hex	CLEO 3270 Symbols	Symbols Explanation
1	B2	S	S	3174 control unit is ready
2	CD	B	B	Host connection uses B rules (SNA protocol)
3	F0	†	*	3270 session is connected to the host SSCP program
	CF	■	#	3270 session is connected to a host application
	F1	?	?	3270 session is not connected to a host application or to SSCP
9	C6	X	X	Keyboard locked. This will be followed by a symbol in positions 11-17 indicating the reason why.
11-17	31 85	-f	-f	You used an invalid key
	FA C1 FA C4 C5	O-O[]	O-O()	Printer busy
	FA C1 FB	O-p	O-p	Local copy printer not working
	18 35	?+	?+	Input keystroke could not be accepted

Position	IBM 3270 Hex	CLEO 3270 Symbols	Symbols	Explanation
11-17	AF B1 AE A6 hh hh hh	PROGnnn	PROGnnn	Protocol error in data received from host
	E4 08	†>	*>	You tried to enter data into a full field
	C8 E4 E2	<-†->	<*>	You tried to enter data into a protected field
	E4 AD B4 AC	aNUM	*NUM	You tried to enter invalid data into a numeric field
	C1 C0 C2 C3 hh hh hh	-+z_nnn	-+z_nnn	Communications check code nnn
	B2 B8 B2 B3 A4 AC	SYSTEM	SYSTEM	Host system has locked the keyboard
	C4 C5	[]	()	Waiting for host system to complete a function
21-27	C1 C0 C2 C3 hh hh hh	-+z_nnn	(none)	Last recorded communications check code
	AF B1 AE A6 hh hh hh	PROG nnn	(none)	Last recorded program check code
31-32	B2 20	S0	(none)	3270 session is using base character set
36-38	AD B4 AC	NUM	(none)	Numeric override (NUM OVR key pressed)

Position		IBM 3270 Hex	CLEO 3270 Symbols	Symbols Explanation
52	D0	^	^	Keyboard is in insert mode
60-64	FA C1 FA 18 18	O-O??	(none)	Unknown printer identifier
	FA C1 C7	O-■	O-O()	Printing in progress

B. Sample Program and Examples

This appendix contains a listing of the sample HLLAPI C program (*h_sample.c*) and a small collection of HLLAPI examples (*h_extra.c*) both of which can be found in *usr/lib/linkix/samples/*.

Sample Program and Examples

```
/*          Cleo Communications, Inc.
*****
*          h_sample.c
*          Sample Application program for HLLAPI
*          Connects to PS, then logs on to VM host
*
*          ( Note: This example may differ from the listing in your )
*          ( documentation. The version supplied on your          )
*          ( installation diskette is the most up-to-date.          )
*
*          Substitute your own logon ID and password in the space provided.
*          The original application needed to specify an operating system name
*          {VM, TSO, MVS} on the first screen. The logon id and password were
*          entered on the second screen. An application name could be given
*          after that; this program issues a LOGOFF.
*
*          A single argument may be entered on the command line to specify
*          which session to connect to. {A-Z or 2-33} If no session specified,
*          default is currently connected presentation space.
*****
*/

#include <stdio.h>
#include <ctype.h>
#include "hapi_c.h"

/* There are many ways of entering the logon. With the host we use,
logon and password are entered in two consecutive fields. Adjust this
string to meet your own host's requirements. Two separate send calls
may be necessary. The logon ID is xxxxxxxx, password is yyyyyyyy. */

#define LOGON_STRING "xxxxxxxx@Tyyyyyyy@E"

#define TRACE_FILE "trace.out" /* results of commands are written to
                                stdout and to this file */

FILE * fileptr;
enum {FALSE, TRUE};

#define API_PS_SIZE 1920 /* default, for a Model 2 screen */

int api_func;
int api_len;
int api_ret;
char arg_sess; /* session parameter */

char api_string[API_PS_SIZE]; /* screen buffer */

/***** static putnib(chx) */
static putnib(chx)
int chx; /* convert a hex digit to the equivalent */
{ /* character and display it */
    chx = chx & 0x0F;
    if (chx > 9) chx = chx + (1 + 'A' - '9');
    fputc(chx + '0', fileptr);
}

/***** static putbyt(chx) */
static putbyt(chx)
int chx; /* display a byte as 2 hex characters */
{ /* using putnib */
    putnib(chx >> 4);
    putnib(chx);
}

/***** audit_hll(func,retc) */
audit_hll(func,retc)
int func; /* show the return code for the function */
```

```

int retc;
{
    char * fptr;
    char * rcptr;
    char fstr[40];
    char rcstr[40];

    switch (func)
    {
        case HA_CONNECT_PS:          fptr = "Connect";          break;
        case HA_DISCONNECT_PS:       fptr = "Disconnect";       break;
        case HA_SENDKEY:             fptr = "Send Key";         break;
        case HA_WAIT:                fptr = "Wait";             break;
        case HA_COPY_PS:             fptr = "Copy PS";           break;
        case HA_COPY_PS_TO_STR:      fptr = "Copy PS to String"; break;
        case HA_SEARCH_PS:           fptr = "Search PS";         break;
        case HA_QUERY_CURSOR_LOC:    fptr = "Query Cursor Location"; break;
        case HA_SET_SESSION_PARAMS:  fptr = "Set Session Parameters"; break;
        case HA_QUERY_SESSIONS:      fptr = "Query Sessions";    break;
        case HA_COPY_OIA:            fptr = "Copy OIA";          break;
        case HA_QUERY_FIELD_ATTR:    fptr = "Query Field Attribute"; break;
        case HA_COPY_STR_TO_PS:      fptr = "Copy String to PS"; break;
        case HA_PAUSE:               fptr = "Pause";             break;
        case HA_QUERY_SYSTEM:        fptr = "Query System";       break;
        case HA_RESET_SYSTEM:        fptr = "Reset System";       break;
        case HA_START_HOST_NOTIFY:   fptr = "Start Host Notification"; break;
        case HA_QUERY_HOST_UPDATE:   fptr = "Query Host Update";  break;
        case HA_STOP_HOST_NOTIFY:    fptr = "Stop Host Notification"; break;
        case HA_SEARCH_FIELD:        fptr = "Search Field";       break;
        case HA_FIND_FIELD_POS:      fptr = "Find Field Position"; break;
        case HA_FIND_FIELD_LEN:      fptr = "Find Field Length";  break;
        case HA_COPY_STR_TO_FIELD:   fptr = "Copy String to Field"; break;
        case HA_COPY_FIELD_TO_STR:   fptr = "Copy Field to String"; break;
        case HA_SEND_FILE:           fptr = "Send File";         break;
        case HA_RECEIVE_FILE:        fptr = "Receive File";       break;
        case HA_CONVERT_POS_ROW_COL: fptr = "Convert Position or RowCol"; break;

        default:
            sprintf(fstr, "not recognized - (%d)", func);
            fptr = fstr;
            break;
    }

    switch (retc)
    {
        case HARC_SUCCESS:           rcptr = "0 : succeeded";      break;
        case HARC_INVALID_PS:        rcptr = "1 : invalid PS connection"; break;
        case HARC_BAD_PARM:          rcptr = "2 : parameter error"; break;
        case HARCFT_COMPLETE:        rcptr = "3 : FT complete";    break;
        case HARC_BUSY:              rcptr =
            ((func == 90) || (func == 91)) ? "4 : FT complete - recs seg'd" :
            (func == 4) ? "4 : Timeout in XCLOCK or XSYS" :
            "4 : Host PS Busy"; break;
        case HARC_LOCKED:            rcptr = "5 : Host PS Locked";  break;
        case HARC_TRUNCATION:        rcptr = "6 : Data possibly truncated"; break;
        case HARC_INVALID_PS_POS:    rcptr =
            ((func == 90) || (func == 91)) ? "7 : FT is already active (inuse)" :
            "7 : PS position invalid"; break;
        case HARC_NO_PRIOR_START:    rcptr = "8 : No prior Start Host Notification"; break;
        case HARC_SYSTEM_ERROR:      rcptr = "9 : System Error";    break;
        case HARC_UNSUPPORTED:       rcptr = "10 : Storage Manager N/A"; break;
        case HARC_UNAVAILABLE:       rcptr = "11 : Resource Unavailable"; break;
        case HARC_OIA_UPDATE:        rcptr = "21 : OIA updated";    break;
        case HARC_PS_UPDATE:         rcptr = "22 : PS updated";     break;
        case HARC_PS_OR_OIA:         rcptr = "23 : PS or OIA (or both) updated"; break;
        case HARC_STR_NOT_FOUND_UNFM_PS: rcptr = "24 : No such Field"; break;
        case HARC_HOST_UPDATE:       rcptr = "26 : PS or OIA updated"; break;
        case HARCFT_OS_ERROR_BASE:   rcptr = "300 : FT - Base for OS/2 errors"; break;
        case HARCFT_FUNC:            rcptr = "301 : FT - invalid function number"; break;
        case HARCFT_NOT_FND:         rcptr = "302 : FT - file not found"; break;
        case HARCFT_ACCESS:          rcptr = "305 : FT - access denied"; break;
    }
}

```

Sample Program and Examples

```
case HARCFT_MEMORY:      rcptr = "308 : FT - insufficient memory";      break;
case HARCFT_ENV:         rcptr = "310 : FT - invalid environment";      break;
case HARCFT_FORMAT:      rcptr = "311 : FT - invalid format";          break;
case HARCFT_ALREADY_ACTIVE: rcptr = "110 : FT - already active";        break;

default:
    sprintf(rcstr,"not recognized - (%d)",retc);
    rcptr = rcstr;
    break;
}
fprintf(fileptr,"Function : %s ; Completed, Return code = %s\n",fptr,rcptr);
fprintf(stdout,"Function : %s ; Completed, Return code = %s\n",fptr,rcptr);
}

/***** Connect Presentation Space */
int connect_ps(sess_id)
char sess_id;
{
    api_func = HA_CONNECT_PS;          /* api_len & api_retc not req'd */
    api_string[0] = sess_id;
    clim(&api_func,
        api_string,
        &api_len,
        &api_retc);
    audit_hll(api_func,api_retc);
    return(api_retc);
}

/***** Send Key */
send_key(str)
char * str;                          /* keystroke(s) to be sent */
{
    api_func = HA_SENDKEY;
    api_len = strlen(str);
    clim(&api_func,
        str,
        &api_len,
        &api_retc);
    audit_hll(api_func,api_retc);
    fprintf(fileptr,"string: '%s'\n", str);
    return(api_retc == HARC_SUCCESS ? 0 : -1 );
}

/***** Wait */
hwait()
{
    api_func = HA_WAIT;                /* api_len & api_retc not req'd */
    clim(&api_func,
        api_string,
        &api_len,
        &api_retc);
    audit_hll(api_func,api_retc);
    return(api_retc);
}

/***** Set Session Parameters */
int set_sess_params(str)
char * str;                          /* string of parms to be set */
{
    api_func = HA_SET_SESSION_PARMS;
    api_len = strlen(str);
    api_retc = 0;
    clim(&api_func,
        str,
        &api_len,
        &api_retc);
    audit_hll(api_func,api_retc);
    return(api_retc == HARC_SUCCESS ? 0 : -1 );
}

/***** Search Presentation Space */
int search_presentation_space(str,pos)
```

Sample Program and Examples

```
char * str;                                /* string to search for */
int pos;                                   /* starting position for search */
{
    api_func = HA_SEARCH_PS;
    api_len = strlen(str);
    api_retc = pos;
    clim(&api_func,
         str,
         &api_len,
         &api_retc);
    audit_hll(api_func,api_retc);
    return(api_retc);
}

/***** Pause *****/
int pause(time)
int time;                                /* how long to pause for */
{
    api_func = HA_PAUSE;
    api_len = time;
    api_retc = 0;
    clim(&api_func,
         api_string,
         &api_len,
         &api_retc);
    audit_hll(api_func,api_retc);
    return(api_retc);
}

/***** query host update */
int query_host_update()
{
    api_func = HA_QUERY_HOST_UPDATE;
    api_string[0] = arg_sess;
    clim(&api_func,
         api_string,
         &api_len,
         &api_retc);
    audit_hll(api_func,api_retc);
    return(api_retc);
}

/***** start host notification */
int start_notify()
{
    api_func = HA_START_HOST_NOTIFY;
    api_string[0] = arg_sess < 'A' ? arg_sess : ' ';
    api_string[1] = 'P';                /* request notification of update to PS */
    clim(&api_func,
         api_string,
         &api_len,
         &api_retc);
    audit_hll(api_func,api_retc);
    return(api_retc);
}

/***** Wait for particular screen text */
int wait_4_screen(scan_str)
char *scan_str;                          /* the text to wait for

Begin a loop doing the following:
Scan the PS for the specified character string. Exit the loop if found.
If not found, start host notify and pause until the PS is updated.
[Remember that set_session_parameters was called with IPAUSE so the
pause is interrupted when PS is updated.]
On return from the pause, Query_host_update tells us whether the PS was,
in fact, updated. If not, we timed out; exit with error.
Repeat the loop a few times. (How many depends on your host.)
```

Sample Program and Examples

When the string is found, wait for the keyboard to be unlocked by calling wait.
[Note that the OIA could be checked for unlock. Calling wait is easier.]

```
*/
{
    int rc, i;

    for(i=8; --i;) /* adjust number of tries as required */
    {
        if(search_presentation_space(scan_str, 1)==HARC_SUCCESS)
            return(hwait()); /* return when keyboard is unlocked */

        if((rc=start_notify()) != HARC_SUCCESS)
            return(rc);

        /* wait for PS update */
        if((rc=pause(120)) == HARC_SYSTEM_ERROR)
            return(rc);

        rc = query_host_update();
        if(rc != HARC_PS_UPDATE && rc != HARC_PS_OR_OIA)
            return(rc); /* note that HARC_PS_OR_OIA really means
PS_AND_OIA */
    }
    return(-1);
}

/***** Reset system */
reset()
{
    api_func = HA_RESET_SYSTEM;
    clim(&api_func,
        api_string,
        &api_len,
        &api_ret);
    audit_hll(api_func,api_ret);
    return(api_ret);
}

/***** Disconnect Presentation Space */
disconnect_ps()
{
    api_func = HA_DISCONNECT_PS; /* api_len & api_ret not req'd */
    clim(&api_func,
        api_string,
        &api_len,
        &api_ret);
    audit_hll(api_func,api_ret);
    return(api_ret == HARC_SUCCESS ? 0 : -1 );
}

/***** SEND FILE */
send_file(str)
char * str; /* file transfer parameters to send */
{
    api_func = HA_SEND_FILE;
    api_len = strlen(str);
    clim(&api_func,
        str,
        &api_len,
        &api_ret);
    audit_hll(api_func,api_ret);
    return(api_ret == HARC_SUCCESS ? 0 : -1 );
}

/***** RECEIVE FILE */
receive_file(str)
char * str; /* file transfer parameters to send */
{
    api_func = HA_RECEIVE_FILE;
```

Sample Program and Examples

```
api_len = strlen(str);
clim(&api_func,
     str,
     &api_len,
     &api_retc);
audit_hll(api_func,api_retc);
return(api_retc == HARC_SUCCESS ? 0 : -1 );
}

/***** check arg */
int check_arg(arg)
char *arg;
{
    int sess_n;

    if(sess_n=atoi(arg))
    {
        /* specified a number - check it */
        if(sess_n < 2 || sess_n > 33)
        {
            fprintf(stderr, "Given session number \"%d\": out of range (2-33)\n",
                    sess_n);
            exit(1);
        }
        arg_sess = sess_n; /* indicate using actual device */
        return(set_sess_params("SPECIFICLU")==HARC_SUCCESS);
    }
    else
    {
        if(isalpha(arg[0]) && arg[1] == '\0')
            arg_sess = arg[0];
        else
        {
            fprintf(stderr,
                    "Requested session id parameter \"%s\" is not a single letter\n",
arg);
            exit(1);
        }
    }
    return(TRUE);
}

/***** supplies user ID and password */
logon_to_vm()
{
    /* This function logs on VM. It handles waiting for the proper
screen
and entering the IDs. User must define LOGON_STRING. */
    int rc;

    if((rc=wait_4_screen("ENTER SYSTEM ID :")) != HARC_SUCCESS)
        return(rc);

    if((rc=send_key("VM0E")) != HARC_SUCCESS) /* select op-sys */
        return(rc);

    if((rc=wait_4_screen("USERID   ==>")) != HARC_SUCCESS)
        return(rc);

    if((rc=send_key(LOGON_STRING)) != HARC_SUCCESS)
        return(rc);

    return(hwait()); /* return when keyboard is unlocked */
}

/***** main program */
main(argc, argv)
int argc; char *argv[];
{
    int rc;

    fileptr = fopen(TRACE_FILE, "w");
    setbuf(stdout, NULL); /* don't buffer output -- show it NOW */
}
```

Sample Program and Examples

```
if(fileptr == 0)
{
    puts("Could not open trace.out file - \n check permissions on directory or
existing trace.out file.\n");
    exit(0);
}

setbuf(fileptr, NULL);
fputs("HLLAPI Sample Application\n",fileptr);
puts("HLLAPI Sample Application\n");

if(argc < 2)
    arg_sess = 'A';          /* no argument -- assign default session */
else
{
    if(!check_arg(argv[1]))   /* set arg_sess based on user-specified
argument */
        exit(1);
}

if((rc=reset()) != HARC_SUCCESS)/* reset is a good idea for new sessions */
    exit(rc);

fputs("connecting PS...\n", fileptr);
rc = connect_ps(arg_sess);
if(rc == HARC_SUCCESS || (rc == HARC_LOCKED))
{
    set_sess_params("IPAUSE"); /* PS or OIA changes will interrupt PAUSE */
    if(logon_to_vm() != HARC_SUCCESS)
        exit(1);
    /******
    /*      should now be in CMS with keyboard unlocked      */
    /******
    /******
    /*      send string to run application program            */
    /******
    /* May need to scan for "R:" if CMS PROFILE sets short ready */
    /* message. Better way would be to have PROFILE write some */
    /* unique string to scan for, then wait for keyboard unlock. */

    if((rc=wait_4_screen("Ready;")) == HARC_SUCCESS)
    {
        rc=send_key("identify #query time@E"); /* run a CMS application */
    }

    /******
    /* time to let the user take over... disconnect program from PS. */
    /******

    fprintf(fileptr,"disconnecting PS...\n");
    rc = disconnect_ps();
}
else
{
    exit(1);          /* connect was un-successful */
}
exit(0);
}
```


C. Header File: hapi_c.h

This appendix contains a listing of the C header file *hapi_c.h*. Please take note that the exact formatting of the header file you have may differ slightly from the way it appears here.

```
/* hapi_c.h */
/* (C) COPYRIGHT CLEO COMMUNICATIONS 1993 */

#ifndef HA_RESET_SYSTEM

/*****
/*
/* HLLAPI Definitions for User Application Programs
/*
/*
*****/

/*****
/*
/* HLLAPI verbs
/*
/*
*****/

/*****
/* Initialise/Reset Functions
/*
*****/
#define HA_RESET_SYSTEM 21

#define HA_SET_SESSION_PARMS 9

#define HA_QUERY_SESSIONS 10

#define HA_QUERY_SESSION_STATUS 22

#define HA_CONNECT_PS 1

#define HA_DISCONNECT_PS 2

/*****
/* Interaction Functions
/*
*****/
#define HA_RESERVE 11

#define HA_RELEASE 12

/*****
/* Extraction Functions
/*
*****/
#define HA_COPY_OIA 13

#define HA_COPY_PS 5

#define HA_COPY_PS_TO_STR 8

#define HA_COPY_FIELD_TO_STR 34
```

Header File: hapi_c.h

```
/* **** */
/* Orientation Functions */
/* **** */
#define HA_SEARCH_PS 6

#define HA_QUERY_CURSOR_LOC 7

#define HA_QUERY_FIELD_ATTR 14

#define HA_SEARCH_FIELD 30

#define HA_FIND_FIELD_POS 31

#define HA_FIND_FIELD_LEN 32

#define HA_CONVERT_POS_ROW_COL 99

/* **** */
/* Monitoring Functions */
/* **** */
#define HA_START_HOST_NOTIFY 23

#define HA_STOP_HOST_NOTIFY 25

#define HA_QUERY_HOST_UPDATE 24

#define HA_WAIT 4

#define HA_PAUSE 18

/* **** */
/* Sending Functions */
/* **** */
#define HA_COPY_STR_TO_FIELD 33

#define HA_COPY_STR_TO_PS 15

#define HA_SENDKEY 3

#define HA_SET_CURSOR 40

/* **** */
/* Storage Management Function */
/* **** */
#define HA_STORAGE_MGR 17

/* **** */
/* File Send/Receive Functions */
/* **** */
#define HA_SEND_FILE 90

#define HA_RECEIVE_FILE 91

/* **** */
/* Keystroke Intercept Functions */
/* **** */
#define HA_START_KEY_INTERCEPT 50

#define HA_STOP_KEY_INTERCEPT 53

#define HA_GET_KEY 51

#define HA_POST_INTERCEPT_STATUS 52

/* **** */
/* Debugging Functions */
/* **** */
#define HA_QUERY_SYSTEM 20
```

HLLAPI Programmer's Guide C-3

Header File: hapi_c.h

```

/*****
/* New return codes for MS/COM
*****/
#define HARC_ALREADY_INT      401 /* already intercepting keystrokes */
#define HARCFT_IN_PROG      402 /* FT - already in progress */

/*****
/* HLLAPI status codes on the Convert Position or RowCol verb
*****/
#define HARC99_INVALID_PS      9998 /* alternative for code 1 on CPR */
#define HARC99_INVALID_CONV_OPT 9999 /* character 2 in string not P or R */
#define HARC99_INVALID_INP      0 /* incorrect PS position */

/*****
/* HLLAPI color codes for the DISPLAY_ATTRIBUTES verb
*****/
#define HLL_BLACK      0x00
#define HLL_BLUE      0x01
#define HLL_GREEN      0x02
#define HLL_TURQ      0x03
#define HLL_RED      0x04
#define HLL_PINK      0x05

#define HLL_YELLOW      0x06
#define HLL_WHITE      0x07
#define HLL_INT_BLUE      0x08
#define HLL_INT_GREEN      0x09
#define HLL_INT_TURQ      0x0A
#define HLL_INT_RED      0x0B
#define HLL_INT_PINK      0x0C

#define HLL_INT_YELLOW      0x0D
#define HLL_INT_WHITE      0x0E

#define HLL_MAX_COLORS      0x0F

/*****
/* Function Prototypes
*****/
extern void hllapi();

extern void hllc();

extern void clim();

/*****
/* Defines and typedefs for Super-HLLAPI call
*****/

/* Maximum number of HLL functions in one super-HLLAPI call. */
#define MAX_HLL_FUNCTIONS 20

/* Minimum number of bytes needed to store information on a single */
/* function that makes up part of a super-HLLAPI command. */
#define MIN_SUB_FUNCTION_SIZE 5*sizeof(int)

/* Error codes */
#define OUT_OF_MEMORY (-1) /* Not enough memory to store the function */
/* in shared memory. */

/* Individual function entries in table passed to Super-HLLAPI call */
typedef struct HLL_PARAMETERS
{
    int func_number; /* These have the meaning defined in the */
    char *string; /* documentation for individual functions. */
    int data_length;
    int ps_pos;
} HLL_PARAMETERS;

```

```
/* Format used to store individual function entries into the 'api_string' */
/*   area of shared memory. */
typedef struct SUB_FUNCTION
{
    int api_func;          /* Function number of this individual function. */
    int api_ret;          /* Ps_pos on entry, return code from 3270. */
    int api_len;          /* Api-len passed by caller. */
    int copy_to;          /* Number of bytes passed to 3270. */
    int copy_back;        /* Number of bytes returned by 3270. */
    char api_string[1];   /* Variable length storage for passed parameters */
                        /* and returned results. */
} SUB_FUNCTION;

#endif

/* hapi_c.h */
```

Header File: hapi_c.h

Index

3270

3270

screen models, 1-6

3270 keys

¢ (cent), 5-102

¬ (logical NOT), 5-102

I (solid vertical bar), 5-102

DUP, 5-33, 5-36

FMARK, 5-33, 5-36

mnemonics, 5-103

3270 session

information, 5-74

status, 5-72

A

AID keys, 3-5, 3-6, 3-8

Application design

compatibility

with DCA/Microsoft products, 3-16

with OS/2 extended edition, 3-16

declaring HLLAPI parameters, 3-10

general guidelines, 3-2

initializing function parameters, 3-12

passing the function code, 3-11

passing/returning a presentation space

position, 3-11

passing/returning a string of text, 3-11

portability, 3-16

requesting a function, 3-12

synchronizing with the host, 3-8

waiting for host responses, 3-6

ASCII copy

from field, 5-18

from presentation space, 5-25, 5-28

to presentation space, 5-34

ASCII/EBCDIC translation, 5-31, 5-34

Attention identifier (AID) key, 2-12,
2-19

Automating manual operations, 3-4

C

CGA terminal format, 5-30, 5-113

Character attributes

character set, 2-14

color, 2-14, 2-17

highlighting, 2-14

Character translation, 5-31, 5-34

Compiling, 4-2

Copy OIA, A-1

Cursor

movement, 2-19

D

Data types, 3-10

Debugging, 4-4

DOS HLLAPI implementations, 1-3

E

Error-handling, status codes, 5-17

Escape character, 5-58, 5-114

Extended attribute buffer, 2-13

F

Field attribute byte, 2-10

Field attributes

character set, 2-13

color, 2-13, 2-17

highlighting, 2-13

Fields, 5-54
 attribute byte, 2-9
 attributes, 5-67, 5-111
 auto-skip, 2-11
 characteristics, 2-16
 copying to, 5-31
 first position, 5-53
 length, 5-50
 locating, 2-11
 Modified Data Tag (MDT), 2-11
 searching, 5-90, 5-116

File transfer
 receive, 5-80
 return codes, 5-81, 5-97
 send, 5-96
 timeout, 5-117

Formatted presentation space, 2-8

Function number, 5-2

Functions
 CONNECT PRESENTATION SPACE, 3-2, 3-8, 5-12, 5-83, 5-99, 5-117
 CONVERT POSITION OR ROWCOL, 2-7, 5-15
 COPY FIELD TO STRING, 3-11, 5-14, 5-18, 5-30, 5-113
 COPY OIA, 3-7, 3-11, 5-14, 5-21
 COPY PRESENTATION SPACE, 3-4, 3-11, 5-14, 5-25, 5-111, 5-113
 COPY PRESENTATION SPACE TO STRING, 3-4, 3-11, 5-14, 5-26, 5-28, 5-111, 5-113
 COPY STRING TO FIELD, 2-11, 2-19, 3-4, 3-5, 3-6, 5-14, 5-31, 5-71, 5-102, 5-113
 COPY STRING TO PRESENTATION SPACE, 2-11, 2-19, 3-4, 3-5, 3-6, 3-12, 3-13, 5-14, 5-31, 5-71, 5-102, 5-113
 DISCONNECT PRESENTATION SPACE, 3-3, 5-13, 5-14, 5-37, 5-85, 5-87, 5-99
 DISPLAY ATTRIBUTE, 5-14, 5-39
 DISPLAY OFF, 5-14, 5-42
 DISPLAY ON, 5-14, 5-45
 DISPLAY TEXT, 5-14, 5-47, 5-113
 FIND FIELD LENGTH, 5-14, 5-20, 5-50
 FIND FIELD POSITION, 2-11, 5-14, 5-20, 5-53
 GET KEY, 3-2, 5-55, 5-64, 5-114, 5-117, 5-118, 5-119, 5-123
 NOTIFY EVENT, 5-59, 5-114, 5-117
 PAUSE, 3-4, 3-7, 5-62, 5-102, 5-115, 5-120, 5-125
 POST INTERCEPT STATUS, 5-55, 5-64, 5-117, 5-123
 QUERY CURSOR LOCATION, 3-3, 5-14, 5-36, 5-66
 QUERY FIELD ATTRIBUTE, 2-11, 2-12, 5-14, 5-67
 QUERY HOST UPDATE, 3-6, 3-8, 5-63, 5-69, 5-114, 5-117, 5-120, 5-122, 5-125
 QUERY SESSION STATUS, 5-72, 5-117
 QUERY SESSIONS, 3-11, 3-14, 5-27, 5-74, 5-117
 QUERY SYSTEM, 3-4, 3-11, 3-16, 5-77
 RECEIVE FILE, 3-16, 5-80, 5-113, 5-115, 5-117, 5-118
 RELEASE, 3-3, 5-14, 5-84, 5-99
 RESERVE, 3-2, 5-14, 5-38, 5-83, 5-84, 5-86, 5-99, 5-124
 RESET SYSTEM, 3-2, 3-17, 5-13, 5-38, 5-88, 5-99
 SEARCH FIELD, 5-14, 5-90, 5-114, 5-116
 SEARCH PRESENTATION SPACE, 3-4, 5-14, 5-93, 5-114, 5-116
 SEND, 3-6
 SEND FILE, 3-16, 5-96, 5-114, 5-115, 5-117, 5-118
 SEND KEY, 2-11, 2-19, 3-2, 3-4, 3-5, 3-6, 5-14, 5-57, 5-71, 5-106,

5-114, 5-115, 5-116, 5-123,
 5-134
 SEND MULTIPLE, 5-14, 5-105
 SET CURSOR, 2-19, 5-14, 5-16, 5-71,
 5-108
 SET SESSION PARAMETERS, 2-9, 2-18,
 3-2, 3-7, 3-8, 4-7, 5-4, 5-110,
 5-117
 ATTRB, 5-26, 5-29, 5-111
 AUTORESET, 5-101, 5-115
 CONLOG, 5-112
 CONPHYS, 5-112
 EAB, 5-19, 5-20, 5-26, 5-27, 5-29,
 5-32, 5-35, 5-112
 EOT, 5-32, 5-34, 5-36, 5-48, 5-82,
 5-91, 5-94, 5-98, 5-101, 5-113
 ESC, 5-56, 5-58, 5-101, 5-103,
 5-114
 EXTID, 5-13, 5-56, 5-60, 5-65,
 5-70, 5-73, 5-75, 5-116, 5-121,
 5-124, 5-126, 5-128
 FPAUSE, 5-63, 5-115
 IPAUSE, 5-62, 5-63, 5-115, 5-120,
 5-122
 LWAIT, 5-56, 5-118, 5-134
 NOATTRB, 5-26, 5-29, 5-111
 NOEAB, 5-19, 5-26, 5-29, 5-32,
 5-35, 5-112
 NOEXTID, 5-13, 5-56, 5-60, 5-65,
 5-70, 5-73, 5-75, 5-116, 5-121,
 5-124, 5-126, 5-128
 NOQUIET, 5-82, 5-98, 5-115
 NORESET, 5-101, 5-115
 NOXLATE, 5-19, 5-26, 5-29, 5-112
 NOXNOTIFY, 5-60, 5-70, 5-114,
 5-121, 5-126
 NWAIT, 5-56, 5-118, 5-134
 QUIET, 5-82, 5-98, 5-115
 reinitializing, 5-88
 SRCHALL, 5-91, 5-94, 5-116
 SRCHBKWD, 5-91, 5-94, 5-116
 SRCHFROM, 5-91, 5-94, 5-116
 SRCHFRWD, 5-91, 5-94, 5-116
 STREOT, 5-31, 5-32, 5-33, 5-34,
 5-35, 5-47, 5-48, 5-83, 5-90,
 5-91, 5-94, 5-98, 5-101, 5-113
 STRLEN, 5-31, 5-32, 5-33, 5-35,
 5-47, 5-48, 5-83, 5-90, 5-91,
 5-94, 5-98, 5-101, 5-113
 TIMEOUT, 5-83, 5-99, 5-117
 TROFF, 5-118
 TRON, 5-118
 TWAIT, 5-56, 5-118, 5-134
 WAIT, 5-102
 XLATE, 5-19, 5-26, 5-29, 5-30,
 5-112
 XNOTIFY, 5-60, 5-70, 5-114,
 5-121, 5-126
 START HOST NOTIFICATION, 5-62,
 5-69, 5-114, 5-115, 5-117, 5-120,
 5-122
 START KEYSTROKE INTERCEPT, 3-2,
 5-56, 5-64, 5-117, 5-123, 5-127
 STOP HOST NOTIFICATION, 5-114,
 5-117, 5-125
 STOP KEYSTROKE NOTIFICATION, 5-117,
 5-127
 STORAGE MANAGER, 5-129
 WAIT, 3-2, 3-4, 3-6, 3-7, 3-8, 5-14,
 5-102, 5-118, 5-119, 5-133,
 5-134

H

hapi_c.h, 3-10
 HLLAPI
 level number, 5-77
 release date, 5-77
 version number, 5-77
 HLLAPI applications
 running, 1-5
 HLLAPI arguments, 3-10
 Host notification

- activating, 5-120
- deactivating, 5-125

I

IPAUSE, 3-7, 3-8

K

Keystrokes

- disabling intercepts, 5-127
- intercepting, 5-123
- retrieving, 5-55
- sending, 5-100

L

Linking, 4-2

LWAIT, 3-2

M

Memory storage, 5-129

N

NWAIT, 3-6, 3-8, 5-134

O

Operating System Error Codes, 5-82, 5-98

Operator Information Area (OIA), 2-2, 5-21, 5-69, 5-120, A-1

P

PC session, 5-75

Presentation space, 2-2

- connecting to, 5-12
- copying, 5-25, 5-28
- cursor movement, 5-108
- cursor position, 5-66

- determining size, 5-27
- disconnecting from, 5-37
- fields, 2-8
- formatted, 2-8, 2-9, 5-54
- locking, 5-86
- position, 5-15
- row and column coordinates, 2-3, 5-15
- screen addresses, 2-3
- searching, 5-93, 5-116
- status, 5-69, 5-72, 5-120, 5-133
- status line, 5-21
- unformatted, 2-8
- unlocking, 5-84

Process ID, 4-8

R

Return code, 5-3

Running a HLLAPI application, 4-3

S

Screen buffer, 2-2

Screen layout, 2-9

Status line, 2-2, 5-21, A-1

Symbolic name, 5-2

T

Terminals, 2-2, 2-7

Tracing, 4-4, 5-118

- environment variables
 - SNACTL, 4-7
 - SNAMSG, 4-6
 - SNATRC, 4-5
 - SNATRUNC, 4-6
- file(s), 4-5
- limiting data, 4-5, 4-6
- messages, ASCII/EBCDIC translation, 4-9

- output sample, 4-8
- suspending HLLAPI trace control, 4-7
 - within HLLAPI applications, 4-7
- Tracking, 2-19
- Translation of characters, 5-31, 5-34
- TRON/TROFF, 4-7
- TWAIT, 3-7, 5-134

U

- Unformatted presentation space, 2-8

W

- WAIT, 3-6
- Waiting for host responses, 3-6

X

- XLATE, 2-18

NetView User's Guide

December 1995



December 13, 1995

NetView User's Guide

December 1995

Interface Systems reserves the right to, without notice, modify or revise all or part of this document and/or change product features or specifications and shall not be responsible for any loss, cost or damage, including consequential damage, caused by reliance on these materials.

This document may not be reproduced, stored in a retrieval system or transmitted, in whole or in part, in any form or by any means (electronic, mechanical, photocopied or otherwise) without the prior written permission of Interface Systems.

GOVERNMENT RESTRICTED RIGHTS

Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Use, reproduction or disclosure is subject to 52.227-19 (a) through (d) and restrictions set forth in the accompanying end user agreement.

GOVERNMENT LIMITED RIGHTS

Limited rights shall be effective indefinitely and are not subject to expiration as set forth in paragraph (3) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Copyright © 1995 Interface Systems, Inc. – All rights reserved.

Portions:

Copyright © 1989–1992 Data Connection Limited

Copyright © 1989, 1990 Digital Communications Associates, Inc.

Copyright © 1989, 1990 Microsoft Corporation

Table of Contents

1. Introduction	
NetView Basics	1 - 2
Accessing the Host NetView Program	1 - 3
NetView Screen Display	1 - 5
NetView Commands	1 - 6
The UNIX Command Facility (UCF)	1 - 7
 2. NetView Commands	
Network Access Commands	2 - 2
Login SP	2 - 3
Display Applications	2 - 4
Node Commands	2 - 5
Display APPC LU	2 - 7
Display Partner LU	2 - 8
Display Mode	2 - 9
Display TP	2-10
Display Link	2-11
Display Connection	2-12
Display Session	2-13
Display Old LU (3270, LUA or RJE)	2-14
Display Old LU Pool (3270 or LUA)	2-16
Activate Connection	2-17
Deactivate Connection	2-18
Activate Session	2-19
Deactivate Session	2-20
Deactivate Old LU (3270, LUA, or RJE)	2-21

The UNIX Command Facility (UCI?)	2 - 22
Output from UNIX Commands	2 - 24
Valid UNIX Commands	2 - 24
Canceling a Command	2 - 25
UCF Security	2 - 25
 3. Network Management API	
Overview of the Network Management API	3 - 2
Summary of NM-API verbs	3 - 2
NM-API Verbs in C Programs	3 - 3
VCB Structure	3 - 3
Issuing a Verb	3 - 4
Designing and Writing NM-API Applications	3 - 5
Message Filtering	3 - 5
Replying to Received Messages	3 - 5
Multiple NM-API Applications	3 - 6
Use of Signals	3 - 6
NM_CLOSE	3 - 7
NM_OPEN	3 - 8
NM_RECEIVE	3 - 11
NM_SEND	3 - 15
NM-API Sample Program	3 - 18
Listing of <i>nmap_i_c.h</i>	3 - 31

1. Introduction

This chapter introduces basic information on the availability and use of NetView features and functions. The following topics are covered:

- ◆ NetView basics
- ◆ Accessing the NetView program
- ◆ Overview of NetView commands
- ◆ Overview of the UNIX Command Facility (UCF)

In order to use this feature, the host must be running version 1 release 2 or later of NetView.

NetView Basics

NetView is a host program that allows you to control and monitor both physical and logical network resources. *Physical* network resources are the hardware components that make the connections between the host, the communications controller, the cluster controller, cables or phone lines, and various devices such as printers and terminals. *Logical* network resources consist of the following network addressable units (NAUs): SSCP, PU and LU; and the SNA sessions that connect the NAUs.

To take advantage of this facility, you use the Diagnostics menu in the configuration program to designate one of your configured connections as the network management connection. This connection must be a host connection. See chapter 10 in the *Administration Guide*, **Configuring Diagnostics**, for instructions on designating a network management connection.

You can then log onto the host NetView program as a NetView operator using the 3270 emulation program to access the host (NetView can also be accessed directly from a NetView console at the host computer). After you've logged onto the host NetView program, you issue commands that can display the Cleo configuration and control the operation of Cleo software. With the additional UNIX command facility you can also issue UNIX commands that will be directed to the UNIX computer.

Accessing the Host NetView Program

To use the NetView program, you will require the following:

- ◆ A login ID and password for the host NetView program (contact your host administrator for this information)
- ◆ A service point name for Cleo 3270, defined at the host for the NetView program (contact your host administrator for this information)
- ◆ A local node, link and connection to the host computer on which the NetView program is running; this connection should be configured as the network management connection
- ◆ A 3270 LU configured for the host; this LU is configured to use the network management connection (it can use another connection to the same host)
- ◆ A 3270 session using the configured LU

Contact your host administrator to obtain the necessary configuration information.

To access the NetView program:

1. Ensure that the control daemon has been started using a configuration file which includes a network management connection to the host, and ensure that this connection is active.
2. Make sure the NETMAN service has been started. You start NETMAN with the *startsrv* utility (see chapter 11 of the *Administration Guide*):

 startsrv netman

 or with *commanage* (provided with the Cleo Management Utilities package)
3. If you are accessing the NetView program using 3270 emulation, start the 3270 emulation program and activate the session to the host. (See the *3270 User's Guide* if necessary.)
4. Follow the instructions provided by the host administrator for starting NetView and logging on. (The operations vary with different versions of NetView.)

Introduction

- 5 . Before issuing any other NetView command, you must first issue the Login SP command. This identifies you as a NetView operator.
- 6 . Issue the necessary NetView commands.
- 7 . When you have finished issuing NetView commands, follow the instructions in the *3270 User's Guide* for ending 3270 emulation.

```
NCCF          N E T V I E W          ISCAN DDAC12      07/18/90 13:52:24 A
* ISCAN      RUNCMD SP=ISCD22PU,APPL=NETMAN,LOGIN-SP NAME(DDAC12)
              NEW HOST OPERATOR IS DDAC12
-----

```

The command input area is typically one line, by default

NetView Commands

The ability to control and monitor network resources is carried out through NetView commands. These commands are divided into the following categories:

Network access commands	These commands allow you to establish communications between the Cleo software and NetView, and to list the Cleo services that are currently running. These commands are processed by the network management daemon.
Node commands	These commands allow you to view information on resources available at a local node, and to control these resources. These commands are processed by the local node that owns the resources.
UNIX Command Facility	The UNIX Command Facility (UCF) allows you to send UNIX commands from NetView to be executed on the UNIX computer. UNIX commands sent from NetView in this way are processed by the UCF daemon, which runs on the same server that the network management daemon is running on and the commands are issued on this server.

In addition, commands may also be sent to an NM-API application. In this case, the possible commands are determined by the application. See chapter 3, **Network Management API**, for more information on designing and writing NM-API applications.

The UNIX Command Facility (UCF)

The UNIX Command Facility (UCF) allows a NetView operator to execute UNIX commands on the computer that's running Cleo software by typing the command text at the NetView console. The NetView operator can also view output from these commands. The facility is not restricted to commands that are solely related to Cleo software; any type of command can be issued, subject to the restrictions described in chapter 2 under **Valid Commands**.

The UCF allows a remote operator to monitor activity on the UNIX computer, to diagnose problems and, in some cases, to take corrective action.

The UNIX commands are processed by the UCF daemon program, which runs on the same server that the network management daemon program is running on and the commands are issued on this server. (If you are using the Cleo software as a standalone product, and not in a client/server setting, the machine that the Cleo software is installed on is the server.) The commands from the UCF are executed by starting a new UNIX shell for each command and running the command in that shell.

The Cleo configuration specifies the name of the UCF user, which must be a valid login name on the server on which the network management daemon is running; the UCF shell is started using the shell program, login ID, permissions and the login or *.profile* specified for that user. This means that the normal UNIX security features can be used to restrict the UCF user's access to files and commands, and hence to limit the range of commands available from UCF.

2. NetView Commands

This chapter describes the NetView commands which you can issue from the host NetView program. These commands allow you to display the Cleo configuration and control the program's operation, or you can issue commands to the UNIX computer. The commands covered in this chapter consist of

- ◆ Network access commands
- ◆ Node commands
- ◆ UNIX Command Facility

Network Management API

In addition to the interactive NetView command capabilities, you may also send commands to a Network Management API (NM-API) application. In this case, the available commands are determined by the application. See chapter 3, **Network Management API**, for complete information on NM-API applications.

Network Access Commands

The two network access commands allow you to establish communication with the NetView program and to display the Cleo software components to which NetView commands can be sent.

The general syntax for network access commands is:

```
runcmd sp=spname, appl=netman, command [param(value)]
```

runcmd	This identifier indicates that the text that follows is a NetView command.
<i>sp=spname</i>	The name of your service point, as defined at NetView. Your host NetView personnel can give you this information.
<i>appl=netman</i>	The name of the Cleo component to which NetView should send the command. Network access commands are sent to the network management daemon (netman).
<i>command</i>	The actual NetView command; this is generally two keywords (an action and a component) separated by a hyphen.

One or more additional parameters maybe required, depending on the command.

<i>param(value)</i>	Parameters indicate data such as the names of software components or resources.

Login SP

This command identifies the name of the NetView operator (the name with which you logged onto NetView). This allows the Cleo software to determine the source of NetView commands it receives, and to include this name in the responses so that NetView can route them correctly.

This command must be issued before you use any other NetView commands.

Syntax

```
login-sp name (userid)
```

Output

NEW HOST OPERATOR IS *userid*

NO CHANGE TO HOST OPERATOR: *userid*
(this appears if you've already issued the command using this user ID)

Notes

userid is the user ID you used to log on to the NetView program.

Display Applications

This command lists the local Cleo components recognized by NetView.

Syntax

```
display -applicats
```

Output

SERVICE POINT - *netnode*

APPLICATION NETMAN

APPLICATION *nodename*

APPLICATION UNIX

APPLICATION *linkname*

Notes

No parameters or options are required with this command.

The SERVICE POINT *netnode* message shows the name of the local node which owns the Network Management connection.

The APPLICATION NETMAN and APPLICATION UNIX messages indicate the two other components to which NetView commands can be sent: the network management daemon and the UCF daemon. Note that APPLICATION UNIX will not be displayed if the configuration file specifies that UCF is not supported, or if the UCF daemon failed to start for any reason.

The APPLICATION *nodename* message appears once for each active local node, and the APPLICATION *linkname* message appears once for each active link service. These messages list the service components recognized by NetView.

Node Commands

Node commands allow you to view information on the currently active services and applications at a local node, and to activate and deactivate local node resources such as connections, 3270 LUs and APPC sessions.

The general syntax for node commands is:

```
runcmd sp=spname, appl=nodename, command [param(value)] [option=choice]
```

runcmd	This identifier indicates that the text that follows is a NetView command.
sp=spname	The name of your service point, as defined at NetView. Your host NetView personnel can give you this information.
appl=nodename	The name of the Cleo component to which NetView should send command. Node commands are sent to the local node that owns the resource to which the command applies. For example, to display information on a local APPC LU or its sessions, you use the name of the local node which owns the connection.
command	The actual NetView command; this is generally two keywords (an action and a component) separated by a hyphen.

One or more parameters or options may also be required.

param(value)	Parameters indicate data such as the names of software component or resources.
option=choice	Options indicate which of a set of mutually exclusive choices is to be used. Each option has a default choice which will be assumed unless you specify a choice.

Abbreviations

Several command names and component or resource names used in NetView commands can be abbreviated. The syntax of each command, shown in this chapter, uses the full name for clarity; the following list gives the minimum abbreviation (usually the first three or four letters) of all names which can be abbreviated.

NetView Commands

Any words not included in the table below cannot be abbreviated and must be specified in full.

Word	Abbreviation
ACTIVATE	ACT
CONNECTION	CONN
DEACTIVATE	DEA
DISPLAY	DISP
IMMED	IMM
SESSION	SES
TYPE	TY

Concurrent Use of NetView Commands

Only one node command may be executed at a time. If you issue a node command before a command previously issued by another user has completed, the second command will return with a **COMMAND IN PROGRESS** message. If this happens, wait a few seconds for the first command to complete, then try again.

The **COMMAND IN PROGRESS** message may also be returned if another user is using the Manage program.

Display APPC LU

This command displays information on an APPC local LU. It must be sent to the local node which owns the local LU.

Syntax

```
display-lu lu(lualias)
```

Output

```
DISPLAY LU FOR LU = netid. luname,  
LOCALLY KNOWN NAME = lualias,  
SESSION LIMIT = nnnn,  
SESSION COUNT = nnnn
```

```
PARTNER LUs - netid. luname netid. luname
```

```
DISPLAY OK
```

Notes

netid. luname is the fully-qualified LU name, consisting of the network ID and LU name as specified in the Cleo configuration.

NetView Commands

Display Partner LU

This command displays information on an APPC LU (local or remote) which is the partner of a local LU. It must be sent to the local node which owns the local LU.

Syntax

For a remote partner LU:

```
display-rlu lu(lualias) rluid(netid) rlunm(luname)
```

For a local partner LU:

```
display-rlu lu(lualias) rluid=NONE rlunm(luname)
```

Output

```
DISPLAY PARTNER LU = netid. luname,  
LOCALLY KNOWN NAME = rl.ualias,  
SESSION LIMIT= nnnn,  
PARALLEL SESSION SUPPORT = YES | NO
```

```
MODE NAMES - modename modename modename modename
```

```
DISPLAY OK
```

Notes

lualias is the alias of the local LU, as specified in the configuration.

rlualias is the alias of the partner LU.

netid. luname is the fully-qualified LU name, consisting of the network ID and LU name as specified in the Cleo configuration. For a local partner LU *netid* is not used.

Display Mode

This command displays information on APPC modes used by a local LU. You can specify a partner alias and/or mode name, or you can display information for all partner LUs and/or modes. The command must be sent to the local node which owns the local LU.

Syntax

```
display-mode lu(lualias) [rlu(rlualias)] [mode(modename)]
```

Output

```
DISPLAY MODE, PARTNER LU = netid. lunarne, MODE= modename,
SEND PACING = nnnn, RECEIVE PACING= nnnn,
SEND RU LOWER= nnnn, SEND RU UPPER= nnnn,
RECEIVE RU UPPER= nnnn, RECEIVE RU LOWER= nnnn,
AUTO-ACTIVATE = nnnn, MAX SESSION LIMIT= nnnn,
MODE SESSION LIMIT = nnnn,
MIN CON WINNERS= nnnn, MIN CON LOSERS = nnnn,
SESSION COUNT = nnnn, WINNERS = nnnn, LOSERS = nnnn
```

```
SESSION IDS =
sessid
sessid
```

DISPLAY OK

Notes

If you do not specify a partner LU alias and/or a mode, information is displayed on all partner LUs and/or modes associated with this local LU. For example, if you specify a partner LU alias but no mode, information is displayed on each mode associated with this pair of LUs. If you specify both the partner LU alias and the mode, information is displayed only on this combination of local LU, partner LU, and mode.

NetView Commands

Display TP

This command displays information on an invocable APPC or CPI-C TP. It may be sent to any local node (see Notes below).

Syntax

```
display-tp tp(tpname)
```

Output

```
DISPLAY CONVERSATIONS FOR TP ALIAS= tpalias  
FULL TP NAME= tpname
```

```
LU = netid.luname, PARTNER LU = netid.luname,  
MODE = modename, CONVERSATION TYPE= convtype,  
SESSION ID= sessid, STATE= convstate
```

```
DISPLAY OK
```

Notes

The first two lines always appear once. If the invocable TP is currently active, using an LU on the local node to which the command is sent, and involved in one or more conversations with other applications, the following three lines appear once for each conversation. (Information on any conversations using LUs on other local nodes is not shown.)

convtype is either MAPPED or BASIC.

convstate is the LU6.2 conversation state. See the *APPC Programmer's Guide* or the *CPI-C Programmer's Guide* for more information.

Display Link

This command displays information on a link. It must be sent to a local node which owns a connection configured to use the link (the connection need not be active).

Syntax

```
display-link link(linkname)
```

Output

```
DISPLAY LINK FOR linkname, TYPE= type, ROLE = role,  
STATUS = ( status1, status2 )
```

```
CONNECTIONS - conname conname conname conname
```

```
DISPLAY OK
```

Notes

type is the link type: SWITCHED (for switched SDLC or X.25 SVC or Token Ring) or LEASED (for leased SDLC or X.25 PVC).

role is the link role: PRIMARY, SECOND'Y (secondary), or NEG'ABLE (negotiable).

status1 indicates the current status: ACTIVE, INACTIVE or OUTAGE.

status2 indicates any change being made to the current status:

RESET	no change in progress
ACT	activation in progress
DEACT(N)	normal deactivation in progress
DEACT(I)	immediate deactivation in progress

conname is a list of all currently active connections owned by the local node to which the command is sent that are using the link service.

NetView Commands

Display Connection

This command displays information on a connection and on the remote node accessed by this connection. It must be sent to the local node which owns the connection.

Syntax

```
display-connection connection(conname)
```

Output

```
DISPLAY CONNECTION FOR conname, LINK SERVICE= linkname,  
REMOTE NODE TYPE = type, ADDRESS= addr,  
NETWORK = netwk, STATUS = (status1 , status2)
```

```
DISPLAY OK
```

Notes

type is PEER or SUBAREA.

addr is the poll address for the connection. It is used only for SDLC connections where the local node is configured as Primary; it is zero for other SDLC connections, X.25 connections and Token Ring connections.

netwk is the network name of the remote node, if known (this name will only be known if the connection is configured to send Format 3 XIDs); otherwise it is the name of the connection.

status1 indicates the current status: ACTIVE, PENDING (pending contact), INACTIVE or OUTAGE.

status2 indicates any change being made to the current status:

RESET	no change in progress
ACT	activation in progress
DEACT(N)	normal deactivation in progress
DEACT(I)	immediate deactivation in progress

Display Session

This command displays information on sessions for a local APPC LU with a specified partner LU (local or remote). You can specify a session ID to display information on a specific session, or you can display information on all sessions with the partner LU. The command must be sent to the local node which owns the local LU.

Syntax

For a remote partner LU:

```
display-session lu(lualias) rluid(netid) rlunm(luname) mode(modename)
[id(sessionid)]
```

For a local partner LU:

```
display-session lu(lualias) rluid=NONE rlunm(luname) mode(modename)
[id(sessionid)]
```

Output

```
DISPLAY SESSION, PARTNER LU = netid. luname,
MODE = modename,
ROLE = role, POLARITY = polarity,
SESSION STATE = (statel, state2),
RU SIZE= nnnn,
SEND PACING = nnnn, RECEIVE PACING= nnnn
```

DISPLAY OK

Notes

lualias is the alias of the local LU, as specified in the configuration.

role is either PRIMARY or SECOND'Y (secondary).

polarity is either CON-WIN (contention winner session for the local LU) or CON-LOSE (contention loser session for the local LU).

statel is BOUND, BINDING or UNBOUND.

state2 is FREE, IN USE or PEND ATT (pending ATTACH).

Display Old LU (3270, LUA or RJE)

This command displays information on 3270, LUA or RJE LUs, or on all such LUs configured on a particular node. The command must be sent to the local node which owns the LU, or the local node for which a display of all LUs is required.

Syntax

Display information on a specific LU:
`display-oldlu lu(luname)`

Display information for all 3270, LUA and RJE LUs that are configured on the local node to which the command was issued:
`display-old lu lu=all`

Output

```
DISPLAY LU FOR LU = luname, POOL= poolname,  
ADDRESS = addr, TYPE = type (lotype),  
CONNECTION = conname, HOST APPL = applname,  
SSCP SESSION STATUS= status1,  
LU STATUS= status2,  
PLU SESSION STATUS= status3
```

DISPLAY OK

Notes

poolname is displayed for 3270 and LUA LUs only (if the LU is in a pool).

addr is the configured LU number, which matches the LU address at the host.

type and *lotype* indicate the LU type. For a 3270 LU, *type* is DISPLAY or PRINTER, and *lotype* is 01 (SCS printer), 02 (display) or 03 (DSC printer). For an LUA LU, *type* is LUA, and *lotype* is 00,01,02 or 03 (the LU type specified on the BIND from the host). For an RJE LU, *type* is RJE, and *lotype* is 01 (SCS). Note that the *lotype* parameter is only valid if the PLU session status is BOUND.

applname is the name of the host application (for RJE, the Job Entry Subsystem) to which the LU is bound (not displayed if the LU is not bound to a host application).

status1 may be ACTIVE or INACTIVE.

status2 may be IN USE or UNUSED.

status3 may be BOUND, BINDING, UNBINDING or UNBOUND.

NetView Commands

Display Old LU Pool (3270 or LUA)

This command displays information on a 3270 LU pool or an LUA LU pool. It must be sent to a local node which owns one or more LUs in the pool; note that it will only display information on the LUs in the pool which are owned by this local node.

Syntax

```
display-pool pool(poolname)
```

Output

DISPLAY LU POOL *poolname*, TYPE = *type*

LUS : *luname luname luname luname*

DISPLAY OK

Notes

type is either DISPLAY (for a 3270 LU pool) or LUA.

Activate Connection

This command activates a connection. The command must be sent to the local node which owns the connection.

Syntax

```
activate-connection [link(linkname)] connection(conname)
```

Output

COMMAND OK AS SPECIFIED

Notes

If you do not specify a link name, the connection is activated using any available link configured for it (if more than one link service is configured). You only need to specify the link name if you want to activate the connection using a specific link.

If you activate a connection that's configured to be activated "on demand" (activated automatically when required by a session, and deactivated when all sessions have stopped using it), then the session effectively becomes an "operator started" connection; it is no longer started or stopped when required by a session and must be activated or deactivated using NetView commands or the Manage program. This change remains in effect until you restart the Cleo software.

The output message "OK AS SPECIFIED" indicates that the command syntax was correct and the command has been issued correctly; it does not necessarily mean that the connection has successfully been activated.

Deactivate Connection

This command deactivates a connection. The command must be sent to the local node which owns the connection.

Syntax

```
deactivate-connection connection(conname) [type=deacttype]
```

Output

COMMAND OK AS SPECIFIED

Notes

deacttype can be:

norm	normal deactivation, wait for sessions on this connection to end before deactivating the connection (default)
immed	immediate deactivation, deactivate the connection immediately without waiting for sessions to end

The OK AS SPECIFIED message indicates that the command syntax was correct and the command has been issued correctly; it does not necessarily mean that the connection has successfully been deactivated.

If you deactivate a connection that's configured to be activated "on demand" (activated automatically when required by a session and deactivated when all sessions have stopped using it), then the session effectively becomes an "operator started" connection; it is no longer started or stopped when required by a session and must be activated or deactivated using NetView commands or the Manage program. This change remains in effect until you restart the control daemon.

Activate Session

This command activates an APPC session between a local APPC LU and its partner LU (which can be either a remote LU or another local LU). The command must be sent to the local node which owns the local LU.

Syntax

```
activate-session lu(lualias) rlu(rlualias) mode(modename)
```

Output

```
SESSION ACTIVATED, POLARITY = polarity,  
SESSION ID= hhhh hhhh hhhh hhhh
```

```
SESSION LIMIT EXCEEDED
```

Notes

polarity is either CON-WIN (contention-winner session for the local LU) or CON-LOSE (contention-loser session for the local LU).

SESSION LIMIT EXCEEDED is displayed if the session cannot be activated because the session limit has been reached.

Deactivate Session

This command deactivates an active APPC session for a local APPC LU. The command must be sent to the local node which owns the local LU.

Syntax

```
deactivate-session lu(lualias) rlu(rlualias) mode(modename)  
[id(sessid)] [type=type]
```

Output

SESSION DEACTIVATED

Notes

The **id** (*sessid*) parameter, which specifies the ID of the session to be deactivated, is required only if the LUs support parallel sessions. It must be entered as a 16-character string representing 8 hexadecimal bytes (use the characters 0-9 and A-F).

type is the type of deactivation required:

norm	normal deactivation, wait until the current conversation on the session ends before deactivation (default)
immed	immediate deactivation, do not wait for end of conversation

Deactivate Old LU (3270, LUA, or RJE)

This command deactivates an active 3270, LUA or RJE LU. The command must be sent to the local node which owns the LU.

Syntax

```
deactivate-oldlu lu(luname)
```

Output

SESSION DEACTIVATED (if the LU was bound, that is, communicating with a host application)

RESOURCE INACTIVE (if the LU was not bound)

The UNIX Command Facility (UCF)

The UNIX Command Facility (UCF) allows a NetView operator to issue UNIX commands, from the NetView console, that execute on the local UNIX computer. The NetView operator will also be able to view any output returned by these commands. This allows a remote operator to monitor activity on the local UNIX computer, to diagnose problems and in some cases to take corrective action.

The UNIX commands are processed by the UCF daemon program, which runs on the same server that the network management daemon program is running on and the commands are issued on this server. (If you are using the Cleo software as a standalone product and not in a client/server setting, the machine that Cleo is installed on is the server.) UNIX commands received from the UCF are processed by starting a new UNIX shell for each command and running the command in that shell.

The confirmation specifies the name of the UCF user, which must be a valid login name on the UNIX server on which the network management daemon is running; the UCF shell is started using the shell program, login ID, permissions and *.login* or *.profile* specified for that user.

The UCF facility does not restrict the UNIX commands that can be issued. This means that the normal UNIX security features must be used to restrict the UCF user's access to files and commands.

Before you can issue UNIX commands from NetView, you need to use the Login SP command to establish communication so that the commands can be routed correctly. See **Network Access Commands** for details on the Login SP command.

The general syntax of UNIX commands is:

runcmd sp=spname, appl=unix, command

runcmd	This identifier indicates that the text that follows is a NetView command.
sp=spname	The name of your service point, as defined at NetView. Your host NetView personnel can give you this information.
appl=unix	The name of the Cleo component to which NetView should send the command. UNIX commands are always sent to the

UNIX Command Facility daemon (**unix**).

command The text of the UNIX command to be issued, as you would enter it on a UNIX terminal. Uppercase characters must be preceded by a backslash character “\”.

Uppercase Characters and Escape Characters

Although UNIX distinguishes between upper- and lowercase alphabetic characters, the NetView program does not (it translates all characters into uppercase before sending them to the UNIX computer). Also, the host character set may not support the square brackets “[” and “]”, which is required with some UNIX commands.

UCF provides support for uppercase characters and square brackets using the backslash character ‘\’. To include an uppercase character in the UNIX command string, precede the character with a backslash; any alphabetic character not preceded by a backslash is interpreted as lowercase. To include the square brackets “[” and “]”, use the sequences “\([” and “\)” respectively. To include the backslash character ‘\’ itself, type it twice. If a single backslash is followed by any other non-alphabetic character, the backslash is ignored and the character is left unchanged.

Some examples are shown below.

To include	Type
ABcd	\a\bcd
\a	\\a
[]	\([\)
\[	\\([

Example

The following is an example of a UNIX command, as you would enter it from the NetView operator console:

```
runcmd sp=mypname, appl=unix, grep \temp \ (ab\ ) *. c > \t\e\m\p. out
```

The command which would run on the UNIX computer is:

```
grep Temp [ab]'. c > TEMP. out
```

Output from UNIX Commands

When a command is successfully issued, the following messages are displayed on the NetView screen:

EXECUTING UNIX COMMAND

(any output from the command, including error messages)

UNIX COMMAND COMPLETED

Note that these messages may not appear on the NetView screen at the same time. The EXECUTING UNIX COMMAND message appears as soon as the UCF daemon receives the command and returns control to the NetView operator. Any output from the command is sent to NetView as it is produced and may appear as a series of separate messages. The UNIX COMMAND COMPLETED message appears when the UNIX command has finished and its shell has ended.

If you issue a UNIX command before a previous command has completed (before the UNIX COMMAND COMPLETED message is received), the COMMAND IN PROGRESS message is displayed and the second command is ignored. Before issuing the command again, wait for the first command to complete, or cancel the first command (see **Canceling a Command**, later in this section).

Valid UNIX Commands

The UCF is designed for use with commands that complete without any further interaction from the user (whether or not any output is produced). For example, you can issue the *cat* command which completes after displaying the contents of a file, or the *mv* command which completes with no output unless an error occurs.

The UCF does not provide a means of sending keystrokes to an interactive process started by a UNIX command. For example, if you issue the *vi* command with a filename (*vi filename*) the first screen of the file is displayed but there is no mechanism for sending further keystrokes to *vi* to move within the file, edit it, save it or exit the program. The only action you can take at this point is to issue the command that terminates the shell running *vi* (*ux-cancel*).

Because all UNIX commands run with the login ID and permissions of the configured UCF user, the valid commands for that user are limited by the access rights of the UCF user's login. In particular, root or superuser commands are not permitted. See **UCF Security** for more information.

Note that only one UNIX command may be outstanding at a time; if you issue another command before the previous one has completed, an error message is returned. If you do not wish to wait for the output to each individual command, you can use "&" at the end of each command, as you would on a standard UNIX terminal to run the command in the background. In this case, output from different commands may appear interleaved (as it would from a series of background commands issued on the same terminal).

Canceling a Command

The UCF provides a command for canceling the last UNIX command issued while it is still in progress (`ux-cancel`). This command can be used to stop the current command from executing, or to cancel an interactive command which will not complete without further input; it is equivalent to using **CTRL + C** to stop a process running on a terminal, or using a UNIX *kill* command to stop the process.

The command syntax is the same as for any other UNIX command:

```
runcmd sp=spname, appl=unix, ux-cancel
```

If the command was canceled successfully, the message CANCELING UNIX COMMAND is displayed. The UNIX shell in which the previous command was running has been stopped. Further UNIX commands may be issued as necessary.

If a UNIX command starts a daemon process on the UNIX computer, this process may not be stopped by this command. It may be necessary to use the *kill* command (either on a terminal or by using UCF) to stop such a process explicitly.

If no UNIX command is running when the cancel command is used, the NO OUTSTANDING COMMANDS message is displayed, and the cancel command is ignored. No further action is necessary. This may occur if the cancel command is issued after the previous command has finished but before the "UNIX COMMAND COMPLETED" message is received.

UCF Security

Because the UCF allows a remote operator to issue commands on the UNIX computer and to receive output from these commands, it is important to consider the security implications. For example, you need to ensure that the operator cannot access private information or issue UNIX commands which may disrupt other users.

NetView Commands

These security considerations require you to configure a specific UNIX user name as the UCF user; all UCF commands are executed with this user's ID and, therefore, with the access permissions of this user.

It is intended that you use the normal security features provided by UNIX to restrict the commands that are accessible by the UCF user, in order to allow only those commands you consider reasonable for use from UCF. The following guidelines may be useful:

- ◆ The UCF user name should be one which is used solely for UCF; you should not use an existing login which is also used for other purposes. This makes it easier to define the privileges of this user to include only those which are reasonable for UCF, it also allows you to identify processes which were started using UCF.
- ◆ You may need to restrict the users and groups to which the UCF user can change user ID or group ID. In particular, the UCF *user must not be allowed to become root or superuser.*

3. Network Management API

This chapter covers the following topics:

- ◆ Overview of the Network Management API (NM-API)
- ◆ NM-API verbs in C programs
- ◆ Designing and writing NM-API applications
- ◆ NM-API verb reference
- ◆ Header file listing

Overview of the Network Management API

The Network Management API (NM-API) is an application programming interface which allows Cleo applications, running on a UNIX client or server (or standalone system), to communicate with the host NetView program. An NM-API application program can receive complete Network Management Vector Transport messages (NMVTs) sent from NetView, and can return application data in the form of NMVTs back to NetView. Each received NMVT is logged in the audit log file; each NMVT supplied to NM-API by the application is also logged (although NM-API may subsequently detect an error and not send the NMVT).

To allow the use of NM-API applications, the configuration file must specify that NetView commands and alerts are supported, and must specify a Network Management connection (a connection to the NetView host). For more information, see chapter 10 of the *Administration Guide*.

This chapter does not attempt to describe the syntax or usage of NMVTs. For information on NMVT syntax, refer to the IBM manual *SNA Formats*; for information on the usage of NMVTs with the NetView program, refer to the documentation for the version of NetView you are using or contact your host personnel.

Summary of NM-API verbs

The following paragraphs provide a summary of the four NM-API verbs. More detailed information is given in later sections of this chapter.

NM_CLOSE

NM_CLOSE informs the local node that the application has finished using the API. The application should issue this verb when completing NM-API communications, in order to clean up resources at the local node.

NM_OPEN

NM_OPEN connects the application to the local node that owns the Network Management connection, and identifies it as an NM-API application. The application must successfully issue this verb before it can use any other verb.

NM_RECEIVE

NM_RECEIVE receives an incoming NMVT from NetView, if one is available. If no incoming NMVT is available when the verb is issued, the verb may either wait for one or return immediately indicating that no incoming data was available. Any incoming NMVT is logged in the audit log file.

NM_SEND

NM_SEND supplies an NMVT to be sent to NetView (optionally requesting acknowledgement). The supplied NMVT is logged in the audit log file.

NM-API Verbs in C Programs

An application accesses the NM-API library through a single entry point, the C function `NMAPI_C`. This has a single parameter: a pointer to the Verb Control Block (VCB). The VCB contains the data that identifies the verb to be issued, as well as other input parameters. On return from the function, return codes are also placed in the VCB.

VCB Structure

The verb control block structure for all verbs is as follows.

Field	Type	Length	Notes
opcode	unsigned short	2	Verb operation code specifying the verb to be issued
options	unsigned char	1	Options (see individual verb descriptions)
retcode	unsigned char	1	Return code
dptr	unsigned char *	4	Pointer to a buffer containing the data to be sent, or in which data will be returned
datalength	unsigned short	2	Length in bytes of the data sent or received
sess_status	unsigned char	1	Status of the application's communications with the host
origntr_id	unsigned char[8]	8	Application identifier

Issuing a Verb

To issue a verb:

1. Create a VCB structure variable.

```
#include <nmap_i_c.h>
```

```
NM_VCB local_vcb;
```

The VCB structure NM_VCB is declared in the header file *nmap_i_c.h*; a listing of this file is included at the end of this chapter.

2. Clear (set to zero) the variables within the structure.

```
memset (local_vcb, ' \0' , sizeof ( nm_vcb ) ) ;
```

Although not required, this step eliminates the danger of future versions of a verb being sensitive to fields that are ignored in the current version.

3. Assign values to the VCB fields.

```
local_vcb.opcode      = NM_RECEIVE ;  
local_vcb.dptr        = bufptr;  
local_vcb.datalength  = 500;
```

This example sets up a receive verb, with data to be returned in a buffer pointed to by **bufptr**.

4. Invoke the verb.

```
NMAPI_C(&local_vcb) ;
```

5. Use the return codes.

```
if ( local_vcb.retcode != NM_OK )  
{  
    /* error handling */  
  
}
```

Designing and Writing NM-API Applications

This section lists some of the considerations you need to be aware of when writing NM-API applications.

Message Filtering

The destination of an NMVT sent from NetView is specified by the Application Name Subfield (part of the Namelist Subvector). NMVTS can be sent to the network management daemon (application name NETMAN), to a local node (application name is the node name) or the UNIX Command Facility (application name is UNIX), as well as to NM-API applications.

All incoming messages are examined by the NM-API library to determine whether or not to pass them to the application. Messages not sent to the application are ignored, but may be handled by other applications or by a local software component. There are two cases in which a message may be rejected:

- ◆ The NM_FILTER flag is set. In this case, messages are rejected unless they are NMVTS with the Application Name Subfield matching the **origintr_id** supplied on the NM_OPEN verb (see the description of this verb below).
- ◆ The message is a NMVT with an Application Name Subfield which matches one of the known component names (NETMAN, UNIX, or the name of a local node).

For secure operation, the application should use the NM_FILTER option, and the NetView operator should set the Application Name Subfield to the originator ID supplied by the application on the NM_OPEN verb. This ensures that the application will not receive messages intended for other applications or for other software components.

If the application does not use this option, it will receive all incoming NMVTs except those intended for the network management daemon, local node or UNIX Command Facility; it must determine which messages it is responsible for.

Replying to Received Messages

The application must reply to any received messages which require a response; that is, after issuing an NM_RECEIVE which returns a command NMVT intended for this application, the application should issue at least one NM_SEND containing the

Network Management API

reply to the command. If this is not done, the NetView console application may hang while awaiting a reply to its command. This may disrupt the operation of other components, such as the network management daemon.

NMVTs sent by the application should not have the sequence field flags (which indicate first, middle or last NMVT) set. The first response to an incoming message should be a Reply message; further replies should be sent as unsolicited messages.

The application may also send Alert NMVTs; these should be sent as unsolicited messages.

Note that the NM-API library does not perform any syntax checking on NMVTs supplied by the application before sending them to NetView; it is up to the application to ensure that any NMVTs it sends are valid for the version of NetView in use at the host.

Multiple NM-API Applications

When multiple applications are using the NM-API library, only one should issue replies to an incoming message. This can be done by specifying a different originator ID for each application, and using the NM_FILTER option on the NM_OPEN verb to filter on this ID. If multiple applications send responses to the same NMVT, this may disrupt the NetView console and other Cleo software components.

Use of Signals

The UNIX **sigpoll** signal, which maybe passed to the NM-API library, is handled by a signal catcher, set up on the NM_OPEN call. Applications may require this signal for other purposes, so the library daisy-chains to any existing signal catchers.

Applications which need to add signal catchers should preserve the address of the active signal catcher (returned by the **sigset** call) and daisy-chain by calling this address from within the new signal catchers.

Applications must not issue any further NM-API function calls from within a signal catcher.

NM_CLOSE

The NM_CLOSE verb informs the local node that the application has finished communications with NetView. This call should always be issued before the application terminates, or when no further communication is required.

Supplied Parameters

opcode NM_CLOSE

Returned Parameters

retcode NM_OK
The verb executed successfully.

 NM_INVALID_VERB
An invalid value was supplied for the **opcode** parameter.

 NM_NOT_OPENED
The NM_OPEN verb has not been issued by this application.

Usage and Restrictions

This must be the last verb issued by the application. After this verb has completed with a return code of NM_OK, the only valid NM-API verb is NM_OPEN (to restart communications with NetView); no other verbs may be issued.

NM_OPEN

The NM_OPEN verb enables communication between the NM-API library and the local node that owns the Network Management connection. This must be the first verb issued by the application; the other NM-API verbs cannot be issued until NM_OPEN has completed successfully.

Supplied Parameters

opcode	NM_OPEN
origintr_id	An identifier of up to eight ASCII characters for the application. This identifier is used when logging data messages, and for message filtering (see the “Options” parameter below). It must not match the reserved names NETMAN or UNIX, or the name of any local node in the current configuration. Since the identifier is used when logging messages sent by this application, you should also avoid using an identifier which clashes with the name of a link, invocable TP or 3270 user in the current configuration.
options	NM_FILTER Filter incoming messages. An incoming message is only forwarded to the application if it is a NMVT with an Application Name Subfield which matches origintr_id supplied on the NM_OPEN. NM_NONE Do not filter incoming messages. All incoming messages (except those intended for NETMAN, UNIX, or local nodes) are forwarded to the application; it is up to the application to determine whether it is responsible for a particular message.

Returned Parameters

retcode	NM_OK The verb executed successfully.
	NM_ALREADY_OPEN This application has already issued the NM_OPEN verb successfully.
	NM_COMM_SUBSYSTEM_NOT_LOADED The Cleo software is not running. No NM-API verbs can be issued until it is active; the application will need to issue NM_OPEN again when the software has been started.
	NM_INVALID_OPTIONS An invalid value was supplied for the option parameter.
	NM_INVALID_VERB An invalid value was supplied for the opcode parameter.
	NM_NODE_UNAVAILABLE The local node that owns the Network Management connection is not running. No NM-API verbs can be issued until it is active; the application will need to issue NM_OPEN again when the local node is started.
	NM_OPEN_FAILED The application failed to connect to the local node (the local node sent an error response).
	NM_OPEN_TIMEOUT The application failed to connect to the local node (the local node did not reply within ten seconds). This error code will be returned if no Network Management connection is configured.
	NM_OPEN_TOO_MANY_APPS This return code is unlikely to occur in normal use. The maximum number of NM-API applications are already running; the application may retry later when other NM-API applications have ended.

Network Management API

sess_status	NM_SESSION_ACTIVE
	Communications with the host have been established.
	NM_SESSION_LINK_ERROR
	An error has occurred on the communications link to the host.
	NM_SESSION_PU_INACTIVE
	Communication with the host has not yet been established; the network management daemon may not be running, or the connection to the host may not have been started.

Usage and Restrictions

This must be the first verb issued by the application. No other verbs may be issued until this verb has completed with a return code of NM_OK.

If the attempt to contact the local node fails, the application may retry by issuing the NM_OPEN verb again. Note that NM_OPEN will continue to fail (with return code NM_OPEN_TIMEOUT) if no network management connection is configured.

The application is responsible for determining which messages it must process (unless it uses the NM_FILTER option to filter incoming messages) and replying to any messages which require a response. Disruption to other network management applications may occur if replies are not sent when required, or if more than one application replies to the same message. See **Designing and Writing NM-API Applications**, earlier in this chapter, for more information.

After the application has finished communication with the NetView program, it should issue the NM_CLOSE verb to free resources at the local node.

NM_RECEIVE

The NM_RECEIVE verb attempts to receive an incoming message from NetView (either an NMVT or an acknowledgement for a previous NM_SEND verb). If no message is available then the function may return immediately, or wait until a message is received. The function may also return if the status of the connection changes.

Any incoming NMVT received by the application is logged in the audit log file.

Supplied Parameters

opcode	NM_RECEIVE
options	NM_NONE Do not wait for messages. If no incoming message is available, the verb returns immediately. The return code indicates whether a message was returned. NM_RCV_WAIT_MSG Wait for messages. The verb does not return until one of the following occurs: ◆ An incoming message is available ◆ Communication status (sess_status) is PU_INACTIVE or LINK_ERROR ◆ The verb does not complete because an error occurs NM_RCV_WAIT_IGNORE_STATUS Wait for messages and ignore communication status. As for NM_RCV_WAIT_MSG, except that any changes in the session status are ignored; the verb waits until an incoming message is available.
dptr	A pointer to a data buffer to hold the received NMVT data.
datalength	The length of the application buffer pointed to by dptr . This defines the maximum number of bytes of data which will be returned.

Network Management API

Returned Parameters

Successful execution

If the verb executes successfully, the following parameters are returned.

retcode	NM_OK A message was received, and is in the buffer specified by dptr .
	NM_OK_DATA_TRUNCATED A message was received, but the buffer length specified by datalength was not large enough to hold all the data. The message is in the buffer specified by dptr , truncated to datalength bytes.
	NM_ACK_RECEIVED An acknowledgement message was received, indicating that the previous NM_SEND was successful.
	NM_NACK_RECEIVED A negative acknowledgement message was received. This indicates that the host rejected the data sent by the previous NM_SEND verb. The sense codes from the NACK-1 message returned by the host (4 bytes) are returned in dptr .
	NM_RCV_NO_MESSAGE No message was received. If the option parameter specified NM_NONE, this return code means there are no new received messages; if it specified NM_RCV_WAIT_MSG, this return code means the session status is inactive. This return code will not be returned if the option parameter specified NM_RCV_WAIT_IGNORE_STATUS.
data_length	Length of data returned in the buffer specified by dptr (in bytes).

Unsuccessful execution

The following indicates that the verb did not complete successfully.

retcode	NM_COMM_SUBSYSTEM_NOT_LOADED The Cleo software is not running. No NM-API verbs can be issued until it is active; the application will need to issue NM_OPEN again when the software has been started.
	NM_INVALID_OPTIONS An invalid value was supplied for the option parameter.
	NM_INVALID_VERB An invalid value was supplied for the opcode parameter.
	NM_NODE_UNAVAILABLE The local node that owns the Network Management connection is not running. No NM-API verbs can be issued until it is active; the application will need to issue NM_OPEN again when the local node is started.
	NM_NOT_OPENED The application has not issued the NM_OPEN verb, or has issued the NM_CLOSE verb.

Session Status

The following values indicate the status of communications between the application and the host.

sess_status	NM_SESSION_ACTIVE Communications with the host are established.
	NM_SESSION_LINK_ERROR An error has occurred on the communications link to the host. This return code will not occur if the option parameter specified NM_RCV_WAIT_IGNORE_STATUS. The application may either retry the NM_RECEIVE until the session status becomes active, or issue NM_RECEIVE with an option parameter of NM_RCV_WAIT_IGNORE_STATUS to wait indefinitely until the status becomes active and a message is received.

Network Management API

NM_SESSION_PU_INACTIVE

Communications with the host have not yet been established; the network management daemon may not be running, or the connection to the host may not have been started. This return code will not occur if the **option** parameter specified NM_RCV_WAIT_IGNORE_STATUS.

The application may either retry the NM_RECEIVE until the session status becomes active, or issue NM_RECEIVE with an **option** parameter of NM_RCV_WAIT_IGNORE_STATUS to wait indefinitely until the status becomes active and a message is received.

Usage and Restrictions

The returned data is a complete NMVT; the input buffer should be 512 bytes long to allow for the maximum length of an NMVT.

If the NM_FILTER option was used on the NM_OPEN verb, the NMVT will be one which is intended for this application (the originator ID on the NMVT matches the **origintr_id** parameter specified on the NM_OPEN verb); otherwise it is up to the application to determine whether it is responsible for this NMVT.

The application is responsible for replying to any messages which require a response. Disruption to other network management applications may occur if replies are not sent when required, if invalid replies are sent, or if more than one application replies to the same message. See **Designing and Writing NM-API Applications**, earlier in this chapter, for more information.

NM_SEND

The NM_SEND verb supplies an NMVT to NM-API for sending to NetView. The supplied NMVT is logged in the audit log file, with the application name specified in the **origintr_id** parameter of the NM_OPEN verb. Note that this logging occurs before NM-API attempts to send the NMW, the log does not guarantee that the NMVT will be sent.

Supplied Parameters

opcode	NM_SEND
options	NM_NONE No acknowledgement of the sent message is required. NM_SEND_ACK The application requires an acknowledgement message from the host for the data sent. See Usage and Restrictions below for more information.
datalength	The length of the NMVT data to be sent. This must not exceed 512 bytes.
dptr	The address of the buffer holding the data to be sent.

Returned Parameters

retcode	NM_OK The verb executed successfully. NM_COMM_SUB SYTEM_NOT_LOADED The Cleo software is not running. No NM-API verbs can be issued until it is active; the application will need to issue NM_OPEN again when the software has been started. NM_DATA_SIZE_ERROR The datalength parameter exceeds 512 bytes. NM_INVALID_OPTIONS An invalid value was supplied for the option parameter.

Network Management API

NM_INVALID_VERB

An invalid value was supplied for the **opcode** parameter.

NM_NODE_UNAVAILABLE

The local node that owns the Network Management connection is not running. No NM-API verbs can be issued until it is active; the application will need to issue NM_OPEN again when the local node is started.

NM_NOT_OPENED

The application has not issued the NM_OPEN verb, or has issued the NM_CLOSE verb.

NM_SEND_AWAITING_RESPONSE

The application has issued a previous NM_SEND verb specifying that an acknowledgement is required, but has not yet received the acknowledgement. It must continue to issue NM_RECEIVE until it receives either a positive or a negative acknowledgement before it can issue another NM_SEND.

NM_SEND_SESSION_INACTIVE

Communications with the host are not established. The application may retry the NM_SEND until the session becomes active.

sess_status

NM_SESSION_ACTIVE

Communications with the host are established.

NM_SESSION_LINK_ERROR

An error has occurred on the communications link to the host. The application may retry the NM_SEND periodically until the session status becomes active.

NM_SESSION_PU_INACTIVE

Communications with the host have not yet been established; the network management daemon may not be running, or the connection to the host may not have been started. The application may retry the NM_SEND periodically until the session status becomes active.

Usage and Restrictions

If the application uses the value NM_SEND_ACK for the **option** parameter, to request an acknowledgement from the host, it must issue NM_RECEIVE to get the response before issuing another NM_SEND. The NM_RECEIVE may return an NMVT, as there may be a delay before the acknowledgement is received. The application should reissue NM_RECEIVE until the send response (either NM_ACK_RECEIVED or NM_NACK_RECEIVED) is returned.

The application is responsible for determining which incoming messages it is responsible for (unless it uses the NM_FILTER option to filter incoming messages), and using NM_SEND to reply to any messages which require a response. Disruption to other network management applications may occur if replies are not sent when required, or if more than one application replies to the same message. See **Designing and Writing NM-API Applications**, earlier in this chapter, for more information.

The supplied data must be a complete NMVT of up to 512 bytes. Note that the NM-API library does not check the syntax of the supplied NMVT; it is up to the application to ensure that the data is valid for the version of NetView in use at the host.

NM-API Sample Program

In the `/usr/lib/cleo/samples` directory is a sample NM-API application, `nmapitest.c`. This program, which allows you to interactively test every NM-API verb, is shown below.

```
/* Version 100.9 */

/**MOD+*****/
/* Module:      nmapitest.c                                */
/*                                                     */
/* Purpose:     Test program for NM-API interface.        */
/*                                                     */
/* Copyright (c) 1989 - 1995 Interface Systems, Inc.     */
/*                                                     */
/**MOD-*****/

#include <stdio.h>
#include <ctype.h>

#include <svigsig.h>
#include <svgistr.h>

#include <btypes.h>

#include <nmap_i_c.h>

#define ptrc    sna_ptrc

char          *new_gets();
void          sessstat();
unsigned char  databuff[600];
static void   (*oldsig)();
char          signalset = FALSE;

unsigned char NMVTalert[156] = {
0x41,0x03,0x8D,0x00, 0x00,0x00,0x00,0x00,
0x00,0x94,0x00,0x00, 0x24,0x05,0x22,0x10,
0x00,0x09,0xC9,0xE2, 0xC3,0xC4,0xF0,0xF5,
0xD7,0xE4,0x00,0x81, 0x09,0xC2,0xC1,0xE2,
0xC5,0xD4,0xE2,0xF2, 0x40,0x00,0x3B,0x07,
0xC1,0xC4,0xC1,0xD7, 0xE3,0xF1,0x00,0xF7,
0x0B,0x92,0x00,0x00, 0x01,0x33,0x00,0x9D,
0x4F,0xB5,0x6B,0x04, 0x93,0x21,0x04,0x1D,
0x94,0x04,0x01,0x02, 0xA1,0x06,0x82,0x91,
0x86,0x01,0x72,0x06, 0x81,0x00,0xC0,0x01,
0x01,0x02,0x82,0x09, 0x82,0x91,0x33,0x11,
0xC3,0xD6,0xD4,0xD7, 0x0C,0x95,0x06,0x01,
0x10,0x23,0xF0,0x14, 0x04,0x81,0x20,0x11,
0x0A,0x96,0x04,0x01, 0x20,0x01,0x04,0x81,
0x01,0x01,0x2A,0x10, 0x00,0x27,0x11,0x0C,
0x08,0x04,0xF0,0xF2, 0xF0,0xF0,0xF0,0xF0,
0x09,0x08,0xE4,0xE2, 0xC5,0xD9,0xF0,0x40,
0x40,0x13,0x06,0xE2, 0xD5,0xC1,0xD7,0x60,
0xC9,0xE7,0x40,0xE2, 0xE2,0x40,0x40,0x40,
0x40,0x40,0x40,0x40 };

unsigned char NMVTresp[63]={
0x41,0x03,0x8D,0x00, 0x00,0x00,0x00,0x80,
0x00,0x16,0x00,0x61, 0x12,0x44,0x04,0x01,
0x00,0x01,0x06,0x10, 0x00,0x00,0x00,0x01,
0x06,0x11,0x00,0x00, 0x02,0x00,0x00,0x21,
0x13,0x00,0x1d,0x31, 0xe2,0xd6,0xd3,0xc9, /* ....soli */
0xc3,0xc9,0xe3,0xc5, 0xc4,0x40,0xd9,0xc5, /* cited.re */
0xe2,0xd7,0xd6,0xd5, 0xe2,0xc5,0x40,0xc6, /* sponse.f */
};
```

```

0xd9,0xd6,0xd4,0x40, 0xd5,0xd4,0x40 };          /* rom.nm.. */

/* The above NMVT can be extended at the end; the length of the following */
/* text is stored in the 31st and 32nd bytes */

unsigned char NMVTusresp[50]={
0x41,0x03,0x8D,0x00, 0x00,0x00,0x00,0x80,
0x00,0x10,0x00,0x6f, 0x0c,0x06,0x0a,0x50,
0xe4,0xc4,0xc3,0xd5, 0xd7,0xe6,0xc1,0x40,
0x00,0x1a,0x13,0x00, 0x16,0x31,0xe4,0xd5, /* .....un */
0xe2,0xd6,0xd3,0xc9, 0xc3,0xc9,0xe3,0xc5, /* sollicite */
0xc4,0x40,0xd9,0xc5, 0xe2,0xd7,0xd6,0xd5, /* d.respon */
0xe2,0xc5 }; /* se */

unsigned char NMVTlong[348]={
0x41,0x03,0x8D,0x00, 0x00,0x00,0x00,0x80,
0x00,0x10,0x00,0x6f, 0x0c,0x06,0x0a,0x50,
0xe4,0xc4,0xc3,0xd5, 0xd7,0xe6,0xc1,0x40,
0x01,0x44,0x13,0x00, 0xa0,0x31,0xe4,0xd5, /* .....un */
0xe2,0xd6,0xd3,0xc9, 0xc3,0xc9,0xe3,0xc5, /* sollicite */
0xc4,0x40,0xd9,0xc5, 0xe2,0xd7,0xd6,0xd5, /* d.respon */
0xe2,0xc5,0x40,0x40, 0x40,0x40,0x40,0x40, /* se..... */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xf0,0x40, /* line.00. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xf1,0x40, /* line.01. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xf2,0x40, /* line.02. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xf3,0x40, /* line.03. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xf4,0x40, /* line.04. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xf5,0x40, /* line.05. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xf6,0x40, /* line.06. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xf7,0x40, /* line.07. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xf8,0x40, /* line.08. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xf9,0x40, /* line.09. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xc1,0x40, /* line.0a. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xc2,0x40, /* line.0b. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xc3,0x40, /* line.0c. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xc4,0x40, /* line.0d. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xc5,0x40, /* line.0e. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf0,0xc6,0x40, /* line.0f. */
0xc5,0xd5,0xc4,0x40, 0xa0,0x31,0xe4,0xd5, /* end...un */
0xe2,0xd6,0xd3,0xc9, 0xc3,0xc9,0xe3,0xc5, /* sollicite */
0xc4,0x40,0xd9,0xc5, 0xe2,0xd7,0xd6,0xd5, /* d.respon */
0xe2,0xc5,0x40,0x40, 0x40,0x40,0x40,0x40, /* se..... */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xf0,0x40, /* line.10. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xf1,0x40, /* line.11. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xf2,0x40, /* line.12. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xf3,0x40, /* line.13. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xf4,0x40, /* line.14. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xf5,0x40, /* line.15. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xf6,0x40, /* line.16. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xf7,0x40, /* line.17. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xf8,0x40, /* line.18. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xf9,0x40, /* line.19. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xc1,0x40, /* line.1a. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xc2,0x40, /* line.1b. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xc3,0x40, /* line.1c. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xc4,0x40, /* line.1d. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xc5,0x40, /* line.1e. */
0xd3,0xc9,0xd5,0xc5, 0x40,0xf1,0xc6,0x40, /* line.1f. */
0xc5,0xd5,0xc4,0x40 }; /* end. */

/**PROC+*****
/* Name:          testopen
/*
/* Purpose:      To test the NM_OPEN verb.
/*
/* Returns:      None.
/*
/* Params:       None.
/*
/* Operation:
/*
/*
/**PROC-*****

```

Network Management API

```
void testopen()
{
    NM_VCB block;
    char *result;
    char buff[10];
    int i,j;

    /*****
    /* Now initialise vcb */
    *****/
    block.opcode=NM_OPEN;
    printf("Options? \n");
    printf("      NM_FILTER = %d\n",NM_FILTER);
    printf("      NM_NONE   = %d\n",NM_NONE);
    printf("Enter option required : ");
    result=new_gets(buff);
    block.options=atoi(result);
    block.retcode=0x00;
    block.dptr=0x00;
    block.datalength=0x00;
    block.sess_status=0x00;
    printf("Enter Origntr_id (Upper case for NetView) :");
    result=new_gets(buff);
    printf("\n");
    strncpy((char *)block.origntr_id,buff ,8);

    /*****
    /* Now call the verb */
    *****/

    NMAPI_C(&block);

    /*****
    /* Now check the return codes */
    *****/
    printf("NMAPI_C call returned : ");

    switch (block.retcode) {
        case NM_OK: printf("NM_OK \n");
                    break;

        case NM_ALREADY_OPEN: printf("NM_ALREADY_OPEN\n");
                               printf("An open has already been issued\n");
                               break;

        case NM_OPEN_TIMEOUT: printf("NM_OPEN_TIMEOUT\n");
                               printf("Open has timed out\n");
                               break;

        case NM_INVALID_OPTIONS: printf("NM_INVALID_OPTIONS\n");
                                  printf("Invalid options\n");
                                  break;

        case NM_OPEN_TOO_MANY_APPS : printf("NM_OPEN_TOO_MANY_APPS\n");
                                      printf("No more PU-CNMS sessions are
available\n");
                                      break;

        case NM_OPEN_FAILED: printf("NM_OPEN_FAILED\n");
                              printf("PU-CNMS Open Response error\n");
                              break;

        case NM_NODE_UNAVAILABLE: printf("NM_NODE_UNAVAILABLE\n");
                                   printf("The SNA node has failed\n");
                                   break;

        case NM_COMM_SUBSYSTEM_NOT_LOADED: printf("NM_COMM_SUBSYSTEM_NOT_LOADED\n");
                                               printf("The Cleo subsystem has stopped\n");
```

```

        break;

    case NM_INVALID_VERB: printf("NM_INVALID_VERB\n");
                        printf("Invalid verb\n");
                        break;

    default:            printf("Eh? Unknown return code = %d \n",block.retcode);
                        break;
}
sessstat(block);

} /* end of testopen */

/**PROC+*****
/* Name:          testclose
/*
/* Purpose:      To test the NM_CLOSE verb.
/*
/* Returns:      None.
/*
/* Params:       None.
/*
/* Operation:
/*
/**PROC-*****
void testclose()

{
    NM_VCB block;
    int i,j;

    /*******
    /* Now initialise vcb */
    /*******
    block.opcode=NM_CLOSE;
    block.options=0x00;
    block.retcode=0x00;
    block.dptr=0x00;
    block.datalength=0x00;
    block.sess_status=0x00;

    /*******
    /* Now call the verb */
    /*******

    NMAPI_C(&block);

    /*******
    /* Now check the return codes */
    /*******
    printf("NMAPI_C call returned : ");

    switch (block.retcode) {

        case NM_OK: printf("NM_OK \n");
                    break;

        case NM_NOT_OPENED: printf("NM_NOT_OPENED\n");
                            printf("No NM_OPEN issued\n");
                            break;

        case NM_INVALID_VERB: printf("NM_INVALID_VERB\n");
                              printf("Invalid verb\n");
                              break;

        default:            printf("Eh? Unknown return code = %d \n",block.retcode);
                            break;
    }
}

```

Network Management API

```
    sessstat(block);
} /* end of testclose */

/**PROC+*****
/* Name:      testreceive
/*
/* Purpose:   To test the NM_RECEIVE verb.
/*
/* Returns:   None.
/*
/* Params:    None.
/*
/* Operation:
/*
/*
/*PROC-*****
void testreceive()

{
    NM_VCB block;
    char *result;
    char buff[10];
    int ii;

    /*******
    /* Now initialise vcb */
    /*******
    block.opcode=NM_RECEIVE;
    printf("Options? \n");
    printf("      NM_RCV_WAIT_MSG      = %d\n",NM_RCV_WAIT_MSG);
    printf("      NM_RCV_WAIT_IGNORE_STATUS = %d\n",NM_RCV_WAIT_IGNORE_STATUS);
    printf("      NM_NONE                     = %d\n",NM_NONE);
    printf("Enter option required : ");
    result=new_gets(buff);
    printf("\n");
    block.options=atoi(result);
    block.retcode=0x00;
    block.dptr=databuff;
    printf("Size of receive buffer? ");
    result=new_gets(buff);
    printf("\n");
    block.datalength=atoi(result);
    block.sess_status=0x00;
    strcpy((char *)block.origntr_id,"");

    /*******
    /* Now call the verb */
    /*******

    NMAPI_C(&block);

    /*******
    /* Now check the return codes */
    /*******
    printf("NMAPI_C call returned : ");

    switch (block.retcode) {

        case NM_OK: printf("NM_OK \n");
                    break;

        case NM_OK_DATA_TRUNCATED: printf("NM_OK_DATA_TRUNCATED\n");
                                    printf("OK, but data was longer than the
receive buffer\n");
                                    break;

        case NM_ACK_RECEIVED: printf("NM_ACK_RECEIVED\n");
                                printf("OK; ACK received\n");
                                break;

        case NM_NACK_RECEIVED: printf("NM_NACK_RECEIVED\n");
```



```

        printf("OK; NACK received\n");
        strncpy(buff, (char *)block.dptr, 4);
        printf("Sense data is %s\n", buff);
        break;

    case NM_NOT_OPENED: printf("NM_NOT_OPENED\n");
        printf("No NM_OPEN issued\n");
        break;

    case NM_INVALID_OPTIONS: printf("NM_INVALID_OPTIONS\n");
        printf("Invalid options\n");
        break;

    case NM_RCV_NO_MESSAGE: printf("NM_RCV_NO_MESSAGE\n");
        printf("No message received\n");
        break;

    case NM_NODE_UNAVAILABLE: printf("NM_NODE_NOT_AVAILABLE\n");
        printf("The SNA node has failed\n");
        break;

    case NM_COMM_SUBSYSTEM_NOT_LOADED: printf("NM_COMM_SUBSYSTEM_NOT_LOADED\n");
        printf("The Cleo subsystem has stopped\n");
        break;

    case NM_INVALID_VERB: printf("NM_INVALID_VERB\n");
        printf("Invalid verb\n");
        break;

    default:      printf("Eh? Unknown return code = %d \n", block.retcode);
        break;
}
printf("\n");
printf("returned data size is %d\n\n", block.datalength);
printf("returned data is :\n");
for (ii = 0 ; ii < (int)block.datalength; ii++ )
{
    printf( "%2.2x ", block.dptr[ii] );
    if ( (ii%16) == 15 ) printf("\n");
}

sessstat(block);

printf("message PRID is %d", block.dptr[6]);
printf(". Use this when sending a solicited response \n");

} /* end of testreceive */

/**PROC+*****
/* Name:      testsend
/*
/* Purpose:   To test the NM_SEND verb.
/*
/* Returns:   None.
/*
/* Params:    None.
/*
/* Operation:
/*
/**PROC-*****
void testsend()

{
    NM_VCB block;
    char *result;
    char buff[10];
    int i, len;

    /*******
    /* Now initialise vcb */
    /*******

```

Network Management API

```
block.opcode=NM_SEND;
block.retcode=0x00;
block.sess_status=0x00;
printf("Options? \n");
printf("      NM_SEND_ACK = %d\n",NM_SEND_ACK);
printf("      NM_NONE      = %d\n",NM_NONE);
printf("Enter option required : ");
result=new_gets(buff);
printf("\n");
block.options=atoi(result);
/*****
/* Select data type */
*****/
printf("What kind of data do you want to send?\n\n");
printf("1. NMVT alert data\n");
printf("2. NMVT response data\n");
printf("3. NMVT unsolicited response data\n");
printf("4. NMVT unsolicited response data >256 bytes\n");
printf("Enter number: ");
result = new_gets(buff);
i= atoi(result);
switch(i)
{
    case 1: memcpy(databuff,(char *)NMVTalert,156);
             printf("You have selected NMVT alert.\n");
             len=156;
             break;

    case 2: memcpy(databuff,(char *)NMVTresp,63);
             printf("Enter message PRID : ");
             result = new_gets(buff);
             i= atoi(result);
             printf("\n");
             databuff[6] = i;

             len=63;
             break;

    case 3: memcpy(databuff,(char *)NMVTusresp,50);
             len=50;
             break;

    case 4: memcpy(databuff,(char *)NMVTlong,348);
             len=348;
             break;

    default:
             len = 0;
             break;
}

block.dptr=databuff;
block.datalength=len;

/*****/
/* Now call the verb */
/*****/

NMAPI_C(&block);

/*****/
/* Now check the return codes */
/*****/
printf("NMAPI_C call returned : ");

switch (block.retcode) {

    case NM_OK: printf("NM_OK \n");
                break;
```

```

        case NM_NOT_OPENED: printf("NM_NOT_OPENED\n");
                           printf("No open issued\n");
                           break;

        case NM_INVALID_OPTIONS: printf("NM_INVALID_OPTIONS\n");
                                printf("Invalid options\n");
                                break;

        case NM_SEND_SESSION_INACTIVE: printf("NM_SEND_SESSION_INACTIVE\n");
                                printf("The PU-CNMS session is not
active\n");
                                break;

        case NM_SEND_AWAITING_RESPONSE: printf("NM_SEND_AWAITING_RESPONSE\n");
                                printf("A SEND may not be issued until
aresponse to the previous send is received\n");
                                break;

        case NM_DATA_SIZE_ERROR: printf("NM_DATA_SIZE_ERROR\n");
                                printf("Data size exceeds 512 bytes\n");
                                break;

        case NM_NODE_UNAVAILABLE: printf("NM_NODE_UNAVAILABLE\n");
                                printf("The SNA node is unavailable\n");
                                break;

        case NM_COMM_SUBSYSTEM_NOT_LOADED: printf("NM_COMM_SUBSYSTEM_NOT_LOADED\n");
                                printf("The Cleo subsystem has stopped\n");
                                break;

        case NM_INVALID_VERB: printf("NM_INVALID_VERB\n");
                             printf("Invalid verb\n");
                             break;

        default:      printf("Eh? Unknown return code = %d \n",block.retcode);
                     break;
    }

    sessstat(block);
} /* end of testsend */

/**PROC+*****
/* Name:      tstbadvb
/*
/* Purpose:   To test a bad verb number.
/*
/* Returns:   None.
/*
/* Params:    None.
/*
/* Operation:
/*
/**PROC-*****

void tstbadvb()
{
    NM_VCB block;

    /**
    /* Now initialise vcb */
    /**
    block.opcode=6;
    block.retcode=0x00;

    printf("Testing invalid verb\n");
    /**
    /* Now call the verb */
    /**

```

Network Management API

```
NMAPI_C(&block);

/*****
/* Now check the return codes */
*****/
printf("NMAPI_C call returned : ");

switch (block.retcode) {

    case NM_OK: printf("NM_OK \n");
                break;

    case NM_NOT_OPENED: printf("NM_NOT_OPENED\n");
                        break;

    case NM_INVALID_OPTIONS: printf("NM_INVALID_OPTIONS\n");
                             printf("Invalid options\n");
                             break;

    case NM_INVALID_VERB: printf("NM_INVALID_VERB\n");
                           printf("Invalid verb\n");
                           break;

    case NM_NODE_UNAVAILABLE: printf("NM_NODE_UNAVAILABLE\n");
                              printf("The SNA node has failed\n");
                              break;

    case NM_COMM_SUBSYSTEM_NOT_LOADED: printf("NM_COMM_SUBSYSTEM_NOT_LOADED\n");
                                         printf("Comm subsystem not loaded\n");
                                         break;

    default:      printf("Eh? Unknown return code = %d \n",block.retcode);
                  break;

}

sessstat(block);

} /* end of testbadvb */

/****PROC+*****/
/* Name:      testerr */
/*
/* Purpose:   To test the other error cases.
/*
/* Returns:   None.
/*
/* Params:    None.
/*
/* Operation:
/*
/*
/****PROC-*****/

void testerr()
{
    NM_VCB block;

    printf("Testing some error conditions. \n");
    block.retcode = 0;
    block.opcode = NM_OPEN;
    NMAPI_C (&block);
    if (block.retcode != 0)
    {
        printf("Done NM_OPEN\n");
        block.retcode = 0;
    }
    block.opcode = NM_CLOSE;
    NMAPI_C (&block);
    if (block.retcode != 0)
    {
        printf("Done NM_CLOSE\n");
    }
}
```

```

        block.retcode = 0;
    }
    block.opcode = NM_SEND;
    NMAPI_C (&block);
    if (block.retcode != 0)
    {
        printf("Done NM_SEND\n");
        block.retcode = 0;
    }
    block.opcode = NM_RECEIVE;
    NMAPI_C (&block);
    if (block.retcode != 0)
    {
        printf("Done NM_RECEIVE\n");
        block.retcode = 0;
    }
}
/* end of testerr */

```

```

/**PROC+***/
/* Name:      new_gets                                     */
/*          */
/* Purpose:   To get string                               */
/*          */
/* Returns:   pointer to string                           */
/*          */
/**PROC-***/
char *new_gets( s )
char *s;
{
    while (!gets( s ));
    return( s );
}

```

```

/**PROC+***/
/* Name:      sessstat                                     */
/*          */
/* Purpose:   To get verb and switch to appropriate routine. */
/*          */
/* Returns:   None.                                         */
/*          */
/* Params:    block                                        */
/*          */
/* Operation: Outputs session status                       */
/*          */
/**PROC-***/
void sessstat( block )
NM_VCB block;
{
    printf("\nSession status is NM_SESSION_");
    switch(block.sess_status)
    {
        case NM_SESSION_ACTIVE:
            printf("ACTIVE\n");
            printf("- PU-CNMS session is active\n");
            break;

        case NM_SESSION_PU_INACTIVE:
            printf("PU_INACTIVE\n");
            printf("- PU-CNMS session is inactive\n");
            break;

        case NM_SESSION_LINK_ERROR :
            printf("LINK_ERROR\n");
            printf("- PU-CNMS session is inactive\n");
            break;

        default:

```

Network Management API

```
        printf("????\n-   Unknown session status = %d \n",block.sess_status);
        break;
    }
    return;
}

int sig_catcher (sig)
int sig;
{
    if ( (oldsig != SIG_DFL) && (oldsig != SIG_IGN) &&(oldsig !=SIG_HOLD) )
    {
        (*oldsig)();
    }
}

void sigcatch()
{
    if (signalset)
    {
        printf("Signal catcher already added\n");
        return;
    }
    oldsig = SIGNAL_SET (SIGPOLL, sig_catcher);
    signalset = TRUE;
    printf("Signal catcher started\n");
    return;
}

void sigstop()
{
    if (!signalset)
    {
        printf("No signal catcher was added\n");
        return;
    }
    signalset = FALSE;
    SIGNAL_SET (SIGPOLL, oldsig);
    printf("Signal catcher stopped\n");
    return;
}

/**PROC+*****/
/* Name:      main                                     */
/* Purpose:   To get verb and switch to appropriate routine. */
/* Returns:   None.                                     */
/* Params:    None.                                     */
/* Operation: Offers user a menu and switches on input.  */
/* ****PROC-*****/

void main()
{
    char buff[10];
    int i;
    char *result;
    printf("TESTING THE NM-API LIBRARY...\n\n");
    do
    {
        printf("Which verb? \n");
        printf(" 0 - Exit test program\n");
        printf(" 1 - NM_OPEN\n");
        printf(" 2 - NM_RECEIVE\n");
        printf(" 3 - NM_CLOSE\n");
        printf(" 4 - NM_SEND\n");
        printf(" 5 - test bad verb number\n");
    }
```

```
#if 0
    printf(" 6 - test other error cases\n");
    printf(" 7 - Add a signal catcher\n");
    printf(" 8 - Remove signal catcher\n");
#endif
    printf("Enter option:  ");
    result = new_gets(buff);
    printf("\n");
    i= atoi(result);
    switch(i)
    {
        case 0: break;
        case 1: testopen();
                break;
        case 2: testreceive();
                break;
        case 3: testclose();
                break;
        case 4: testsend();
                break;
        case 5: tstbadvb();
                break;
    }
    #if 0
        case 6: testerr();
                break;
        case 7: sigcatch();
                break;
        case 8: sigstop();
                break;
    #endif
    default: printf("Invalid option\n");
            break;
    }
    if (i)
    {
        printf("\n\n");
    }
} while (i >= 1);

printf("\nExit from NM-API test program\n");

} /* end of main */
```

Compiling *nmapitest.c*

To compile an dlink the application, issue the following command:

```
cc -I /usr/include/cleo -o nmpitest nmpitest.c -lcom
```

The -I option indicates the include path, */usr/include/cleo*, for the required header files *svgsig.h*, *svigstr.h*, *btypes.h* and *nmap.c.h*

The -o option gives the name of the output executable file, *nmapitest*.

The source file is *nmapitest.c*.

The -l option indicates the name of the library to be used, */usr/lib/libcom.a*.

Network Management API

Compiling Under AIX

If you're compiling the application under the AIX operating system, issue a command based on the following example:

```
cc -I /usr/include/cleo -o nmapitest nmapitest.c -lcom \  
> -bimport: /lib/pse.exp [-DSXSYS_AIX | -DSXSYS_AIX3xlx51
```

The `-b` option imports external streams symbols that are required for successful compilation.

Also, if you're compiling under AIX 3.2.x, use the `-D` option:

`-DSXSYS_AIX`

or under AIX 3.1.5 use:

`-DSXSYS_AIX3xlx5`

Listing of nmap_i_c.h

The header file *nmap_i_c.h*, listed below, contains the following:

- ◆ The NM-API function prototype
- ◆ The structure declaration for the NM-API verb control block (VCB)
- ◆ The #define statements that substitute meaningful symbolic constants for hexadecimal values supplied to and returned by NM-API verbs.

```

/* Version 100.1 */
/* nmap_i_c.h
/* Copyright (c) 1989 - 1995 Interface Systems, Inc.

/*****
/*      Function prototype for the NM-API
/*****

extern void NMAPI_C();

/*****
/*      Verb Control Block definition
/*****

/****STRUCT****
/* Structure: NM_VCB
/*
/* Description: Definition of Verb Control Block for NMAPI library routines.
/****STRUCT-****
typedef struct _nm_api
{
    unsigned short opcode;          /* Verb identifying operation code.
    unsigned char  options;         /* Options flags
    unsigned char  retcode;         /* Return code from verb
    unsigned char *dptr;           /* Pointer to i/o buffer
    unsigned short datalength;      /* Length of i/o data
    unsigned char  sess_status;     /* Active/inactive session flag
    unsigned char  origntnr_id[8]; /* Application name (ASCII)
} NM_VCB;
/****STRUCT-****

/*****
/*      NM-API command verbs - set in opcode.
/*****
#define NM_OPEN          10
#define NM_CLOSE         20
#define NM_RECEIVE       30
#define NM_SEND           40

/*****
/*      Parameters for NM-API verbs - set in options.      For use with...
/*****
#define NM_NONE          0          /* all verbs
#define NM_FILTER        1          /* NM_OPEN
#define NM_RCV_WAIT_MSG  2          /* NM_RECEIVE

```

Network Management API

```
#define NM_RCV_IGNORE_STATUS      4      /* NM_RECEIVE */
#define NM_RCV_WAIT_IGNORE_STATUS 6      /* NM_RECEIVE */
#define NM_SEND_ACK               8      /* NM_SEND */

/*****
/* Return codes from NM-API verbs, returned in retcode.
*****/
#define NM_OK                      0
#define NM_OK_DATA_TRUNCATED      1

#define NM_COMM_SUBSYSTEM_NOT_LOADED 99
#define NM_NODE_UNAVAILABLE        98

#define NM_INVALID_VERB           4

#define NM_OPEN_TIMEOUT           7
#define NM_OPEN_TOO_MANY_APPS    8
#define NM_OPEN_FAILED            9

#define NM_ALREADY_OPEN           10
#define NM_NOT_OPENED             11

#define NM_INVALID_OPTIONS        12
#define NM_DATA_SIZE_ERROR        13
#define NM_RCV_NO_MESSAGE         14

#define NM_ACK_RECEIVED            20
#define NM_NACK_RECEIVED          21

#define NM_SEND_SESSION_INACTIVE  25
#define NM_SEND_AWAITING_RESPONSE 26

/*****
/* Session status settings - returned in sess_status.
*****/
#define NM_SESSION_PU_INACTIVE    0
#define NM_SESSION_ACTIVE         1
#define NM_SESSION_LINK_ERROR     2
```

Index

A

Accessing NetView, 1-3
Activate-connection (command), 2-17
Activate-session (command), 2-19

C

commanage, 1-3
Commands (NetView)
 activate-connection, 2-17
 activate-session, 2-19
 deactivate-connection, 2-18
 deactivate-oldlu, 2-21
 deactivate-session, 2-20
 display-applicats, 2-4
 display-connection, 2-12
 display-link, 2-11
 display-lu, 2-7
 display-mode, 2-9
 display-oldlu, 2-14
 display-pool, 2-16
 display-rlu, 2-8
 display-session, 2-13
 display-tp, 2-10
 login-sp, 1-4, 2-3, 2-22

D

Deactivate-connection (command), 2-18
Deactivate-oldlu (command), 2-21
Deactivate-session (command), 2-20
Display-applicats (command), 2-4
Display-connection (command), 2-12
Display-link (command), 2-11
Display-lu (command), 2-7
Display-mode (command), 2-9

Display-oldlu (command), 2-14
Display-pool (command), 2-16
Display-rlu (command), 2-8
Display-session (command), 2-13
Display-tp (command), 2-10

L

Login-sp (command), 1-4, 2-3, 2-22

N

NETMAN, 1-3
NetView command abbreviations, 2-5
NetView command categories, 1-6
 network access commands, 1-6, 2-2
 node commands, 1-6, 2-5
 UNIX commands, 1-6, 2-22
NetView commands, 2-1
 activate-connection, 2-17
 activate-session, 2-19
 deactivate-connection, 2-18
 deactivate-oldlu, 2-21
 deactivate-session, 2-20
 display-applicats, 2-4
 display-connection, 2-12
 display-link, 2-11
 display-lu, 2-7
 display-mode, 2-9
 display-oldlu, 2-14
 display-pool, 2-16
 display-rlu, 2-8
 display-session, 2-13
 display-tp, 2-10
 login-sp, 2-3, 2-22
NetView operator, 1-2, 1-4, 2-3, 2-22
NetView screen, 1-5
 input area, 1-5

- changing size, 1-5
- Network access commands, 2-2
- Network addressable units (NAUs), 1-2
- Network management API
 - see NM-API
- Network management connection, 1-2, 3-2
- Network management vector transport (NMVT) messages, 3-2, 3-5, 3-6, 3-8, 3-11, 3-14, 3-15, 3-17
- Network resources, 1-2
 - logical, 1-2
 - physical, 1-2
- NM-API, 3-1
 - assigning values to VCB variables, 3-4
 - clearing VCB variables, 3-4
 - entry point, 3-3
 - input parameters, 3-3
 - message filtering, 3-5
 - return codes, 3-3, 3-4
 - verb, 3-3
 - issuing call, 3-4
 - verb control block (VCB), 3-3
 - verbs
 - invoking, 3-4
- NM-API applications
 - compiling and linking, 3-29
 - designing and writing, 3-5
 - initializing, 3-8
 - multiple, 3-6
- NM-API verbs, 3-2
 - NM_CLOSE, 3-2, 3-7
 - NM_OPEN, 3-2, 3-6, 3-8
 - NM_RECEIVE, 3-3, 3-5, 3-11
 - NM_SEND, 3-3, 3-5, 3-11, 3-15
- Node commands, 2-5

S

- Signal catcher, 3-6
- sigpoll, 3-6
- startsrv*, 1-3

U

- UCF
 - see UNIX command facility
- UNIX
 - security features, 1-7, 2-22
 - shell, 1-7, 2-22
- UNIX command facility (UCF), 1-6, 1-7, 2-22
 - background processing, 2-25
 - canceling a command, 2-24, 2-25
 - daemon, 1-6, 1-7, 2-22
 - output, 2-24
 - security, 2-25
 - shell, 1-7
 - uppercase characters, 2-23
 - user, 1-7, 2-22
 - valid commands, 2-24

V

- Verb control block (for NM-API), 3-3, 3-31

LINKix

Management Utilities

June 1994

RESTRICTED RIGHTS

Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (C)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013. Cleo Communications.

LIMITED RIGHTS

Limited rights shall be effective indefinitely and are not subject to expiration as set forth in paragraph (3) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013.

Interface Systems, Inc. 5855 Interface Drive, Ann Arbor, MI 48103



June 10,1994

Management Utilities

June 1994

Cleo Communications reserves the right to, without notice, modify or revise all or part of this document and/or change product features or specifications and shall not be responsible for any loss, cost or damage, including consequential damage, caused by reliance on these materials.

This document may not be reproduced, stored in a retrieval system or transmitted, in whole or in part in any form or by any means (electronic, mechanical, photocopied or otherwise) without the prior written permission of Cleo Communications.

© 1994 Cleo Communications – All rights reserved.

Portions:

© 1989–1993 Data Connection Limited

© 1989, 1990 Digital Communications Associates, Inc.

© 1989, 1990 Microsoft Corporation

Table of Contents

1. Introduction	
About This Manual	1-2
Documentation Conventions	1-3
Related Publications	1-4
Management Overview	1-7
Diagnostics Support	1-9
 2. Using Manage	
Management Tasks	2-2
Starting Manage	2-3
Starting and Stopping Services	2-5
Starting a Local Node	2-8
Starting a Local Node Without Tracing	2-8
Starting a Local Node With Tracing	2-9
Starting a Link	2-12
Starting a Link Without Tracing	2-12
Starting a Link With Tracing	2-13
SDLC or X.25 links	2-14
Vendor X.25 links	2-16
Token Ring links	2-17
Starting a Queued TP	2-20
Starting a Queued TP Without Tracing	2-20
Starting a Queued TP With Tracing	2-21
Starting the Network Management Daemon	2-24
Starting the Network Management Daemon Without Tracing	2-24
Starting the Network Management Daemon With Tracing	2-26
Terminating Services	2-28

Table of Contents

Managing an Active Link	2-30
SDLC, X.25 or Vendor X.25 links	2-30
Token Ring Links	2-32
Managing an Active Queued TP	2-35
Managing Node Resources	2-37
Selecting a Node Resource	2-37
Managing 3270 LUs	2-39
Status of 3270 LUs	2-39
Status of a Selected 3270 LU	2-41
Managing APPC LUs	2-44
Options for APPC LUs	2-44
Options for Individual APPC LUs	2-46
Activating Remote APPC LU Sessions	2-48
Activating APPC Sessions When No Active Sessions Exist	2-48
Activating APPC Sessions When Active Sessions Exist	2-50
Deactivating Sessions With a Partner LU on a Specific Mode	2-51
Viewing Sessions With a Partner LU	2-53
Managing LUA LUs	2-54
Status of LUA LUs	2-54
Status of a Selected LUA LU	2-56
Managing RJE LUs	2-58
Status of RJE LUs	2-58
Status of a Selected RJE LU	2-60
Managing Connections	2-62
Viewing and Changing Audit Log and Trace Settings for a Local Node	2-65
Managing Log and Trace Settings for the Network Management Daemon	2-68
Managing 3270 Users	2-71
Forcing Off a 3270 User	2-72
Managing APPC Applications	2-74
Forcing Off an APPC Application	2-75
Managing LUA Applications	2-77
Forcing Off an LUA Application	2-78

Table of Contents

Managing RJE Workstations	2-80
Forcing Off an RJE Workstation	2-81
3. Using Browse	
Starting the Program	3-2
Viewing Audit and Error Log Files	3-5
Opening a Log File	3-5
Displaying a Log File	3-6
Customizing a Log File	3-7
Selecting Log File Display Options	3-8
Selecting Log File Filter Options	3-10
Viewing Trace Files	3-13
Opening a Trace File	3-13
Displaying a Trace File	3-15
Customizing a Trace File	3-16
Selecting Trace File Display Options	3-17
Selecting Trace File Filter Options	3-18
Resetting the Audit and Error Log Files	3-20
Printing Diagnostic Reports	3-22
Exiting the Browse Program	3-24
4. Additional Utilities	
Managing Connections With comconn	4-2
Viewing the Status of a Connection	4-2
Activating a Connection	4-3
Deactivating a Connection	4-4
Viewing Status of LUs	4-5
Output Format	4-5

Table of Contents

1. Introduction

The *Management Utilities User's Guide* describes the software and techniques that you can use to manage the LINKix software components in an IBM Systems Network Architecture (SNA) network. Before reading this manual, you should be familiar with the general concepts of configuring and using LINKix.

You should also be familiar with IBM's SNA network concepts and terminology. If you do not understand these products, see the **Related Publications** section at the end of this chapter for a list of manuals that provide more information.

About This Manual

This manual contains the following sections:

Chapter 1, **Introduction**, outlines the concepts you must know before you can begin managing the system and lists some of the required tasks.

Chapter 2, **Using Manage**, explains how to use the Manage program to monitor and control the operation of various LINKix components.

Chapter 3, **Using Browse**, explains how to use the Browse program to view audit and error log files, review trace files, copy and reset diagnostic files and print reports from diagnostic files.

Documentation Conventions

The following documentation conventions are used in this manual:

italics Italicized words are used for emphasis, to identify the titles of manuals and to indicate file and directory names.

bold Bold type is used for cross-references to other chapters within the manual or to other sections within a chapter.

KEYNAME A key name in this format, for example **ENTER**, refers to a specific key on your terminal's keyboard.

Some keyboard commands require that you press one key and hold it down while you press one or more other keys. These are also called combination keystrokes. To indicate this, key names are shown together with a + sign between them (example: **CTRL + D**). While holding down the first key, press the other key(s).

KEYNAME A key name in uppercase letters, for example **ENTER**, refers to a 3270 key. Appendix A lists the default UNIX terminal keys used to represent these 3270 keys. If you remap your keyboard, you can always find the UNIX terminal keystroke used to represent a particular 3270 key by using the View and Define Key Definitions dialog.

0xnn A number preceded by 0x (for example 0x20) indicates a hexadecimal value.

Related Publications

The documentation that accompanies this product does not attempt to fully describe IBM's Systems Network Architecture (SNA), any of the 3270 information display system products, APPC, LUA, RJE or the UNIX operating system. You may, therefore, find one or more of the following documents useful.

SNA

For more information on SNA, see the following publications:

GC30-3073	<i>IBM Systems Network Architecture: Technical Overview</i>
GC30-3072	<i>IBM Systems Network Architecture: Concepts and Products</i>
GA27-3136	<i>IBM Systems Network Architecture: Formats</i>
SC30-3112	<i>IBM SNA Format and Protocol Reference Manual: Architectural Logic</i>
GC20-1869	<i>IBM Systems Network Architecture: Introduction to Sessions between Logical Units</i>
GG24-1557	<i>IBM Advanced Communications Function Products Installation Guide</i>

Common User Access

For information on Common User Access (CUA), refer to:

SC26-4582	<i>IBM Systems Application Architecture Common User Access: Advanced Interface Design Guide</i>

3270 Display Systems

For more information on 3270 information display system products, see the following publications:

GA23-0061	<i>IBM 3270 Information Display System: 3274 Control Unit Description and Programmer's Guide</i>

GA33-3056	<i>IBM 3270 Information Display System: Color and Programmed Symbols</i>
GA23-0023	<i>IBM 3270 Information Display System: 3274 Control Unit Display Station: Operator's Guide</i>
GA23-0218	<i>IBM 3270 Information Display System: 3174 Control Unit Functional Description</i>

APPC/LU 6.2

For information about APPC (LU 6.2), refer to these documents:

GG24-1584	<i>IBM Systems Network Architecture: Introduction to APPC</i>
GC30-3084	<i>IBM Systems Network Architecture: Transaction Programmer's Reference Manual for LU Type 6.2</i>
SC30-3269	<i>IBM Systems Network Architecture: Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2</i>
SC30-6808	<i>IBM Systems Network Architecture LU6.2 Reference: Peer Protocols</i>
SC30-3400	<i>IBM Systems Network Architecture: VTAM Programming for LU Type 6.2</i>

You may also want to refer to the list of general SNA documents

LUA

The following SNA documentation provides important information that can be of use when programming with the LUA API:

GA27-3136	<i>IBM Systems Network Architecture: Formats</i>
SC30-3112	<i>IBM Systems Network Architecture Format and Protocol Reference Manual: Architectural Logic</i>
GC20-1869	<i>IBM Systems Network Architecture: Introduction to Sessions between Logical Units</i>

Introduction

RJE

For information about RJE, refer to these documents:

SC23-0046 *System Programming Library: JES2 Initialization and Tuning*

SC23-0064 *MVS/Extended Architecture Operations: JES2 Commands*

GC28-1352 *MVS/Extended Architecture JCL Reference*

UNIX

For information on the UNIX operating system and its utilities, refer to the user documentation provided with your UNIX system.

Management Overview

As the administrator, you are responsible for configuring the LINKix software and managing the operation of the system. To perform these tasks, we provide you with a number of configuration and management programs.

With these utilities, you can perform the following operations:

1. Start and stop the LINKix software as needed.
2. Start and stop LINKix services (local nodes, links and the network management daemon program) as needed.
3. Manage LINKix resources (such as connections and 3270 LUs) as needed while communicating with remote computers.
4. Diagnose and troubleshoot problems that arise during LINKix operation.

After you configure LINKix, you can manage its resources by using the Manage program, which allows you to perform the following functions:

- ◆ Start and stop services
- ◆ Activate and deactivate connections
- ◆ Reset 3270 LUs, LUA LUs, and RJE LUs
- ◆ Activate and deactivate APPC LUs
- ◆ Stop 3270 users, APPC, CPI-C or LUA applications, or RJE workstations
- ◆ Specify how diagnostic information will be kept
- ◆ View system resource usage
- ◆ If you are using a connection configured to be started “By operator,” you need to start the connection before any system resource that uses the connection can be used.

Introduction

- ◆ Manage node resources as needed for your configuration. This includes the following tasks:
 - ◆ Monitor the session status of 3270 LUs, APPC LUs, LUA LUs, and RJE LUs, and reset (deactivate) LUs if needed
 - ◆ Activate, monitor, and deactivate connections as needed
 - ◆ Initialize or reset APPC parallel session limits as needed
 - ◆ Review audit and error log settings, and change as needed
- ◆ Monitor 3270 users, APPC, CPI-C, and LUA applications, and RJE workstations, and force them off if necessary.
- ◆ Troubleshoot any problems that arise during operation by monitoring the LINKix error and audit log files.

Diagnostics Support

LINKix also provides a number of diagnostics tools to help correct problems encountered during operation.

- ◆ The Manage program can be used to view the status of various LINKix components.
- ◆ Any component detecting an error writes an entry to an error log file, named in the LINKix configuration.
- ◆ All significant system events can be recorded in an audit log file, specified in the LINKix configuration. The level of detail recorded in the file is controlled by both the LINKix configuration settings and the Manage program.
- ◆ Many components can produce trace files which record their activity. Because tracing degrades system performance, it is normally disabled.
- ◆ The Browse program can be used to view log and trace files, and to select specific information from them for display or printing.
- ◆ LINKix can generate alerts to be sent to the NetView program on a designated host connection, to provide information on link and connection problems. Alerts can also be generated by the CSV TRANSFER_MS_DATA verb (see the *CSV Programmer's Guide* for more information). You can also configure alerts to be sent by 3270 emulation program users (see the *Administration Guide*) — note that these are sent on the user's current session to the host for that session, not on the designated host connection.

Introduction

2. Using Manage

This chapter discusses the Manage program, which allows you to monitor and control LINKix operation. It explains how you can use the program's menus and dialogs to monitor services (local nodes, links, queued TPs and the network management daemon), connections, LUs, 3270 users, APPC, CPI-C and LUA applications, and RJE workstations. This chapter contains the following information:

- ◆ How to use the Manage program
- ◆ How to start and stop services
- ◆ How to manage services
- ◆ How to manage node resources: 3270 LUs, APPC LUs, LUA LUs, RJE LUs and connections
- ◆ How to manage 3270 users
- ◆ How to manage APPC TPs and CPI-C applications
- ◆ How to manage LUA applications
- ◆ How to manage RJE workstations

Management Tasks

Managing services involves monitoring local nodes, links, LUs, 3270 users, APPC, CPI-C and LUA applications, RJE workstations and connections. Management also involves monitoring audit logs, error logs and trace files for any type of service or for the network management daemon program. You can use the Manage program to do the following:

- ◆ Start or stop local nodes, links, queued TPs and the network management daemon
- ◆ View detailed information for services, LUs, connections, 3270 users, APPC, CPI-C and LUA applications, and RJE workstations
- ◆ Refresh screen information
- ◆ Adjust audit logging levels
- ◆ Control tracing for local nodes, links, queued TPs and the network management daemon program
- ◆ Force off 3270 users, APPC TPs, CPI-C applications, LUA applications or RJE workstations
- ◆ Terminate connections, 3270 LUs, APPC LUs, LUA LUs or RJE LUs

Starting Manage

Before you start the Manage program, the LINKix control daemon must be started — refer to chapter 11 of the *Administration Guide* if necessary. The Manage program obtains the necessary configuration information from the running configuration file on the master server.

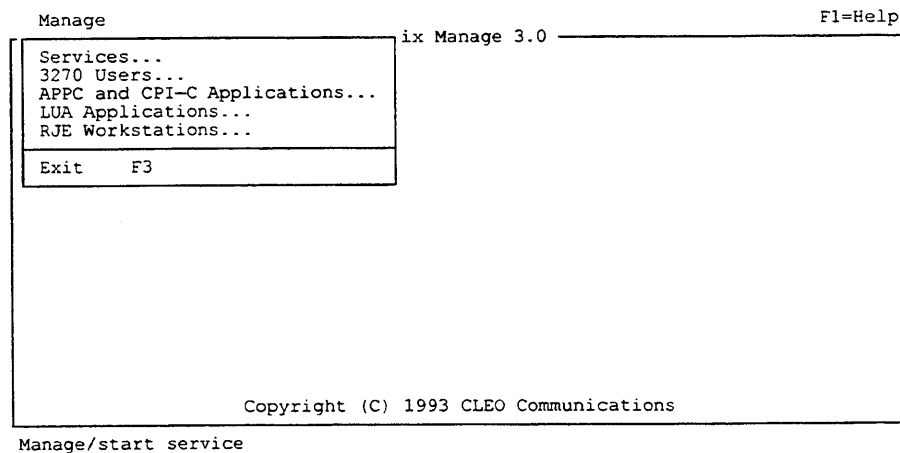
To start the Manage program, enter the following command at the UNIX command prompt:

```
commanage
```

Alternatively, to start the Motif version of the program, enter the following command:

```
xcommanage
```

The Manage program main screen is shown below.



Using Manage

When you first start the Manage program, the Manage menu is automatically selected. It is also automatically selected when you end a series of dialogs and return to the main screen. This menu allows you to:

- ◆ Start and stop services
- ◆ Manage links
- ◆ Manage queued TPs
- ◆ Manage the network management daemon
- ◆ Manage node resources: 3270 LUs, APPC LUs, LUA LUs, RJE LUs, connections, and node audit log and trace files
- ◆ Manage 3270 users
- ◆ Manage APPC TPs and CPI-C applications
- ◆ Manage LUA applications
- ◆ Manage RJE workstations

Note that there are different options for managing APPC or CPI-C invocable TPs, depending on their configuration:

- ◆ Invocable TPs configured in the LINKix configuration file as “queued operator-started” are listed in the Start/Manage Services dialog, and can be started or stopped from this dialog. While they are active, you can zoom on them to view audit log and trace parameters.
- ◆ Invocable TPs configured in the LINKix configuration file as “queued auto-started” are listed in the StartManage Services dialog only if they are active. They can be stopped from this dialog. While they are active, you can zoom on them to view audit log and trace parameters.
- ◆ Invocable TPs configured as non-queued, and all invoking TPs, are not listed in the Start/Manage Services dialog.

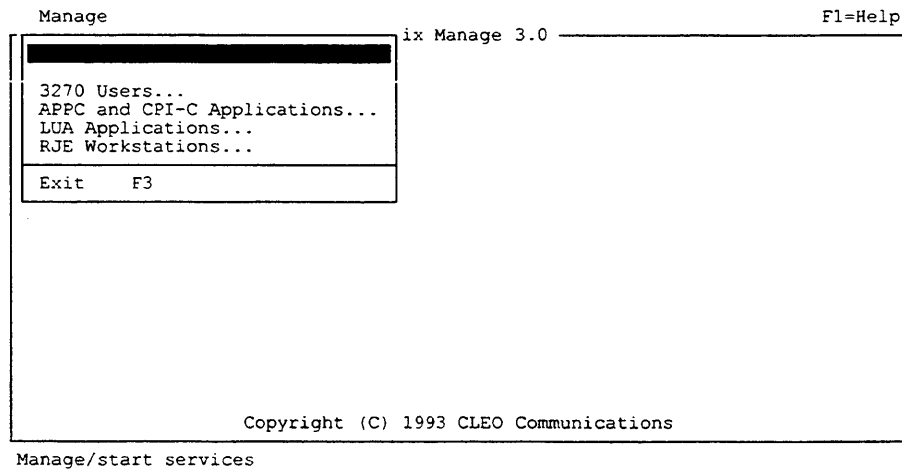
In addition, all active APPC CPI-C applications (whether invoked or invoking) can be viewed using the APPC and CPI-C Applications option from the Manage menu, and can be forced off if necessary.

To access help information on the Manage program, you can press **F1** from the main screen, or from any menu item or dialog. If you need more information on how to use the menu interface, refer to chapter 2 of the *Administration Guide*.

Starting and Stopping Services

Before you can start 3270 terminal emulation, APPC or LUA sessions, or RJE workstations, one or more local nodes and one or more links must be running. The local node provides an access point in the SNA network, and the link provides support for the communications adapter(s) in your UNIX computer.

To start or stop services, select Services from the Manage menu.



The Start/Manage Services dialog is displayed.

Using Manage

Manage F1=Help

Start/Manage Services

Status of services: ☒ All ☐ Active ☐ Inactive

Name	Description	Computer	Type	Started
NODE1	Local node for TEST	newserve	NODE	07-08 11:25
SDLC1	SDLC link to MAINHQ	newserve	SDLC	07-08 11:25
X25ONE	X.25 link to main network		X25	Inactive

^
█

v

< Zoom > < Start > < Start with Trace > < Terminate > < Done >

Copyright (C) 1993 CLEO Communications

This dialog contains the following:

Status of services

This radio group determines which services will be listed, as follows:

- All** This option, which is the default, provides status information on all services (active and inactive).
- Active** This option shows only the active services.
- Inactive** This option shows only the inactive services.

Name

Contains the name of the service — the link name, queued TP name, local node name, or NETMAN for the network management daemon program.

Description

Contains a description of the service. (For a link, local node, or queued TP, this is the description you included in the configuration file.)

Computer

If the service is active, this field shows the name of the computer on which the service is running.

Type

Contains the type of service — NODE for a local node; SDLC, X25, VX25, or TR for a link; or NTMN for the network management daemon program.

Note: The name, description, and type of the service are defined during configuration and cannot be changed here. Refer to **Configuring Services**, in the *Administration Guide*, for more information.

Started

Displays the current status of the service. One of the following values should appear in this column:

Inactive	The service is not active.
Starting	The service is being started.
(Date and Time)	The service is active, and was activated at the date and time shown. If an asterisk (*) appears next to the date, the LINKix configuration file has been changed and this service must be stopped and restarted for the changes to take effect.
Stopping	The service is being terminated.

Choose the < Zoom > pushbutton to access management options for an active service. It is disabled (grayed) for an inactive service.

Choose the < Start > pushbutton to start the service and record standard diagnostics in audit and error logs. Refer to the following sections for a discussion of starting each type of service. This pushbutton is disabled (grayed) for an active service.

Choose the < Start with trace > pushbutton to start the service. In addition to recording standard diagnostics in audit and error logs, this choice also provides an option for recording diagnostics in trace files. Refer to the following sections for a discussion of starting each type of service. This pushbutton is grayed for an active service.

Choose the < Terminate > pushbutton to terminate an active service. See **Terminating Services**, later in this chapter, for more information. This pushbutton is grayed for an inactive service.

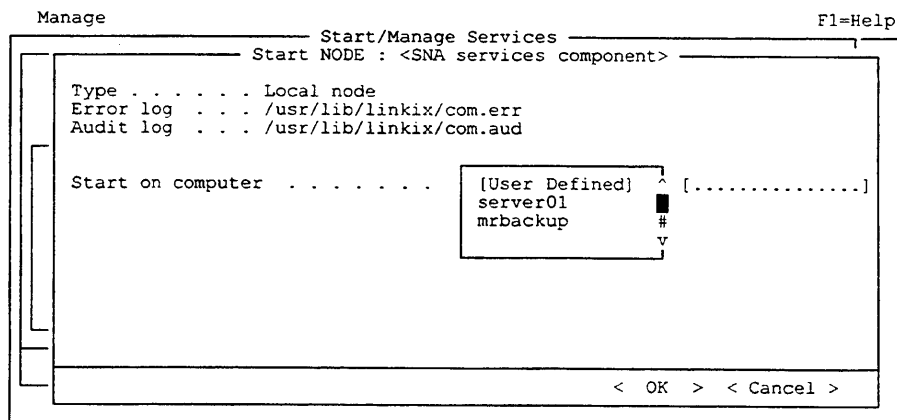
Starting a Local Node

Before you can use most of the facilities of LINKix, one or more local nodes must be started. You can start local nodes with or without tracing.

Starting a Local Node Without Tracing

This method for starting a local node records standard diagnostics for the node in audit and error log files, but does not activate tracing.

1. Select the local node in the list box on the Start/Manage Services dialog.
2. Choose < Start >. The Start Node dialog is displayed.



The title of this dialog contains the name and description of the node.

3. Complete the information in the Start Node dialog.

The following paragraphs describe the field in this dialog.

Type

Identifies what type of service is being started — local node.

Error Log

Displays the directory and error log file name.

Audit Log

Displays the directory and audit log file name.

Note: The log file names and directories are set up in the LINKix configuration and cannot be changed here. These files track the error and audit messages for LINKix. To configure these log files, refer to the *Administration Guide*.

Start on computer

Identifies the server on which to start the local node. By default this is the server on which you are running the Manage program. To start the local node on a different server, select the correct server from the list box. Alternatively, you can select User Defined from the list box and type the name of the connect server in the edit box to the right of the list box.

4. Choose < OK > to start the node, or select < Cancel > to abandon the operation.

If you choose < OK > to start the node, an acknowledgment dialog is displayed. Choose < Exit Message > to acknowledge the message and continue. The program returns to the Start/Manage Services dialog and refreshes the status column. It first reports a “starting” status, and then the date and time the node was started.

If you choose < Cancel >, you return to the Start/Manage Services dialog and abandon starting the node.

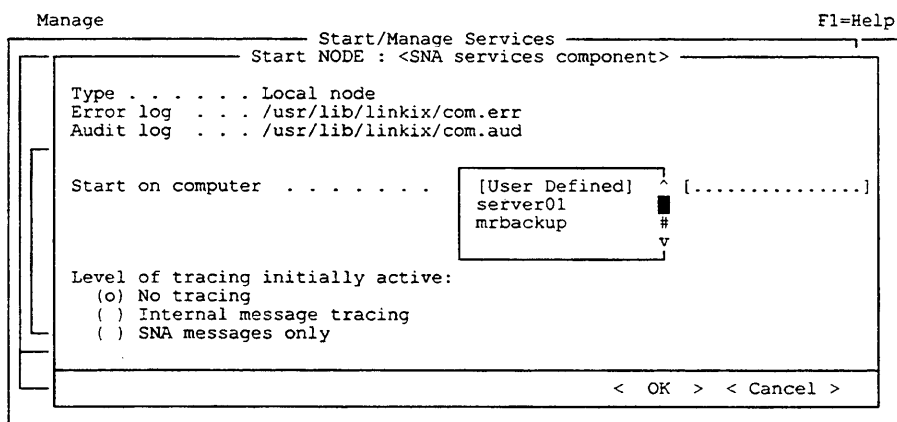
Starting a Local Node With Tracing

This method starts a local node and records standard diagnostics for the node in audit and error log files, and also provides for tracing to track the calls and activities of APIs and their communication links associated with the node. You can activate tracing when you start the node or turn it on later as required. When you activate tracing, LINKix gathers this diagnostic information and stores it in trace files. (For a discussion of trace files, see the *Administration Guide*.)

Using Manage

To start a local node with tracing:

1. Select the local node in the list box in the Mart/Manage Services dialog.
2. Choose < Start with Trace >. The Start Node dialog is displayed.



The dialog title contains the name and description of the node.

3. Complete the information in the Start Node dialog.

Type

Identifies the type of service being started — local node.

Error Log

Displays the directory and error log file name.

Audit Log

Displays the directory and audit log file name.

Note: The log file names and directories are setup during LINKix configuration and cannot be changed here. These files track the error and audit

messages for LINKix. To configure these log files, refer to the *Administration Guide*.

Start on computer

Identifies the server on which to start the local node. By default this is the server on which you are running the Manage program. To start the local node on a different server, select that server from the list box. Alternatively, you can select User Defined from the list box and type the name of the correct server in the edit box to the right of the list box.

Level of tracing initially active

Specifies the level of tracing to use for the local node. The choices areas follows:

No Tracing	Tracing is not activated on the local node. If you use this setting, you can activate tracing later using the audit logging and message tracing option for the node described later in this chapter.
Internal message tracing	Initiates tracing for all messages that are internal to the node including SNA messages.
SNA messages only	Initiates tracing for only SNA messages on the node.

4. Choose < OK > to start the node with tracing, or select < Cancel > to abandon the operation.

If you choose < OK > to start the local node, an acknowledgment message is displayed. Choose < Exit Message > to acknowledge the message and continue. The program returns to the Wart/Manage Services dialog and refreshes the status column. It first reports a “starting” status, and then the date and time the node was started.

If you choose < Cancel >, you abandon the operation and return to the Start/Manage Services dialog.

Starting a Link

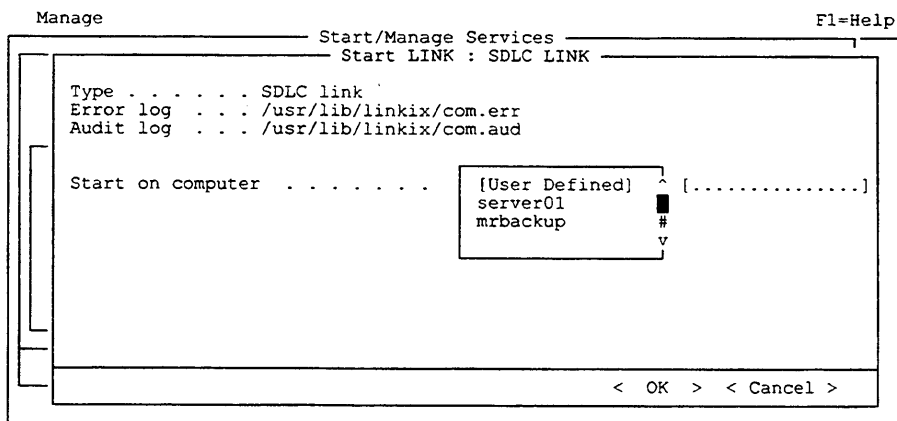
You can start links from the Start/Manage Services dialog with or without tracing; these options are explained in the next two sections.

Starting a Link Without Tracing

This method starts the link and records standard diagnostics for the link in audit and error log files, but does not activate tracing.

To start a link without tracing:

1. Select the link to start from the Mart/Manage Services dialog.
2. Choose Start. The Start Link dialog is displayed.



The title of this dialog contains the name and description of the link you selected.

3. Complete the information in the dialog.

Type

Identifies the type of link — SDLC, X.25, VX25 (Vendor X.25), or TR (Token Ring).

Error Log

Displays the directory and error log file name.

Audit Log

Displays the directory and audit log file name.

Note: The log file names and directories are set up in configuration and cannot be changed here. These files track the error and audit messages for this service. To configure these log files, refer to the *Administration Guide*.

Start on computer

Identifies the server on which to start the link. By default this is the server on which you are running the Manage program. To start the link on a different server, select the correct server from the list box. Alternatively, you can select User Defined from the list box and type the name of the correct server in the edit box to the right of the list box.

4. Choose < OK > to start the link, or select < Cancel > to abandon the operation.

If you choose < OK > to start the link, an acknowledgment message is displayed. Choose < Exit Message > to acknowledge the message and continue. The program returns to the Start/Manage Services dialog and refreshes the status column. It first reports a “starting” status, and then the date and time the link was started.

If you choose < Cancel > the operation is abandoned and you return to the Start/Manage Services dialog.

Starting a Link With Tracing

This method starts the link and records standard diagnostics in audit and error log files, and also provides tracing. When you activate tracing, LINKix gathers this diagnostic information and stores it in trace files. You can activate tracing when you start the service or turn it on later as required. For a discussion of trace files, see the *Administration Guide*.

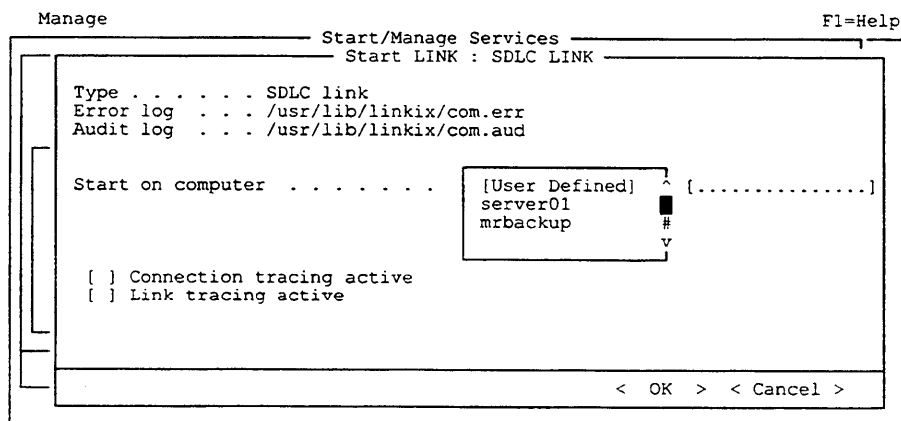
Using Manage

To start a link with tracing:

1. Select the link to start from the Start/Manage Services dialog.
2. Choose the < Start with Trace > pushbutton. The Start Link dialog is displayed with the name and description of the selected link in its title. Note that this dialog is slightly different for the different link types; the sections that follow describe the different versions.
3. Complete the information in the dialog.
4. Choose < OK > to start the link, or select < Cancel > to abandon the operation.

SDLC or X.25 links

The Start Link with Trace dialog for an SDLC link is shown below. The dialog for an X.25 link is not shown, but is the same as for SDLC. (However, the dialogs for Vendor X.25 and Token Ring links are different; see the following sections.)



The following paragraphs describe the Start Link with Trace dialog.

Type

Identifies the type of link being started — SDLC or X.25.

Error Log

Displays the directory and error log file name for LINKix.

Audit Log

Displays the directory and audit log file name for LINKix

Note: The log file names and directories are set up in configuration and cannot be changed here. These files track the error and audit messages for LINKix. To configure these log files, refer to the *Administration Guide*.

Start on computer

Identifies the server on which to start the link. By default this is the server on which you are running the Manage program. To start the link on a different server, select the correct server from the list box. Alternatively, you can select User Defined from the list box and type the name of the correct server in the edit box to the right of the list box.

Connection tracing active

This option activates tracing on the data flowing between the node and the link. The default setting initially disables connection tracing. To activate tracing initially, select this check box option. If you retain the default setting, you can activate tracing later. To do this, use the options described later in the **Managing Links** section.

Link tracing active

This option activates tracing on the link itself. The default setting initially disables link tracing. To activate tracing initially, select this check box option. If you retain the default setting, you can activate tracing later. To do this, use the options described later in the **Managing Links** section.

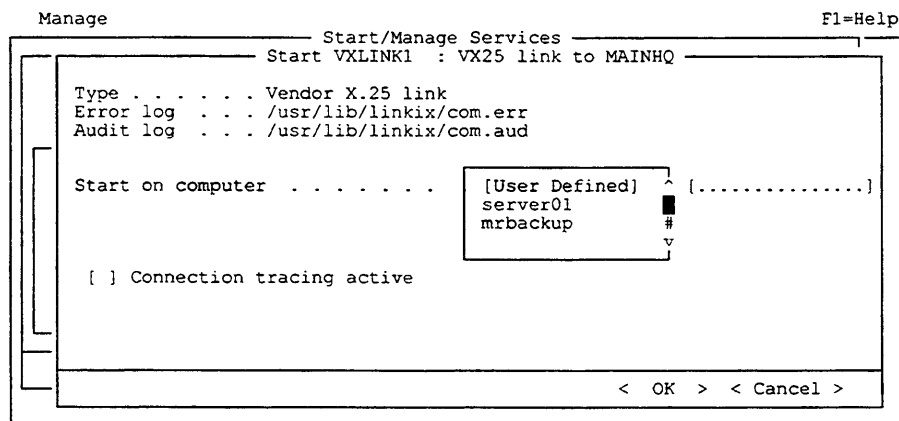
If you choose < OK > to start the service, an acknowledgment message is displayed. Choose < Exit Message > to acknowledge the message and continue. The program returns to the Start/Manage Services dialog and refreshes the status column. It first reports a “starting” status and then the date and time the service was started.

If you choose < Cancel >, the operation is abandoned and you return to the Start/Manage Services dialog.

Using Manage

Vendor X.25 links

The Start Link with Trace dialog for a Vendor X.25 link is shown below.



The following paragraphs describe the Start Link with Trace dialog.

Type

Identifies the type of link being started — Vendor X.25.

Error Log

Displays the directory and error log file name for LINKix.

Audit Log

Displays the directory and audit log file name for LINKix.

Note: The log file names and directories are setup in configuration and cannot be changed here. These files track the error and audit messages for LINKix. To configure these log files, refer to the *Administration Guide*.

Start on computer

Identifies the server on which to start the link. By default this is the server on which you are running the Manage program. To start the link on a different server, select

the correct server from the list box. Alternatively, you can select User Defined from the list box and type the name of the correct server in the edit box to the right of the list box.

Connection tracing active

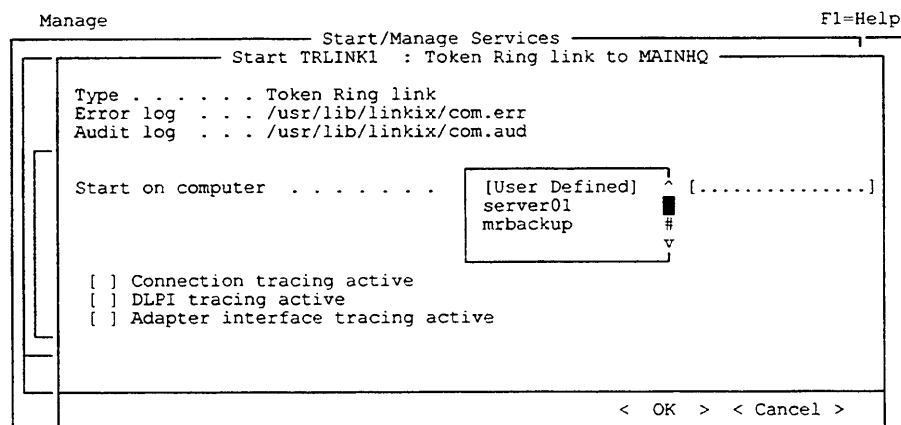
This option activates tracing between the node and the link. The default setting initially disables connection tracing. To activate tracing initially, select this check box option. If you retain the default setting, you can activate tracing later. To do this, use the options described later in the **Managing Links** section.

If you choose < OK > to start the service, an acknowledgment message is displayed. Choose < Exit > Message to acknowledge the message and continue. The program returns to the Start/Manage Services dialog and refreshes the status column. It first reports a “starting” status and then the date and time the service was started.

If you choose < Cancel > the operation is abandoned and you return to the Start/Manage Services dialog.

Token Ring links

The Start Link with Trace dialog for a Token Ring link is shown below.



The following paragraphs describe the Start Link with Trace dialog.

Type

Identifies the type of link being started — Token Ring.

Using Manage

Error Log

Displays the directory and error log file name.

Audit Log

Displays the directory and audit log file name.

Note: The log file names and directories are set up in configuration and cannot be changed here. These files track the error and audit messages for LINKix. To configure these log files, refer to the *Administration Guide*.

Start on computer

Identifies the server on which to start the link. By default this is the server on which you are running the Manage program. To start the link on a different server, select the correct server from the list box. Alternatively, you can select User Defined from the list box and type the name of the correct server in the edit box to the right of the list box.

Connection tracing active

This option activates tracing on the data flowing between the node and the link. The default setting initially disables connection tracing. To activate tracing initially, select this check box option. If you retain the default setting, you can activate tracing later. To do this, use the options described later in the **Managing Links** section.

DLPI tracing active

This option activates tracing on the Data Link Provider Interface, which is an industry-standard interface between different levels of the Token Ring software. The default setting initially disables DLPI tracing. To activate tracing initially, select this check box option. If you retain the default setting, you can activate tracing later. To do this, use the options described later in the **Managing Links** section.

Adapter interface tracing active

This option activates tracing on the data sent between the Token Ring software and the adapter card. The default setting initially disables adapter interface tracing. To activate tracing initially, select this check box option. If you retain the default setting, you can activate tracing later. To do this, use the options described later in the **Managing Links** section.

If you choose < OK > to start the service, an acknowledgment message is displayed. Choose < Exit Message > to acknowledge the message and continue. The program

returns to the Start/Manage Services dialog and refreshes the status column. It first reports a “starting” status and then the date and time the service was started.

If you choose < Cancel > the operation is abandoned and you return to the Start/Manage Services dialog.

Starting a Queued TP

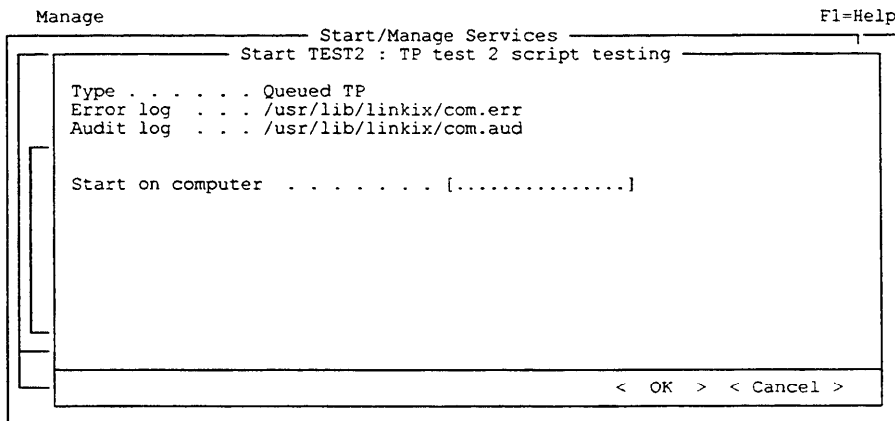
APPC or CPI-C invocable TPs which are configured as “queued operator-started” are generally referred to as “queued TPs”. You can start these TPs from the Start/Manage Services dialog, with or without tracing; these options are explained in the next two sections.

Starting a Queued TP Without Tracing

This method starts the TP and records standard diagnostics for the TP in audit and error log files, but does not activate tracing.

To start a queued TP without tracing:

1. Select the queued TP to start from the Wart/Manage Services dialog.
2. Choose < Start >. The Start TP dialog is displayed.



The title of this dialog contains the name and description of the TP you selected.

3. Complete the information in the dialog.

Type

Identifies the type of service — queued TP.

Error Log

Displays the directory and error log file name.

Audit Log

Displays the directory and audit log file name.

Note: The log file names and directories are set up in configuration and cannot be changed here. These files track the error and audit messages for this service. To configure these log files, refer to the *Administration Guide*.

Start on computer

Identifies the server on which to start the TP. By default this is the server on which you are running the Manage program. To start the TP on a different server, enter the server name in the edit box.

4. Choose < OK > to start the TP, or select < Cancel > to abandon the operation.

If you choose < OK > to start the service, an acknowledgment message is displayed. Choose < Exit Message > to acknowledge the message and continue. The program returns to the Start/Manage services dialog and refreshes the status column. It first reports a “starting” status, and then the date and time the service was started.

If you choose < Cancel > the operation is abandoned and you return to the Start/Manage Services dialog.

Starting a Queued TP With Tracing

This method starts the TP and records standard diagnostics in audit and error log files, and also provides tracing. When you activate tracing, LINKix gathers this diagnostic information and stores it in trace files. You can activate tracing when you start the TP or turn it on later as required. For a discussion of trace files, see the *Administration Guide*.

Using Manage

To start a queued TP with tracing:

1. Select the TP to start from the Wart/Manage Services dialog.
2. Choose the < Start with Trace > pushbutton. The Start TP dialog is displayed.

Manage Start/Manage Services F1=Help

Start TEST2 : TP test 2 script testing

Type Queued TP
Error log . . . /usr/lib/linkix/com.err
Audit log . . . /usr/lib/linkix/com.aud

Start on computer [backup.....]

[] Application interface tracing active

Primary trace file . . . [.....]
Secondary trace file . . [.....]

< OK > < Cancel >

The title of this dialog contains the name and description of the TP you selected.

3. Complete the information in the dialog.

Type

Identifies the type of service being started — queued TP.

Error Log

Displays the directory and error log file name.

Audit Log

Displays the directory and audit log file name.

Note: The log file names and directories are setup in configuration and cannot be changed here. These files track the error and audit messages. To configure these log files, refer to the *Administration Guide*.

Start on computer

Identifies the server on which to start the TP. By default this is the server on which you are running the Manage program. To start the TP on a different server, enter the server name in the edit box.

Application interface tracing active

Runs API tracing on the TP (see chapter 13 of the *Administration Guide* for more information). The default setting initially disables tracing. To activate tracing initially, select this check box option and specify the trace file or files (see below).

If you leave tracing inactive, you can activate it later. To do this, use the tracing option for queued TPs, described later in the **Managing Queued TPs** section.

Primary trace file / Secondary trace file

If you selected tracing using the previous check box, specify the file or files to which trace output is written, including the full path. You must specify the primary trace file; the secondary trace file is optional. If you specify both files, tracing is written alternately to the two files with a maximum size of 250kb for each file; if you specify only the primary trace file, tracing continues to this file with no limit on file size. See chapter 13 of the **Administration**

4. Choose < OK > to start the TP, or select < Cancel > to abandon the operation.

If you choose < OK > to start the TP, an acknowledgment message is displayed. Choose < Exit Message > to acknowledge the message and continue. The program returns to the Start/Manage Services dialog and refreshes the status column. It first reports a “starting” status and then the date and time the TP was started.

If you choose < Cancel >, the operation is abandoned and you return to the Mart/Manage Services dialog.

Starting the Network Management Daemon

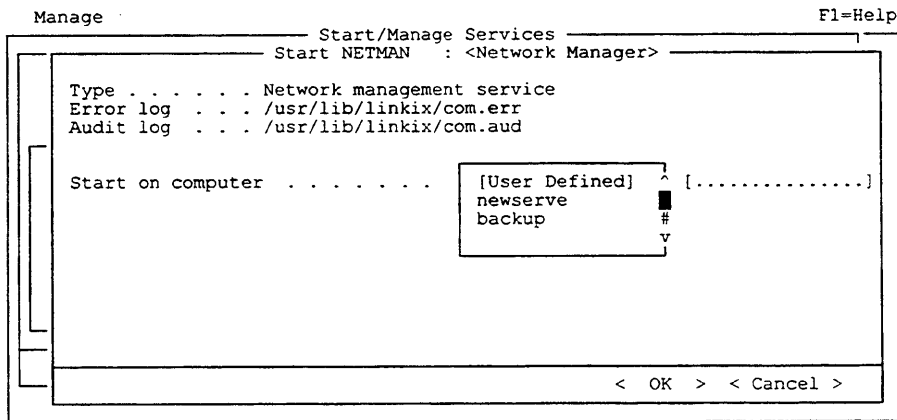
You can start the network management daemon with or without tracing; these options are explained in the next two sections.

Starting the Network Management Daemon Without Tracing

This method starts the network management daemon and records standard diagnostics in audit and error log files, but does not activate tracing.

Follow these steps to start the network management daemon without tracing:

1. Select NETMAN from the Start/Manage Services dialog.
2. Choose < Start >. The Start NETMAN dialog is displayed.



The title of this dialog contains the name and description of the network management daemon.

3. Complete the information in the dialog.

Type

Identifies the type of service — network management daemon.

Error Log

Displays the directory and error log file name.

Audit Log

Displays the directory and audit log file name.

Note: The log file names and directories are set up in configuration and cannot be changed here. These files track the error and audit messages for this service. To configure these log files, refer to the *Administration Guide*.

Start on computer

Identifies the server on which to start the network management daemon. By default this is the server on which you are running the Manage program. To start the network management daemon on a different server, select the correct server from the list box. Alternately, you can select User Defined from the list box and type the name of the server in the edit box to the right of the list box.

Note that, if you have configured LINKix to allow use of the UNIX Command Facility (UCF), all UCF commands will be issued on the server on which the network management daemon is running. See the *Administration Guide* for information on configuring UCF, and chapter 5 of this manual for information on using it.

4. Choose < OK > to start the network management daemon, or select < Cancel > to abandon the operation.

If you choose < OK > to start the service, an acknowledgment message is displayed. Choose < Exit Message > to acknowledge the message and continue. The program returns to the Start/Manage Services dialog and refreshes the status column. It first reports a “starting” status, and then the date and time the service was started.

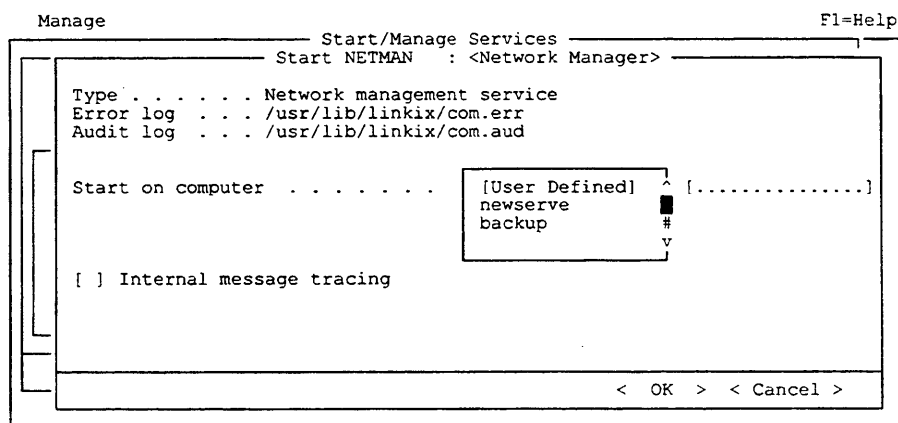
If you choose < Cancel >, the operation is abandoned and you return to the Start/Manage Services dialog.

Starting the Network Management Daemon With Tracing

This method starts the network management daemon and records standard diagnostics in audit and error log files, and also provides tracing on both the network management daemon and the UNIX Command Facility (UCF) daemon. When you activate tracing, LINKix gathers this diagnostic information and stores it in trace files. You can activate tracing when you start the service or turn it on later as required. For a discussion of trace files, see the *Administration Guide*.

To start the network management daemon with tracing:

1. Select NETMAN from the Start/Manage Services dialog.
2. Choose the < Start with Trace > pushbutton. The Start NETMAN dialog is displayed.



The title of this dialog contains the name and description of the network management daemon.

3. Complete the information in the dialog.

Type

Identifies the type of service being started — network management daemon.

Error Log

Displays the directory and error log file name.

Audit Log

Displays the directory and audit log file name.

Note: The log file names and directories are set up in the LINKix configuration and cannot be changed here. These files track the error and audit messages for LINKix. To configure these log files, refer to the *Administration Guide*.

Start on computer

Identifies the server on which to start the network management daemon. By default this is the server on which you are running the Manage program. To start the network management daemon on a different server, select the correct server from the list box. Alternatively, you can select User Defined from the list box and type the name of the server in the edit box to the right of the list box.

If you have configured LINKix to allow use of the UNIX Command Facility (UCF), all UCF commands will be issued on the server on which the network management daemon is running. See the *Administration Guide* for information on configuring UCF, and chapter 5 of this manual for information on using it.

Internal message tracing

Runs tracing for the network management daemon. The default setting is tracing disabled. To activate initial tracing, select this check box. If you keep the default setting, you can activate tracing later. To do this, use the message tracing option for the network management daemon, described later in the **Managing Services** section.

4. Choose < OK > to start the network management daemon, or select < Cancel > to abandon the operation.

If you choose < OK > to start the service, an acknowledgment message is displayed. Choose < Exit Message > to acknowledge the message and continue. The program returns to the Start/Manage Services dialog and refreshes the status column. It first reports a “starting” status and then the date and time the service was started.

If you choose < Cancel >, the operation is abandoned and you return to the Start/Manage Services dialog.

Terminating Services

When you finish using LINKix services, you can terminate them using the Manage program. This frees up memory and can improve processing performance. You may also need to terminate one or more services in response to some problem that may develop.

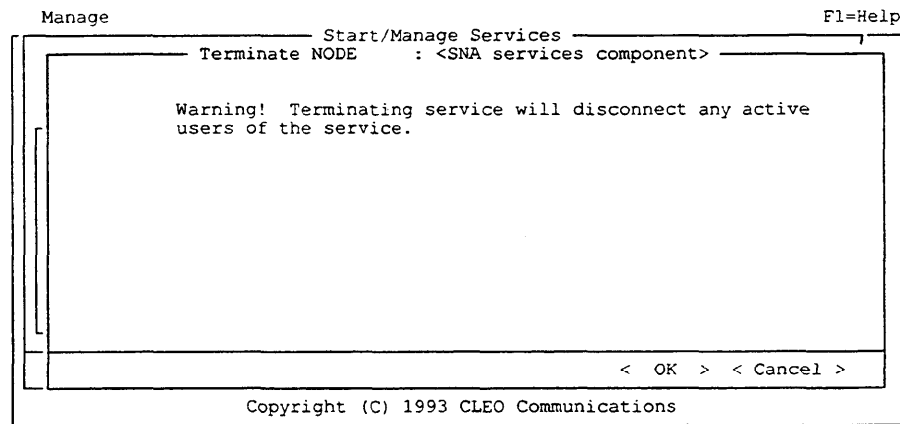
Terminating a link also terminates all active connections and sessions that are using the service, but all other links and their LUs remain active. When you terminate a local node, however, you terminate active sessions on all the node's connections, even though the links used by the connections remain active. In terminating an SDLC link, consider incoming calls for which you may want to keep the service active.

Terminating a queued TP will end any conversations it has with other TPs; they will receive a return code indicating that their partner TP has terminated.

Abruptly terminating sessions may cause problems. If possible, observe customary log-off procedures or end your communications in a conventional manner before you terminate services. Abruptly terminating sessions can leave resources on the other end of the communications link waiting for a response or completing unnecessary time-out periods. For example, you could experience difficulty in returning to a host application because you did not exit the host application or log off the host appropriately. The host can continue to see you as being logged on, even though you terminated the session earlier.

To Follow these steps to terminate a service:

1. Select the service from the Start/Manage Services dialog.
2. Choose the < Terminate > pushbutton; the Terminate Service dialog is displayed.



Note that the title includes the name and description of the selected service.

The dialog informs you that terminating the service disconnects all active users of the service.

If you choose < OK > to terminate the service, an acknowledgment message is displayed.

Choose < Exit Message > to acknowledge the message and continue. The program returns to the Start/Manage Services dialog. The status column first reports "stopping" and then "inactive" as the service is stopped.

If you choose < Cancel > or press **ESC** to abandon termination and return to the previous dialog.

Managing an Active Link

To view or alter diagnostic settings for an active link.

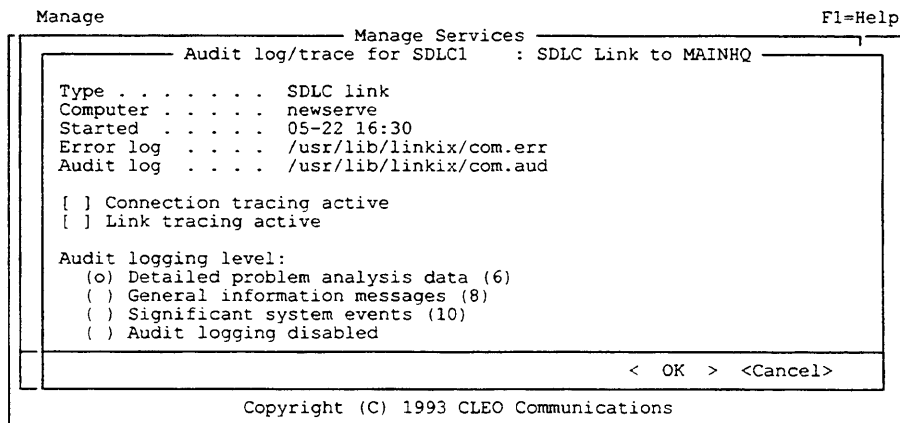
1. Select the link from the Manage Services dialog.
2. Choose < Zoom > to display the Audit Log/Trace for Link dialog.
3. Review the information and make any necessary changes.
4. Choose < OK > to implement any changes in settings, or select < Cancel > to retain the existing settings and return to the Manage Services dialog.

If you choose < OK > to implement changes, the program responds with an acknowledgment message. Choose < Exit Message > to continue and return to the Manage Services dialog.

The Audit Log/Trace for Link dialog is slightly different for the different link types; the following sections describe the different versions.

SDLC or X.25 links

The Audit Log/Trace for Link dialog for an SDLC link is shown below. The dialog for an X.25 link is not shown, but is the same as for SDLC.



The following paragraphs describe the Audit Log/Trace for Link dialog.

Type

Identifies the type of link (SDLC or X.25).

Computer

Identifies the server on which the link is running.

Started

Shows the date and time the service started.

Error log

Displays the current error log file name and directory.

Audit log

Displays the current audit log file name and directory.

Note: The service type, log file names and paths are set up in LINKix configuration and cannot be changed here. These files contain error and audit messages. To configure these log files, refer to the *Administration Guide*.

Connection tracing active

Enables tracing of all data flowing between the node and this link. This is used for diagnostic purposes. Tracing is disabled by default (unless you started the link using the Start with Trace option). Because tracing degrades performance, it should not be used unless specifically required to solve a problem.

Link tracing active

Enables tracing of all data flowing on this link. This is used for diagnostic purposes as for connection tracing above.

Audit logging level

Choose the level of information you want recorded in the audit log file for this link.

Detailed problem analysis data (6)

This option causes all messages at levels 6,8, and 10 to be logged.

General information messages (8)

This option causes all messages at levels 8 and 10 to be logged.

Significant system events (10)

This option causes all messages at level 10 only to be logged.

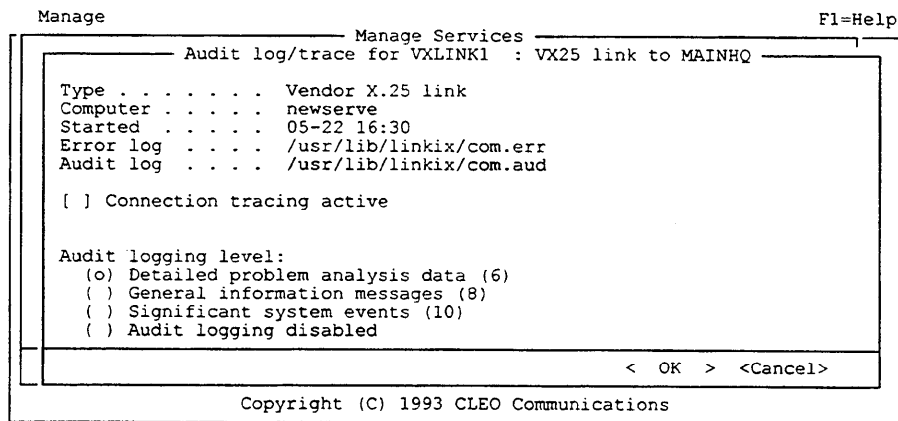
Using Manage

Audit logging disabled

This option suppresses all level 6-10 audit log messages. Level 6 messages provide very detailed information which is useful for problem tracing. Level 8 provides general information messages of system events. Level 10 provides only significant system events.

Vendor X.25 links

The Audit Log/Trace for Link dialog for a Vendor X.25 link is shown below.



The following paragraphs describe the Audit Log/Trace for Link dialog.

Type

Identifies the type of link (Vendor X.25).

Computer

Identifies the server on which the link is running.

Started

Shows the date and time the service started.

Error log

Displays the current error log file name and directory.

Audit log

Displays the current audit log file name and directory.

Note: The service type, log file names, and paths are set up in configuration and cannot be changed here. These files contain error and audit messages for LINKix. To configure these log files, refer to the *Administration Guide*.

Connection tracing active

Enables tracing of all data flowing between the node and this link. This is used for diagnostic purposes. Tracing is disabled by default (unless you started the link using the Start with Trace option); it will degrade performance, and should not be used unless specifically required to solve a problem.

Audit logging level

Choose the level of information you want recorded in the audit log file for this link.

Detailed problem analysis data (6)

This option causes all messages at levels 6, 8, and 10 to be logged.

General information messages (8)

This option causes all messages at levels 8 and 10 to be logged.

Significant system events (10)

This option causes all messages at level 10 only to be logged.

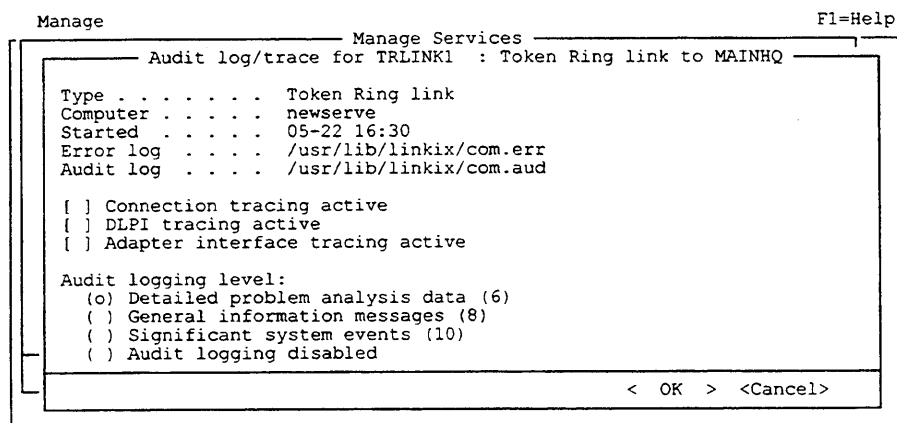
Audit logging disabled

This option suppresses all level 6-10 audit log messages. The default log level is taken from the configuration file — the level you specified in the Audit/Error logs dialog. Level 6 messages provide very detailed information which is useful for problem tracing. Level 8 provides general information messages of system events. Level 10 provides only significant system events.

Using Manage

Token Ring Links

The Audit Log/Trace for Link dialog for a Token Ring link is shown below.



The following paragraphs describe the Audit Log/Trace for Link dialog.

Type

Identifies the type of link (Token Ring).

Computer

Identifies the server on which the link is running.

Started

Shows the date and time the service started.

Error log

Displays the current error log file name and directory.

Audit log

Displays the current audit log file name and directory.

Note: The service type, log file names, and paths are set up in configuration and cannot be changed here. These files contain error and audit messages. To configure these log files, refer to the *Administration Guide*.

Connection tracing active

This option activates tracing on the data flowing between the node and the link. The default setting initially disables connection tracing. To activate tracing initially, select this check box option. If you retain the default setting, you can activate tracing later. To do this, use the options described later in the **Managing Links** section.

DLPI tracing active

This option activates tracing on the Data Link Provider Interface, which is an industry-standard interface between different levels of the Token Ring software. This is used for diagnostic purposes. Tracing is disabled by default (unless you started the link using the Start with Trace option); it will degrade performance, and should not be used unless specifically required to solve a problem.

Adapter interface tracing active

This option activates tracing on the data sent from the Token Ring software to the adapter card. This is used for diagnostic purposes as for connection tracing above.

Audit logging level

Choose the level of information you want recorded in the audit log file for this link.

Detailed problem analysis data (6)

This option causes LINKix to log all messages at levels 6,8, and 10.

General information messages (8)

This option causes LINKix to log all messages at levels 8 and 10.

Significant system events (10)

This option causes LINKix to log all messages at level 10 only.

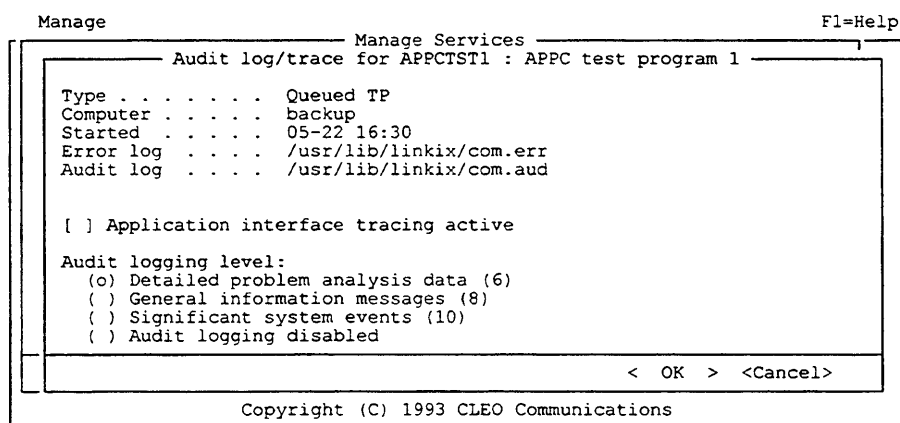
Audit logging disabled

This option suppresses all level 6-10 audit log messages. Level 6 messages provides very detailed information which is useful for problem tracing. Level 8 provides general information messages of system events. Level 10 provides only significant system events.

Managing an Active Queued TP

To view or alter diagnostic settings for an active queued TP, follow these steps:

1. Select the queued TP from the Manage Services dialog.
2. Choose the < Zoom > pushbutton. The Audit Log/Trace for TP dialog is displayed.



3. Review the information and make any necessary changes.

Type

Identifies the type of service - queued TP.

Computer

Identifies the server on which the TP is running.

Started

Shows the date and time the service started.

Error log

Displays the current error log file name and directory.

Audit log

Displays the current audit log file name and directory.

Note: The service type, log file names, and paths are set up in configuration and cannot be changed here. These files contain error and audit messages for LINKix. To configure these log files, refer to the *Administration Guide*

Application interface tracing active

Enables API tracing on the TP (see chapter 13 of the *Administration Guide* for more information). Tracing is disabled by default (unless you started the TP using the Start with Trace option); it will degrade performance, and should not be used unless specifically required to solve a problem.

Audit logging level

Choose the level of information you want recorded in the audit log file for this TP.

Detailed problem analysis data (6)

This option causes all messages at levels 6,8, and 10 to be logged.

General information messages (8)

This option causes all messages at levels 8 and 10 to be logged.

Significant system events (10)

This option causes LINKix to log all messages at level 10 only.

Audit logging disabled

This option suppresses all level 6-10 audit log messages. Level 6 messages provide very detailed information which is useful for problem tracing. Level 8 provides general information messages of system events. Level 10 is only significant system events.

Choose < OK > to implement the changes in the audit log levels. The program responds with an acknowledgment message. Choose < Exit Message > to continue and return to the Manage Services dialog.

Choose < Cancel > to retain the settings and return to the Manage Services dialog.

Managing Node Resources

The following components are configured and managed as local node resources — that is, they are associated with a particular local node:

- ◆ 3270 LUs
- ◆ APPC LUs
- ◆ LUA LUs
- ◆ RJE LUs
- ◆ Connections
- ◆ Audit log and trace levels

Selecting a Node Resource

The Manage program reports status information on these resources and provides management options for them. To select a node resource for extended management options:

1. Select Services from the Manage menu to bring up the Manage Services dialog.
2. Select the local node on which the resource is configured.
3. Choose the < Zoom > pushbutton. This displays the Select Resource Type for Zoom dialog, which contains the following resource types.

3270 Logical Units

Provides an overview of all 3270 LUs configured on this local node.

APPC Logical Units

Provides an overview and management options for all APPC LUs configured on this local node, details for a selected local LU, and sessions with a partner LU.

LUA Logical Units

Provides an overview of all LUA LUs configured on this local node.

RJE Logical Units

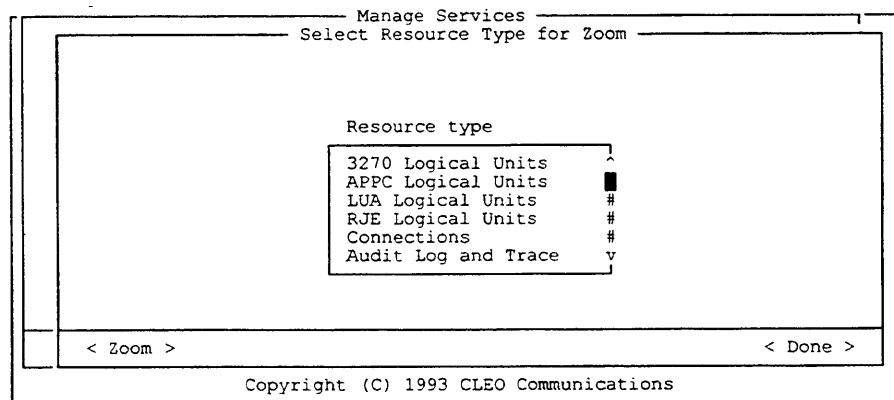
Provides an overview of all RJE LUs configured on this local node.

Connections

Provides an overview of all connections owned by this local node, and management options for them.

Audit Log and Trace

Provides an overview of the audit log level and tracing options for this local node, and allows you to change these settings.



4. Select a resource type in this dialog.

5. To access management operations for the selected resource, choose the < Zoom > pushbutton.

To cancel the request and return to the previous dialog, choose < Done > or press **ESC**.

Managing 3270 LUs

The Manage program provides two dialogs for 3270 LU management. One box shows the status of 3270 LUs; another offers detailed information, and a reset option, for a selected 3270 LU. The following sections describe these dialogs and their management options.

Status of 3270 LUs

The 3270 LUs dialog shows information on 3270 LUs configured on a particular local node. To access this information:

1. From the Resource Type dialog, select the 3270 LU option.
2. Choose Zoom. This displays the 3270 LUs dialog. 3270 LUs on the selected local node.

Manage F1=Help

3270 LUs on NODE : <SNA services component>

LU Status: (o) All () Inuse () SSCP () In sess () Available

Name	Description	LU #	User name	Started
COL03	Colour Video 03	3	Not in use	
COL04	Colour Video 04	4	bp	05-23 16:14
COL05	Colour Video 05	5	Not in use	
COL06	Colour Video 06	6	Not in use	
PRT308	LU Type 3 Printer	8	Not in use	

< Zoom > < Done >

Copyright (C) 1993 CLEO Communications

3. Use the LU Status radio group to select which LUs to view.
4. Select an LU and choose < Zoom > for more information and options on this LU, or choose < Done > to cancel the operation.

The following information is displayed in the 3270 LUs dialog:

LU Status

This radio group allows you to select which LUs to view. The options are:

All	All configured 3270 LUs are displayed.
In use	All LUs which are being used by an enabled session (whether or not a connection to the remote system has been established) are displayed.
SSCP	All LUs which are communicating with SSCP (the host's network control program) are displayed.
In sess	All LUs which are logged on to a host application are displayed.
Available	All LUs which are not being used by an enabled session (and are therefore available for use) are displayed.

Name

Displays the name of the 3270 LU.

Description

Displays the description of the 3270 LU.

LU

Displays the LU number.

Note: The 3270 LU name, description, and LU number are taken from the configuration file. See **Configuring for 3270**, in the *Administration Guide*, for more information.

User name

Displays the name of the user currently using this LU, or a "Not in use" notice if the LU is not being used. The information displayed on the screen is refreshed every 10 to 15 seconds; when a session is activated or deactivated, the dialog reflects the change.

Using Manage

Started

If the LU is in use by a 3270 user, this field shows the date and time the session was established with the host computer.

Choose < Zoom > for detailed information or a reset option for a selected 3270 LU.

Choose < Done > or press **ESC** to return to the previous dialog.

Status of a Selected 3270 LU

To view detailed information on an individual 3270 LU, or to reset the session status for a bound 3270 LU:

1. Select the 3270 LU from the dialog.
2. Choose < Zoom >. This displays the 3270 LU dialog.

The screenshot shows a terminal window titled 'Manage' with a sub-dialog titled 'Manage Services'. The sub-dialog has a title bar that says '3270 LUs on NODE : <SNA services component>' and '3270 LU COL04 : Colour Video 04'. Inside the dialog, there is a list of fields and their values:

LU type	Display
LU number	04
LU pool name	COLPOOL
User name	bp
User started at	05-23 16:14
Computer name	newserve
LU status	Active
Session status	Bound
Host application name	TESTAPP1
Connection	CONN1
Link	SDLCLINK

At the bottom of the dialog, there are two buttons: '< Reset >' on the left and '< Done >' on the right. The top right corner of the window has 'F1=Help'.

3. Choose < Reset > to reset the session status for the LU, or choose < Done > to cancel the operation and return to the previous dialog.

The following paragraphs describe the fields in the 3270 LU dialog for the selected LU. Note that the title of the dialog contains the name and description of the selected

LU. The information in this dialog comes from the configuration file or from the status of the host session.

LU type

The type of LU specified in the configuration file — display or printer.

LU number

The number assigned to the LU in the configuration file.

LU pool name

The name of the pool, if the LU is a member of an LU pool.

User name, User started at, Computer name

The date and time when the 3270 session became active and the computer on which the 3270 emulation program is running (if the LU is in use by a 3270 session). If the LU is assigned to a 3270 session but the session is not active, the message “LU is assigned to a 3270 session” is displayed; if the LU is not assigned to any session, the message “Not in use” is displayed.

LU status

The status of the LU: Active or Inactive.

Session status

If LU status is Active, this field reports the status of the session: unbound (enabled but not logged onto a host application) or bound (logged on to a host application).

Host application name

If the session is bound to a host application, this field shows the host application name. Otherwise, this field is blank.

Connection

The name of the connection you are using to the host computer.

Link

The name of the link you are using from LINKix to the host computer.

Choose < Reset > to reset the session status for a bound 3270 LU. Reset terminates only the PLU-SLU session; the LU remains active and the SSCP logon message

Using Manage

reappears. Reset frees locked sessions, which display an X on the 3270 emulation status line and ignore keystrokes.

Choose < Done > or press **ESC** to return to the previous dialog.

Managing APPC LUs

The Manage program provides two dialogs for APPC LU management. The first box contains information and some management options for all APPC local LUs; the second contains information and management options for individual APPC LUs. The following sections describe these dialogs and their management options.

Options for APPC LUs

The APPC LUs dialog shows information on all local APPC LUs configured on a particular local node. From this dialog, you can select and deactivate a single local LU, or zoom to another dialog for other APPC options.

To access these APPC LU management options:

1. From the Select Resource Type for Zoom dialog, select the APPC Logical Units option.
2. Choose < Zoom >. This displays the APPC LUs dialog.

Manage F1=Help

Manage Services

APPC LUs on NODE : <SNA services component>

LU Status: (o) All () In Session () Not in session

Name	Description	# Sessions
BOUNCELU	Local LU for Bounce	0
RANDOMLU	Local LU for Random	0
TPLU1	For sample TP no. 1	0
TPLU2	For sample TP no. 2	0

< Zoom > < Deactivate > < Refresh > < Done >

Copyright (C) 1993 CLEO Communications

3. Use the LU Status radio group to select which LUs to view.

Using Manage

4. Choose < Zoom > to view details of the partner LUs and modes for the highlighted local LU.

Choose < Deactivate > to terminate all sessions of the highlighted local LU with all its partner LUs. This choice displays a dialog which warns that terminating the LU also terminates all active conversations. You can select that sessions should only be deactivated once all active conversations are ended. Choose < OK > to complete the command or < Cancel > to abandon the operation.

Choose < Refresh > to update the screen information.

The APPC LUs dialog contains the name and description of the node in its title. By default, it lists all APPC LUs configured on the local node. The following information is displayed in this dialog:

LU Status

This radio group allows you to select which LUs to view. The options areas follows:

All	All configured APPC LUs are displayed.
In session	All LUs which are in session with at least one partner LU are displayed.
Not in session	All LUs which are not in session with any partner LU are displayed.

Name

Displays the name of the local APPC LU.

Description

Displays the description of the APPC LU.

Note: The LU name and description are specified in the configuration file. See **Configuring for APPC and CPI-C**, in the *Administration Guide*.

Sessions

Displays the number of active sessions between the local APPC LU and all its partner LUs.

Options for Individual APPC LUs

The APPC LU dialog shows information on partner LUs that can communicate with a specific local LU. From this dialog, you can activate and deactivate sessions with particular partner LUs, or view details of all sessions associated with a partner LU.

To access the APPC LU dialog and management options for a specific local LU:

1. From the APPC LUs dialog, select the local LU.
2. Choose < Zoom >. This displays the APPC LU dialog.

Partner LU	Description	Mode	Connection	Num	Sess	limit
RANDOMLU	Remote Random	MODE1	LINK1	0	0	0
RANDOMLU	Remote Random	MODE2	LINK1	0	0	0
RANDOMLU	Remote Random	SNASVCMG		0	2	#

< Activate > < Deactivate > < Refresh > < Zoom > < Done >

The APPC LU dialog contains the name and description of the local LU in its title. It lists all the partner LUs, together with corresponding modes, that the local LU can communicate with. The following information is displayed in the APPC LU dialog.

Partner LU

Displays the names of the partner LUs with which the local LU can communicate.

Description

Displays the description of each partner LU.

Using Manage

Mode

Displays the name of the mode that can be used in communications between the local LU and partner LU. If the local LU uses several modes with a specific partner, each mode is displayed in the list.

Connection

Displays the connection used by this mode.

Num Sess

Displays the number of active sessions for this mode.

Sess limit

Displays the maximum number of parallel sessions for this mode.

The partner LU name, description, mode, connection, number of sessions, and session limit are defined during configuration, but the number of active sessions and session limit can change. See **Configuring for APPC and CPI-C**, in the *Administration Guide*, for more information.

3. Choose < Activate > to activate sessions with a selected partner LU. The program displays one of two Activate APPC Sessions dialogs, depending on whether sessions are already active. Refer to the section **Activating Remote APPC LU Sessions** for an explanation of these dialogs. (This choice is grayed if the session limit has been reached, if the selected mode is the “service manager” mode SNASVCMG, or if the partner LU is a local LU.)

Choose < Deactivate > to deactivate sessions with a selected partner LU. The program displays the Deactivate APPC Sessions dialog. Refer to the section **Deactivating a Session with a Partner LU** for an explanation of this dialog. (This choice is grayed if no sessions are active, if the selected mode is the “service manager” mode SNASVCMG, or if the partner LU is a local LU.)

Choose < Refresh > to refresh the screen information and display any status change.

Choose < Zoom > to display details of all sessions which the local LU has with the selected partner LU and mode. The program displays the Sessions with Partner LU dialog. Refer to the section **Viewing Sessions with a Partner LU** for an explanation of this dialog.

Choose < Done > or press **ESC** to return to the previous dialog.

Activating Remote APPC LU Sessions

To start a session with a partner LU:

1. Choose < Activate > from the APPC LU dialog. This displays one of the two Activate Session dialogs, explained in the following sections, depending on whether any sessions are already active.
2. Complete the information in the dialog.
3. Choose < OK > to activate the sessions, or < Cancel > to abandon the operation.

If you choose < OK > to activate session, a message box informs you whether the sessions were activated successfully. If the program can activate the sessions, it displays an acknowledgement message. Choose < Exit Message > to respond to this message. If the program cannot activate all sessions, it displays a warning message that states the system was unable to activate all of the sessions, but reset session limits worked correctly. The new number of sessions and session limits in the Remote LU dialog is updated.

Activating APPC Sessions When No Active Sessions Exist

When no sessions are active, the session limits are reset. The default values in the Activate APPC Sessions dialog are supplied from configuration. The following paragraphs describe the Activate APPC Sessions dialog when no sessions are active, as shown below.

```

Manage                                     Manage Services                                     Fl=Help
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| APPC LUs on NODE      : <SNA services component> |
| BOUNCELU : Local LU for Bounce |
| Activate sessions with RANDOMLU |
+-----+-----+-----+-----+-----+-----+-----+-----+
|
| Parallel session limit . . . . . [9..]
|
| Minimum number of contention winner sessions:
|   for Local LU . . . . . [0..]
|   for Partner LU . . . . . [0..]
|
| Number of sessions auto started . . . . . [0..]
|
|                                     < OK > <Cancel>
|
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Using Manage

For more information on the meaning of the fields in this dialog, see **Session limits and contention winner sessions** in the APPC chapter of the *Administration Guide*.

Parallel session limit

Specifies the maximum number of parallel sessions permitted between these two LUs on this mode. If you reduce the parallel sessions limit, the new limit remains as long as any sessions are active. To raise the parallel session limit, deactivate all the sessions, and then reactivate the sessions and specify a higher limit.

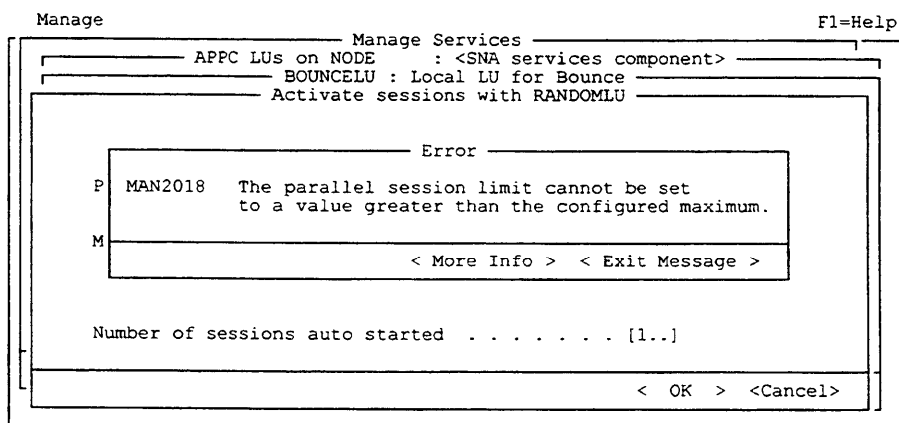
Minimum number of contention winner sessions

Specifies the minimum number of sessions for which local and partner LUs should be the contention winner.

LU for Local Enter the minimum number of sessions for which the local LU can be a contention winner.

LU for Partner Enter the minimum number of sessions for which the partner LU can be a contention winner.

The sum of these two values cannot exceed the value specified for the Parallel Session Limit. If it does, an error message is displayed, as shown below. To respond to the message, choose Exit Message. Then reduce the total for the maximum number of contention winner sessions in the Activate APPC Sessions dialog.



Number of sessions auto started

Specifies the number of sessions to start. The value entered must not exceed the minimum number of contention winner sessions for the local LU.

Activating APPC Sessions When Active Sessions Exist

The following paragraphs describe the Activate APPC Sessions dialog when sessions are already active, as shown below.

The screenshot shows a window titled 'Manage' with a sub-header 'Manage Services' and a button 'F1=Help'. Inside the window, there are three lines of text: 'APPC LUs on NODE : <SNA services component>', 'BOUNCELU : Local LU for Bounce', and 'Activate sessions with RANDOMLU'. Below this, there is a large rectangular area containing the following text:

```

Current session limit . . . . . 5
Current session count . . . . . 1
Number of sessions to activate. . . . [4..]

```

At the bottom right of the dialog, there are two buttons: '< OK' and '> <Cancel>'.

Current session limit

Specifies the maximum number of parallel sessions permitted between the two LUs on this mode. To change the parallel session limit, deactivate all sessions. Then reactivate the sessions and specify the new limit. For more information on parallel session limits, see **Session limits and contention winner sessions** in the APPC chapter of the *Administration Guide*.

Current session count

Specifies the number of sessions currently active on this mode.

Using Manage

Number of sessions to activate

Enter the number of new sessions to activate on this mode. The sum of the “Current Sessions Count” and the “Number of Sessions to Activate” must not exceed the current session limit. If it does, an error message is displayed.

Choose < OK > to activate the sessions. After you activate the sessions, the Remote LU dialog reappears with the new number of sessions and session limits.

If the program can activate the sessions, it displays an acknowledgement message. Choose < Exit Message > to respond to this message. If the program cannot activate all sessions, it displays a warning message that states the system was unable to activate all of the sessions.

Choose < Cancel > to abandon this operation and return to the previous dialog.

Deactivating Sessions With a Partner LU on a Specific Mode

To stop sessions with a partner LU on a specific mode, do the following:

1. Choose < Deactivate > from the APPC LU dialog. This displays the Deactivate APPC Sessions dialog.

The screenshot shows a multi-level dialog box titled 'Manage Services' with a 'Fl=Help' button in the top right corner. Inside, there's a section 'APPC LUs on NODE : <SNA services component>' with a sub-section 'BOUNCELU : Local LU for Bounce'. The main content area is titled 'Deactivate sessions with RANDOMLU' and contains the following options:

- [] All modes
Mode [MODE1]
- (o) Let current conversations end
- () Terminate current conversations

At the bottom right, there are buttons for '< OK >' and '<Cancel>'.

2. Complete the information in the dialog.

All modes

Select this check box to deactivate all sessions, or leave it unchecked if you want to deactivate only the sessions using a particular mode.

Mode

If you do not select the “All modes” check box, specify the mode on which to deactivate sessions.

Let current conversations end

Choose this radio button to deactivate the sessions after the current conversations end.

Terminate current conversations

Choose this radio button to terminate the current conversations and deactivate the sessions immediately. Any applications using these sessions will receive error codes indicating the session failure.

3. Choose < OK > to deactivate the sessions, or choose < Cancel > to abandon the operation.

If you choose < OK >, a message is displayed to inform you whether the sessions were deactivated successfully. Choose < Exit Message > to acknowledge this message.

If you choose < Cancel > you return to the previous dialog.

Manage

Manage Services

F1=Help

APPC LUs on NODE : <SNA services component>

BOUNCELU : Local LU for Bounce

Sessions with partner LU RANDOMLU, mode MODEL

TP Name	Computer name	Started
BOUNCETP	backup	01-09 10:24

< Refresh >

< Done >

Managing LUA LUs

The Manage program provides two dialogs for LUA LU management. One box shows the status of LUA LUs; another offers detailed information, and a reset option, for a selected LUA LU. The following sections describe these dialogs and their management options.

Status of LUA LUs

The LUA LUs dialog shows information on LUA LUs configured on a particular LINKix local node. To access this information:

1. From the Resource Type dialog select the LUA LU option.
2. Choose < Zoom >. This displays the LUA LUs dialog. By default, this shows the status of all LUA LUs on the selected local node.

Manage F1=Help

LUA LUs on NODE : <SNA services component>

LU Status: (o) All () Inuse () SSCP () In sess () Available

Name	Description	LU #	User name	Started
LUA03	LUA LU number 3	3	Not in use	
LUA04	LUA LU number 4	4	steve	05-23 16:14
				#
				#
				#
				#
				#
				#
				v

< Zoom > < Done >

Copyright (C) 1993 CLEO Communications

Using Manage

3. Use the LU Status radio group to select which LUs to view.

LU Status

This radio group allows you to select which LUs to view:

All	All configured LUA LUs are displayed.
In use	All LUs which are being used by an LUA application (whether or not a connection to the remote system has been established) are displayed.
SSCP	All LUs which are communicating with SSCP (the host's network control program) are displayed.
In sess	All LUs which are logged on to a host application are displayed.
Available	All LUs which are not being used by an LUA application (and are therefore available for use) are displayed.

4. Select an LU and choose < Zoom > for more information and options on the LU, or choose < Done > to cancel the operation and return to the previous dialog.

The following information is displayed in the LUA LUs dialog:

Name

Displays the name of the LUA LU.

Description

Displays the description of the LUA LU.

LU

Displays the LU number.

Note: The LUA LU name, description, and LU number are taken from the configuration file. See **Configuring for LUA**, in the *Administration Guide*, for more information.

User name

If the LU is in use by an LUA application, this column displays the user ID with which the application was started.

Started

Displays a “Not in use” notice or the date and time the session was established with the host computer. This information is dynamic because the system refreshes the screen every 10 to 15 seconds. Whenever a session is activated or deactivated, the dialog reflects the change.

Status of a Selected LUA LU

To view detailed information on an individual LUA LU, or to reset the session status for a bound LUA LU, do the following:

1. Select the LUA LU from the dialog.
2. Choose < Zoom >. This displays the LUA LU dialog.

Manage F1=Help

Manage Services

LUA LUs on NODE : <SNA services component>

LUA LU LUA04 : LUA LU number 4

LU type	LUA
LU number	04
LU pool name	NEWPOOL
User name	steve
User started at	05-23 16:14
Computer name	newserve
LU status	Active
Session status	Bound
Host application name	TESTLUA
Connection	CONN1
Link	SDLCLINK

< Reset > < Done >

Note that the title of the dialog contains the name and description of the selected LU. The information found in this dialog comes from the configuration file or from the status of the host session.

LU type

The type of LU — LUA.

Using Manage

LU number

The number assigned to the LU in the configuration file.

LU pool name

If the LU is a member of an LUA LU pool, this field is the name of the pool.

User name, User started at, Computer name

If the LU is in use by an LUA application, these fields show the user ID with which the application was started, the date and time when the LUA session became active, and the computer on which the application is running. If the LU is not in use by an LUA application, the message “Not in use” is displayed.

LU status

Reports the status of the LU — Active or Inactive.

Session status

If the LU status is Active, this field reports the status of the session — unbound (enabled but not logged onto a host application) or bound (logged on to a host application).

Application name

If the LU session is bound (the LU is in session with a host application), the name of the host application is displayed.

Connection

Displays the name of the connection from LINKix to the host computer which is configured for use by this LU.

Link

Displays the name of the link you are using from LINKix to the host computer. If the session is not active, this field is blank.

3. Choose < Reset > to reset the session status for a bound LUA LU. This ends the LU session, but not the SSCP session. An UNBIND message is sent to the application as if the session had been reset by the host.

Choose < Done > or press **ESC** to return to the previous dialog.

Managing RJE LUs

The Manage program provides two dialogs for RJE LU management. One box shows the status of RJE LUs configured on a particular local node; another offers detailed information, and a reset option, for a selected RJE LU.

Status of RJE LUs

The RJE LUs dialog shows information on RJE LUs configured on a particular LINKix local node. To access this information, do the following:

1. From the Resource Type dialog, select the RJE Logical Units option.
2. Choose < Zoom >. This displays the RJE LUs dialog.

Manage F1=Help

Manage Services
RJE LUs on NODE : <SNA services component>

LU Status: (o) All () Inuse () SSCP () In sess () Available

Name	Description	LU #	Workst'n	Started	
RJE03	RJE LUs on SDLC1	3	WKSTA1	05-23 16:14	^
RJE04	RJE LUs on SDLC1	4	WKSTA1	05-23 16:14	█
RJE05	RJE LUs on SDLC1	5	WKSTA1	05-23 16:14	#
RJELU16	RJE LUs on X25ONE	16	Not in use		#
RJELU17	RJE LUs on X25ONE	17	Not in use		#
					#
					#
					#
					v

< Zoom > < Done >

Copyright (C) 1993 CLEO Communications

By default, this dialog shows the status of all RJE LUs on the node.

Using Manage

3. Use the LU Status radio group to select which LUs to view.

LU Status

This radio group allows you to select which LUs to view:

All	All configured RJE LUs are displayed.
In use	All LUs which are being used by an active RJE workstation (whether or not a connection to the remote system has been established) are displayed.
SSCP	All LUs which are communicating with SSCP (the host's network control program) are displayed.
In Sess	All LUs which are logged on to the host's Job Entry Subsystem are displayed.
Available	All LUs which are not being used by an active RJE workstation (that is, LUs owned by workstations which are not currently running, or LUs not owned by any workstation) are displayed.

4. Select an LU and choose < Zoom > for more information and options on the LU, or choose < Done > to cancel the operation and return to the previous dialog.

The following information is displayed in the RJE LUs dialog:

Name

Displays the name of the RJE LU.

Description

Displays the description of the RJE LU.

LU

Displays the LU number.

Note: The RJE LU name, description, and LU number are taken from the configuration file. See **Configuring for RJE**, in the *Administration Guide*, for more information.

Workst'n

If the LU is owned by an active RJE workstation, this field displays the name of the RJE workstation using this LU. If the workstation owning the LU is not active, or if the LU is not owned by a workstation, "Not in use" is displayed. The information displayed on the screen is refreshed every 10 to 15 seconds; when a workstation is started or stopped, the dialog reflects the change.

Started

If the LU is being used by its owning RJE workstation, this field shows the date and time the session was established with the host computer.

Status of a Selected RJE LU

To view detailed information on an individual RJE LU, or to reset the session status for a bound RJE LU, do the following:

1. Select the RJE LU from the dialog.
2. Choose < Zoom >. This displays the RJE LU dialog.

Manage F1=Help

Manage Services

RJE LUs on NODE : <SNA services component>

RJE LU RJE04 : RJE LUs on SDLC1

LU type	RJE
LU number	04
Workstation name	TST1
Workstation started at . . .	05-23 16:14
Computer name	newserve
LU status	Active
Session status	Bound
Host system	JES2
Connection	SDLC1
Link	SDCLINK

< Reset > < Done >

Using Manage

Note that the title of the dialog contains the name and description of the selected LU. The information found in this dialog comes from the configuration file or from the status of the host session.

LU type

Shows the type of LU — RJE.

LU number

Displays the number assigned to the LU in the configuration file.

Workstation name, Workstation started at, Computer name

If the LU is in use by an active RJE workstation, these fields show the name of the workstation, the date and time when the workstation was started, and the computer on which the workstation is running. If the LU is assigned to an RJE workstation but the workstation is not running, the message “LU is assigned to an RJE workstation” is displayed; if the LU is not assigned to any RJE workstation, “Not in use” is displayed.

LU status

Reports the status of the LU — Active or Inactive.

Session status

Reports the status of the host session — Bound or Unbound. This field is displayed only if the LU status is Active.

Host system

If the session is bound to the host Job Entry Subsystem, this field shows the type of Job Entry Subsystem in use; this is taken from the configuration of the RJE workstation using the LU. Otherwise, this field is not displayed.

Connection

Displays the name of the connection to the host computer.

Link

Displays the name of the link you are using to the host computer. This field is blank if the LU is inactive.

3. Choose < Reset > to reset the session status for the LU, or choose < Done > to cancel the operation and return to the previous dialog.

Managing Connections

The Manage program lets you check on the status of all connections owned by a particular local node, and activate or deactivate the ones you choose. To access these management options:

1. Select the Connections option from the Select Resource Type for Zoom dialog.
2. Choose < Zoom >. This displays the Connections dialog.

Manage Manage Services F1=Help

Connections from NODE : <SNA services component>

Connection Status: (o) All () Active () Inactive

Name	Connection description	Status	Link	Type
QLLC1	Vendor X.25 host conn	Pending	SX25	X.25

< Activate > < Deactivate > < Done >

Copyright (C) 1993 CLEO Communications

3. Use the Connection Status radio group to specify which connections are to be displayed.

Connection Status

This radio group allows you to select which connections to view:

All	All configured connections owned by the local node are displayed.
Active	Only active connections are displayed.
Inactive	Only inactive connections are displayed.

Using Manage

- 4 . Review the status information on connections.

Name

Displays the name of the connection.

Connection description

Shows the description of the connection.

status

Shows whether the connection is active, starting, stopping, or inactive, or if a link outage has occurred.

Link

Identifies the link being used by an active connection. Since a connection may have more than one link configured, this field shows the one the connection is using. This information only is displayed if the connection is active.

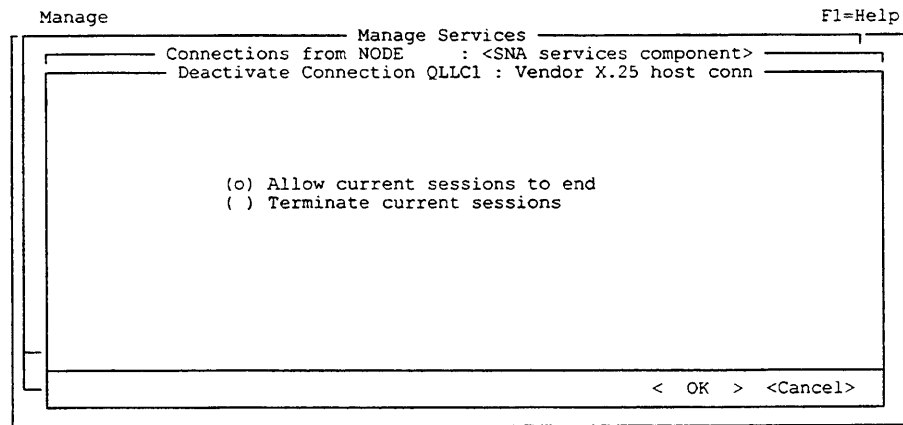
Type

Identifies the type of connection. The connection type can be SDLC, X.25, VX25 (Vendor X.25), or TR (Token Ring).

Note: The name, connection description, link, and type are taken from the configuration file and cannot be changed here. For more information, see **Configuring Services**, in the *Administration Guide*.

5. Choose < Activate > to activate a selected inactive connection. The program displays an acknowledgment message; choose < Exit Message > to respond to this message and activate the connection. This pushbutton is disabled (grayed) if the connection is active, or if it is starting or stopping.

Choose < Deactivate> to deactivate a selected active connection. The Deactivate Connection dialog is displayed when you select this option.



It offers two options for implementing this command. This pushbutton is disabled (grayed) if the connection is inactive, or if it is starting or stopping.

If you choose to let current sessions end, the program postpones deactivation until all active sessions finish. The other option immediately terminates the connection and all active sessions. 3270 sessions using this connection will display the Syndicator and a communications check code on the status line; APPC, CPI-C, or LUA applications using the connection will receive an error return code indicating the session failure.

Choose < OK > to deactivate the connection, or abandon the operation by choosing the < Cancel > pushbutton or pressing **ESC**.

Note that if you activate or deactivate a connection which is configured to be activated "on demand" (that is, it is activated automatically when required by a session and deactivated automatically when no longer required), the connection is now effectively an "operator started" connection and must be started and stopped using Manage or NetView commands. This change remains in effect until the LINKix software is restarted.

- 6 . Choose < Done > to return to the previous dialog.

Viewing and Changing Audit Log and Trace Settings for a Local Node

The Manage program lets you review or change the audit logging level set in the configuration file, and control message tracing, for a particular local node. These changes apply to the selected local node and to all its resources (such as connections and 3270 LUs), but not to links or to any other local node. Therefore, by changing these diagnostic settings, you can selectively track activity.

To view or change these diagnostic settings, do the following:

1. Select the Audit Log and Trace option from the Select Resource Type for Zoom dialog.
2. Choose < Zoom >. This displays the Audit Log/Trace dialog.

The screenshot shows a window titled "Manage" with a menu bar containing "Manage Services" and "F1=Help". The main area is titled "Audit log/trace for NODE : <SNA services component>". It displays the following information:

```
Type . . . . . Local node
Computer . . . . . newserve
Started . . . . . 05-22 16:30
Error log . . . . /usr/lib/linkix/com.err
Audit log . . . . /usr/lib/linkix/com.aud

Message tracing level:
( ) No message tracing
(o) Internal message tracing
( ) SNA messages only

Audit logging level:
(o) Detailed problem analysis data (6)
( ) General information messages (8)
( ) Significant system events (10)
( ) Audit logging disabled
```

At the bottom right of the dialog are buttons "< OK >" and "<Cancel>". At the very bottom of the window is the text "Copyright (C) 1993 CLEO Communications".

The following paragraphs describe the Audit Log/Trace dialog:

Type

Identifies the type of service — local node.

Computer

Identifies the server on which the local node is running.

Started

Gives the date and time at which the local node was started.

Error log

Displays the error log file name and its directory.

Audit log

Displays the audit log file name and its directory.

Message tracing level

Allows you to control tracing on the local node. Your choices areas follows:

No message tracing

Tracing is disabled for the local node.

Internal message tracing

Traces all messages, both SNA data and LINKix internal messages.

SNA messages only

Traces SNA messages only, no internal control messages.

The default is no message tracing (unless you started the local node with tracing). Tracing will degrade performance, so you should not enable it unless you need it to track a specific problem.

Note: If you change the tracing options, the trace file will reflect the change as soon as you choose < OK > to exit the dialog.

Audit logging level

Choose the level of information you want recorded in the audit log file for the local node. See the *Administration Guide* for explanations and examples of the different audit log levels.

Detailed problem analysis data (6)

This option causes all messages at levels 6, 8, and 10 to be logged.

Using Manage

General information messages (8)

This option causes all messages at levels 8 and 10 to be logged.

Significant system events (10)

This option causes all messages at level 10 only to be logged.

Audit logging disabled

This option suppresses all level 6-10 audit log messages. Level 6 messages provides very detailed information which is useful for problem tracing. Level 8 provides general information messages of system events. Level 10 provides only significant system events.

Note: If you change the audit log level for the local node, the trace file reflects the change after you choose < OK > to exit this dialog.

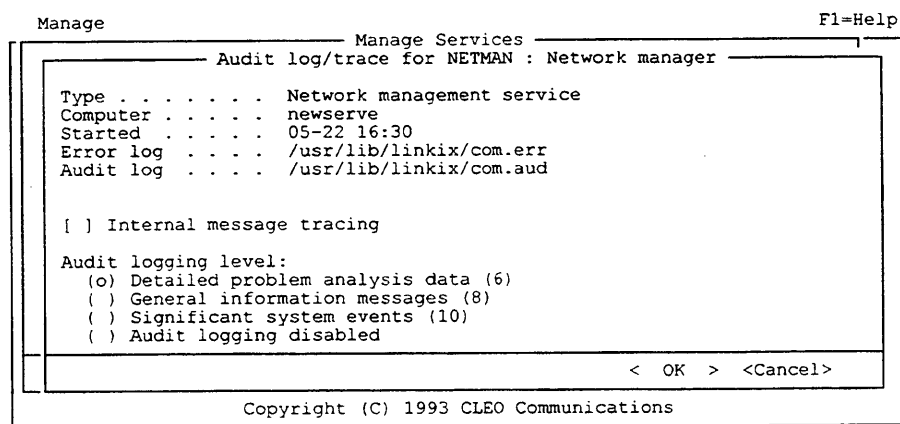
Choose < OK > to implement any changes you have made to these settings. When an acknowledgment message is displayed, choose < Exit Message > to respond and implement the changes.

Choose < Cancel > or press **ESC** to abandon this operation and return to the previous dialog.

Managing Log and Trace Settings for the Network Management Daemon;

To view or alter diagnostic settings for the network management daemon program, follow these steps:

1. Select NETMAN from the Manage Services dialog.
2. Choose < Zoom > to display the Audit Log/Trace for NETMAN dialog.



3. **Review** the information and make any necessary changes to the diagnostic settings.

Type
Network management service.

Computer
Shows the computer on which the service is running.

Using Manage

Started

Shows the date and time the service started.

Error log

Displays the current error log file name and directory.

Audit log

Displays the current audit log file name and directory.

Note: The log file names and paths are set up in configuration and cannot be changed here. These files contain error and audit messages. To configure these log files, refer to the *Administration Guide*.

Internal message tracing

Enables tracing of all messages to and from the network management daemon program and the UNIX Command Facility (UCF) daemon. This is used for diagnostic purposes. Tracing is disabled by default (unless you started the network management daemon using the “Start with Trace” push button); it will degrade performance, and should not be used unless specifically required to solve a problem.

Audit logging level

Choose the level of information you want recorded in the audit log file for the network management daemon program. LINKix audit log levels are as follows:

Detailed problem analysis data (6)

This option causes all messages at levels 6, 8, and 10 to be logged.

General information messages (8)

This option causes all messages at levels 8 and 10 to be logged.

Significant system events (10)

This option causes all messages at level 10 only to be logged.

Audit logging disabled

This option causes LINKix to suppress all level 6-10 audit log messages. Level 6 messages provides very detailed information which is useful for problem tracing. Level 8 provides general information messages of system events. Level 10 provides only significant system events.

- 4 . Choose < OK > to implement changes in the audit log levels. When the program responds with an acknowledgment message, choose < Exit Message > to continue and return to the Manage Services dialog.

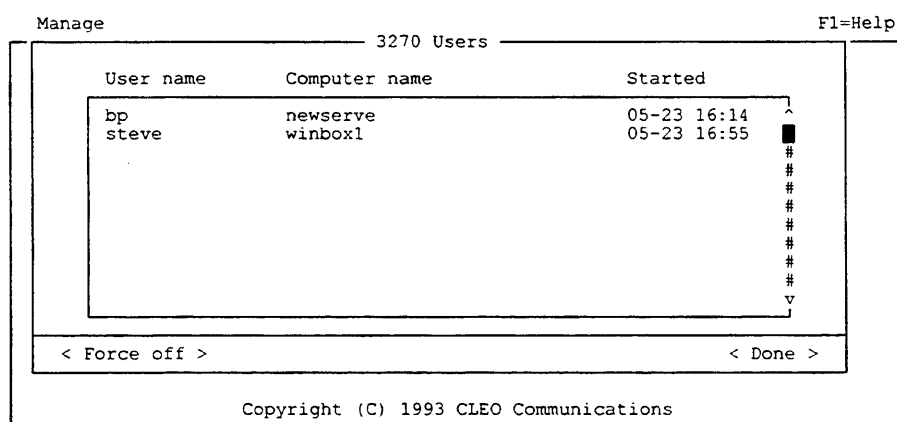
Choose < Cancel > to retain the existing settings and return to the Manage Services dialog.

Managing 3270 Users

The Manage program displays a list of all active 3270 users, and also provides a means to force users off.

To view the status of all users or force off a selected user:

1. Select the 3270 Users option from the Manage menu.
2. Review the status information.



The following paragraphs describe the 3270 Users dialog.

User name

The name of each active 3270 user.

Computer name

The computer on which each user is running the 3270 emulation program.

Started

The date and time at which each user started the 3270 emulation program.

3. Choose < Force Off > to force off a selected user, or choose < OK > to return to the previous dialog.

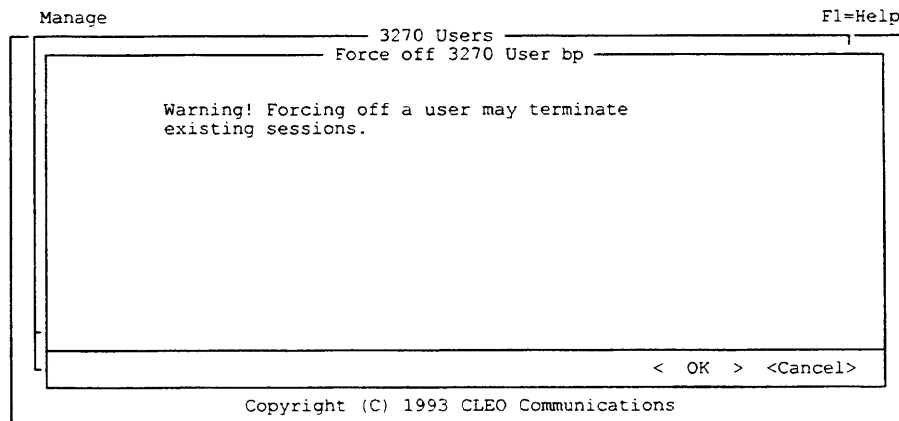
Forcing Off a 3270 User

You can force an active 3270 user off the LINKix computer; this may be useful (for example) if a user's session has become deadlocked due to a problem with the host, or if a user has left the 3270 emulation program running but is not using it. Forcing off a user will cause the 3270 emulation program to exit, and will terminate any sessions it is using.

To force off a user:

1. Select the user from the 3270 Users dialog.
2. Choose < Force Off >.

A message is displayed similar to the one shown below.



3. Choose < OK > to force off the user. When the program responds with an acknowledgment message, choose < Exit Message > to continue and return to the previous dialog. The 3270 emulation program terminates, returning the user to the shell.

Using Manage

Choose < Cancel > or press **ESC** to abandon the operation and return to the previous dialog.

Managing APPC and CPI-C Applications

The Manage program displays a list of all active APPC and CPI-C applications (both invoking and invoked), and also provides a means to terminate them.

To view a status report of all APPC and CPI-C applications, or to force off a selected application:

1. Select the APPC and CPI-C Applications option from the Manage menu. This displays the dialog shown below.
2. Review the status information.

Manage
APPC and CPI-C Applications
F1=Help

Name	Description	Computer name	Started
RANDTPP		newserve	05-23 12:57
TESTCPIC	CPI-C test prog	testbox1	05-23 13:04

< Force off >
< Done >

The following paragraphs describe the fields of the APPC and CPI-C Applications dialog.

Name _____

Displays the TP name of each application. If a TP supplied a blank or null TP name, the string displayed here is “pidnnnnn”, where *nnnnn* is the UNIX process ID of the TP.

Using Manage

Description

For an invoked TP, this field displays the description of the TP given in the configuration file; otherwise it is blank.

Computer name

Displays the computer on which each application is running.

Started

Displays the date and time each application was started.

3. Choose < Force Off > to deactivate a selected application, or choose < OK > to return to the previous dialog.

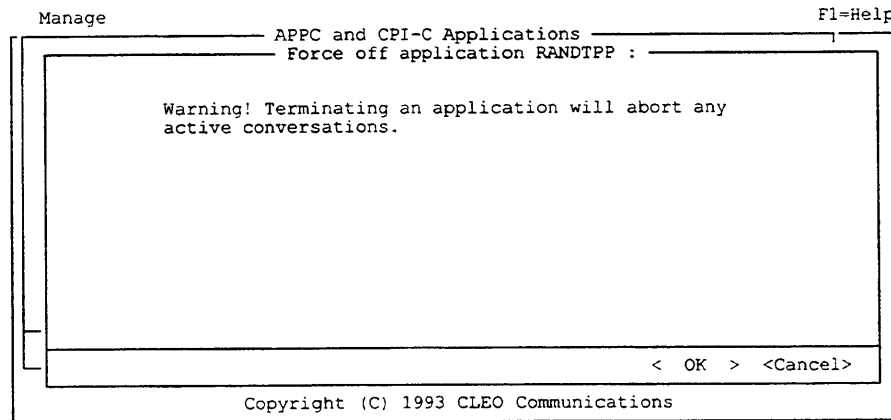
Forcing Off an APPC or CPI-C Application

You can force termination of an active APPC or CPI-C application; this may be useful (for example) if the application has become deadlocked or is looping, or if you believe an unauthorized user is accessing it. Forcing off an application causes it to exit.

To force off an APPC or CPI-C application:

1. Select the application from the list box in the APPC and CPI-C Applications dialog.
2. Choose Force Off.

A message, similar to the one shown below, is displayed when you select this option.



3. Choose < OK > to force off the application. When the program responds with an acknowledgment message, choose < ExitMessage > to continue and return to the previous dialog. Any partner application which had an active conversation with this application will receive an error return code , indicating that its partner ended abnormally.

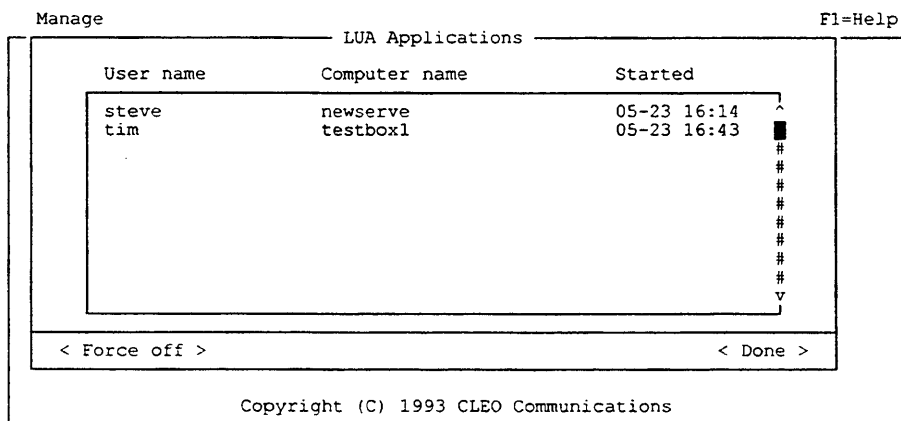
Choose < Cancel > or press **ESC** to abandon the operation and return to the previous dialog.

Managing LUA Applications

The Manage program displays a list of all active LUA applications, and also provides a means to terminate applications.

To view a status report of all applications or force off a selected application:

1. Select the LUA Applications option from the Manage menu. Note that, if an application consists of more than one UNIX process issuing LUA verbs, it is listed once for each process.



2. Review the status information in the LUA Applications dialog.

User name

Displays the user name with which each LUA application was started.

Computer name

Displays the computer on which each application is running.

Started

Displays the date and time each application was started.

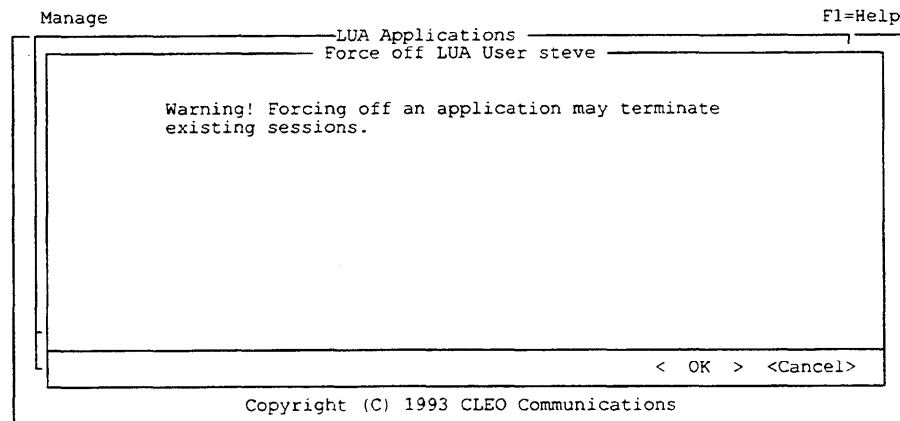
3. Choose < Force Off > to deactivate a selected application, or choose < Done > to return to the previous dialog.

Forcing Off an LUA Application

You can force termination of an active LUA application; this maybe useful (for example) if the application has become deadlocked or is looping. Forcing off an LUA application causes the application to exit.

To force off an LUA application:

1. Select the application from the LUA Applications dialog.
2. Choose < Force Off >. A message, similar to the one shown below, is displayed.



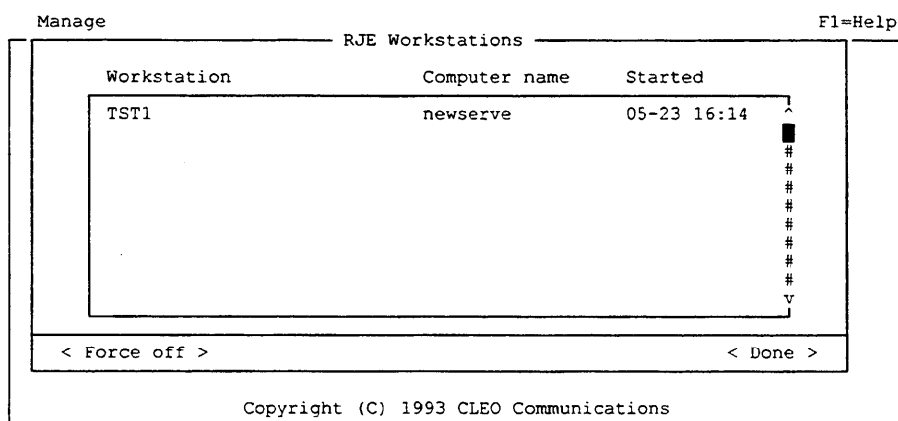
3. Choose < OK > to force off the application. When the program responds with an acknowledgment message, choose < Exit Message > to continue and return to the previous dialog.

Choose < Cancel > or press **ESC** to abandon the operation and return to the previous dialog.

Managing RJE Workstations

The Manage program displays a list of all active RJE workstations, and also provides a means to force workstations to stop. To view the status of all workstations or force off a selected workstation:

1. Select the RJE Workstations option from the Manage menu.



2. Review the status information in the RJE Workstations dialog.

Workstation

Displays the name of each active RJE workstation.

Computer name

Displays the computer on which each workstation is running.

Started

Displays the date and time at which each RJE workstation was started.

3. Choose < Force Off > to force off a selected workstation, or choose < Done > to return to the previous dialog.

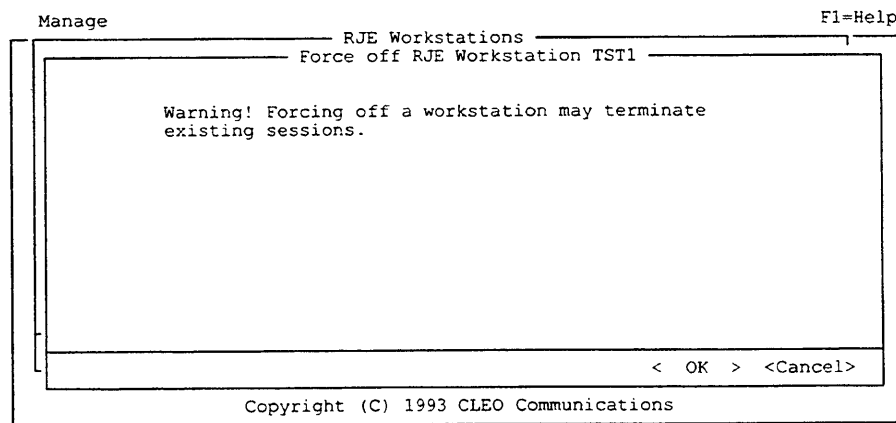
Forcing Off an RJE Workstation

You can force off an active RJE workstation; this may be useful (for example) if a user's session has become deadlocked due to a problem with the host, or if a user has left the workstation running but is not using it. Forcing off a workstation will cause the workstation daemon program to exit, and will terminate any sessions it is using.

To force off a workstation:

1. Select the workstation from the RJE Workstations dialog.
2. Choose < Force Off >.

A message, similar to the one shown below, is displayed.



3. Choose < OK > to force off the workstation. When the program responds with an acknowledgment message, choose < Exit Message > to continue and return to the previous dialog. The RJE workstation terminates.

Choose < Cancel > or press **ESC** to abandon the operation and return to the previous dialog.

3. Using Browse

This chapter discusses the Browse program. It contains the following information:

- ◆ How to use the program
- ◆ How to review audit and error log files
- ◆ How to review trace files
- ◆ How to copy and reset diagnostic files
- ◆ How to print reports from diagnostic files

For explanations of audit and error log messages, see the *Diagnostics Guide*; for explanations of trace file formats, see chapter 13 of the *Administration Guide*.

Starting the Program

The Browse program allows you to review and manage the LINKix log and trace files. With Browse, you can display or print complete files or extract file information selectively. In general, the Browse program does not change the contents of log and trace files. The only exception to this is the Reset Log/Trace File option, which allows you to clear the contents of log and trace files and optionally copy the information to another file before resetting.

You can run the Browse program before the LINKix control daemon is started; if it is not started, however, you will not be able to use the Reset Log/Trace File option.

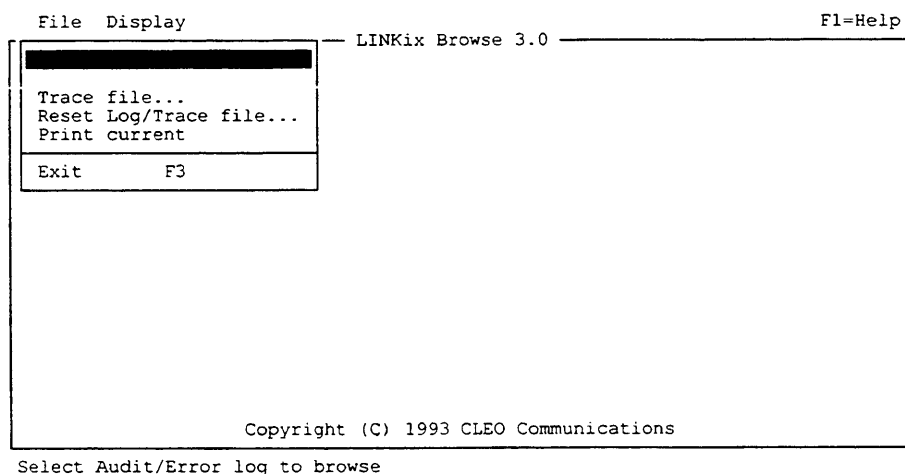
To start the Browse program, enter the following command:

```
combrowse
```

Alternatively, to start the Motif version of the program:

```
xcombrowse
```

The main screen of the program is displayed.



When the Browse program is first started, the File menu is selected automatically.

To access help information, you can press **F1** from any menu or dialog.

The File menu

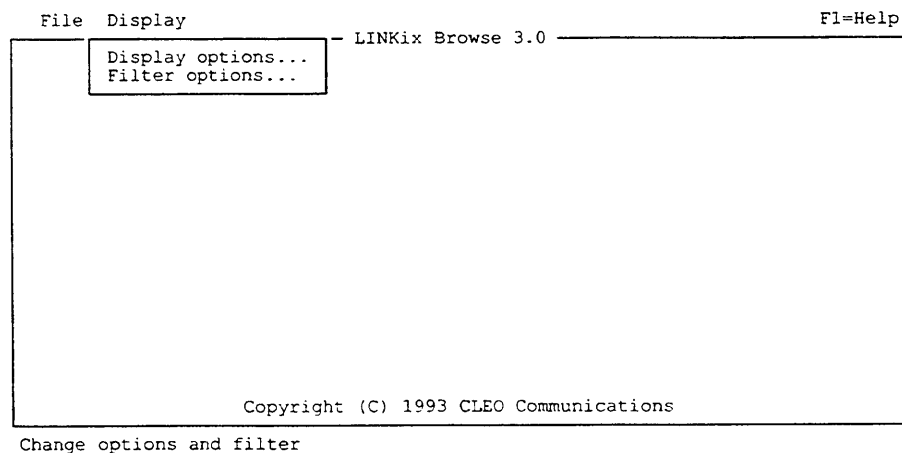
The File menu contains options that allow you to open either a log file or a trace file.

You must open a file before any other features of the Browse program can be used, since all the other options act on the currently open file. When you have opened a file, you can print it or customize it with the Display options.

The File menu also includes the options: Reset Log/Trace File, Print Current and Exit. The Reset Log/Trace File option clears the contents of a diagnostic file (with the option of copying the information to another file if you need to keep it). The Print Current option prints a currently opened diagnostic file. The Exit option quits the Browse program and returns to the UNIX command prompt.

The Display Menu

The Display menu contains options that let you customize the screen display and printout for an open log or trace file by changing the appearance of the report or by filtering the information it includes.



Viewing Audit and Error Log Files

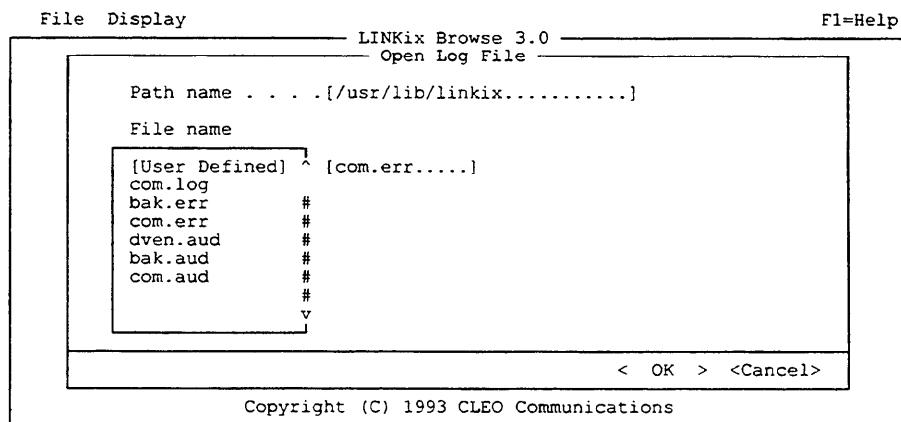
This section explains how to use the Browse program to view audit and error log files. It includes the following information:

- ◆ How to open and display the file
- ◆ How to select information from the file to be displayed or printed

Opening a Log File

To open and view a log file:

1. Select Log File from the File menu. This displays the Open Log File dialog.



The Open Log File dialog contains the following fields.

Path Name

The Path Name edit box contains the path name for the active error log file. If you specify a different path, the File name list box is updated to include all

appropriate files in the directory you have just specified. This updating of the File name list box occurs when you tab out of the Path Name edit field.

File Name

The File Name tracking box contains the name of the active error log file, selected from the list box of all available audit and error log files (all files which have the extension .aud, error log) in the directory specified by the path name.

2. If the file you want to open is in a different directory, type the file's path name in the Path name edit box.
3. Select one of the files in the File name list box, or select [User Defined] and enter a file name in the File name field.
4. Choose < OK > to open and view the log file.
Choose < Cancel > to abandon the operation and return to the main screen.

Displaying a Log File

When you open a log file, the file contents appear on the screen.

```

File Display                               Fl=Help
View Log File : /usr/lib/linkix/com.aud
<--- Start of File --->
=====
===== New run at Mon Mar 14 17:41:19 1994 =====
=====
SNA0445: 03/14 17:41:28 map          dven      3270   10
          3270 emulation started
SNA0446: 03/14 17:41:36 map          dven      3270   10
          3270 emulation finished
COM1591: 03/14 17:41:41 map          ROUTER      10
          Router Completely Terminated
=====
===== New run at Mon Mar 14 17:43:17 1994 =====
=====
SNA2160: 03/14 17:43:22 map          START      ADMIN  10
          Request to start service SDLCNODE locally
COM0039: 03/14 17:43:22 map          SDLCNODE  NODE   10
          Connection SDLCCONN failed - no link service available
SNA2158: 03/14 17:43:49 map          STOP       ADMIN  10
=====
Press Return to select a menu

```

Using Browse

The Browse program displays information that is stored in the log file. This file remains open until you open another file or exit the Browse program. To examine the contents of the displayed log file, use the **UP** and **DOWN** cursor keys, or the **PAGEUP**, **PAGEDOWN**, **HOME** and **END** keys.

The example above shows an uncustomized log file that contains all the entries in the file. Each message in this open log file contains several items of header information. This header information includes the following details:

- ◆ Message number
- ◆ Date and time the message was logged
- ◆ The service or user that logged the message — for example the name of a local node or link, or the name of a 3270 emulation program user
- ◆ Computer on which the service or user was running
- ◆ Message severity level

In addition to the header information, the file also displays the text associated with each error message number. These messages and their meanings are documented in the *LINKix Diagnostics Guide*.

Customizing a Log File

Sometimes you may not need to view all the header information in the file. For example, if you need to generate customized log reports for errors of a certain severity level. You can tailor the log report to show errors at that severity level and those that are more severe. When you specify a severity level (6 is the lowest level, 16 is the highest), the Browse program shows all errors with a severity level that is equal to or greater than the specified level.

You can also customize the screen display to show only certain types of log entries (for example, 3270 entries) and exclude all others. Customizing diagnostic reports lets you efficiently review and use the log file information.

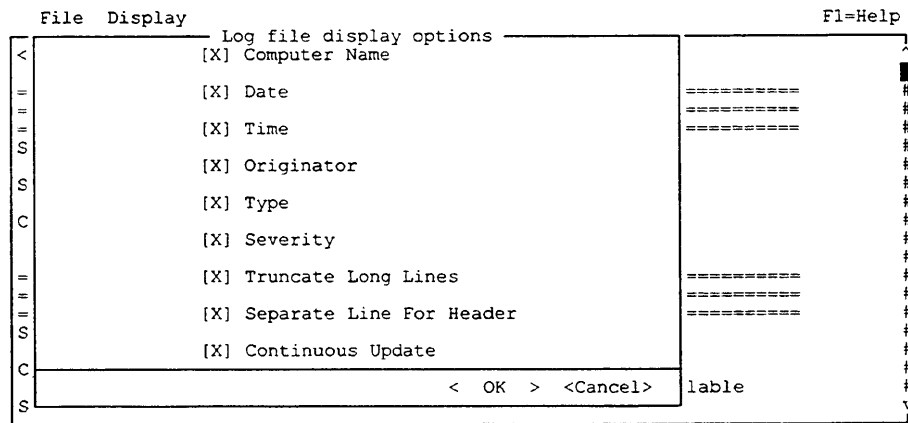
To customize diagnostic log reports or to become more familiar with the information in the log files, choose the Display menu from the menu bar. This provides the following options:

Display options — to select the details that appear in each log file line

Falter options — to select which log entries appear in the log file

Selecting Log File Display Options

Display options control the type of information that appears in each log file line. To access these options, select Display options from the Display menu. The Log file display options dialog is displayed.



Note that all the options in this dialog are selected by default. However, you can control output for an opened log file by changing the display options that are selected. The following paragraphs explain what each option in the Log File Display Options dialog is for.

Computer Name

Displays the name of the computer on which the component that logged the message was running.

Date

Displays the date when the message was logged.

Time

Displays the time when the message was logged.

Using Browse

Originator

Displays the name of the LINKix component or user that logged the message. This can be, for example, the name of a local node or link, CONFIG for the Configuration program or the name of a 3270 user whose copy of the 3270 emulation program logged the message.

Type

Displays the type of service that logged the message — for example, NODE, SDLC, ADMIN or 3270.

Severity

Displays the severity level of the message. LINKix defines audit messages as being at severity levels 6, 8 or 10 and error messages as being at severity levels 12 or 16.

Truncate Long Lines

Truncates entries longer than 78 characters at the edge of the display area. If you do not select this option, long lines wrap to the next line of the display.

Separate Line For Header

Displays the header information for each file entry on a separate line from the text message. The header information contains the message number, date, time, computer name, originator, type and severity level. If you do not select this option, the log entry appears on one line (wrapped or truncated at 78 characters, depending on the Truncate Long Lines option setting).

Continuous Update

Displays new entries being added to the log file as soon as they are available. You will only see these new entries if you are at the end of the file. If you are viewing some other part of the file, you will not see these entries as they are added. If you do not select this option, you can only update the display by using the **DOWN**, **PAGEDOWN** or **END** keys to see the latest entry.

Selecting Log File Filter Options

Filter options determine which log file entries are selected for log file displays and printouts. They allow you to include or exclude information from diagnostic reports. To access these options, select Filter options from the Display menu. The Log file filter options dialog is displayed.

Fi Log file filter options Help

<---	Originator:	Computer name:	
	(o) Include	(o) Include	
	() Exclude	() Exclude	
===	[.....]	[.....]	
===			
===	String match:	Type:	
SNA0	(o) Include	(o) Include	
	() Exclude	() Exclude	
SNA0	[.....]		ALL #
			NODE #
SNA0	Lowest severity:		3270 #
	(o) Detailed problem analysis data (6)		RJE #
SNA0	() General information messages (8)		SDLC #
	() Significant system events (10)		VX25 #
	() Warnings/recoverable errors (12)		X.25 v
	() Fatal errors (16)		
SNA0			
SNA0			

< OK > <Cancel>

Press Return to select a menu

You can control the display of an opened log file by selecting or not selecting the filter options. The following paragraphs explain what each option in the Log File Filter Options dialog performs.

Originator

Performs a match on the originator name, as follows:

- ◆ Specify Include and leave the entry field empty to include all entries.
- ◆ Specify Include and an originator name to include messages from only one originator.
- ◆ Specify Exclude and an originator name to ignore messages from a specific originator.

String match

Specifies a search string to use for filtering entries, as follows:

- ◆ Specify Include and leave the entry field empty to include all entries.
- ◆ Specify Include and a text string to include only messages which contain that string.
- ◆ Specify Exclude and a text string to ignore messages which include that string.

The string match can be used on strings of text in the messages or the message number. For example, if you specify Exclude and the string "SNAO023," all

Using Browse

instances of message number 23 will be ignored; if you specify Include and the string "configuration," only messages concerned with configuration problems will be included.

Computer name

Specifies a computer name to include or exclude, as follows:

- ◆ Specify Include and leave the entry field empty to include all entries.
- ◆ Specify Include and a computer name to include only messages which contain that computer name.
- ◆ Specify Exclude and a computer name to ignore messages which include that computer name.

Type

Specifies a service type to include or exclude from the log file display. Type refers to such entries as NODE, SDLC, X.25, VX25 (Vendor X.25), TOKR ('Token Ring upper component), LLC2 (Token Ring lower component), 3270, TP and ADMIN.

- ◆ Specify Include and leave the entry field empty to include all entries.
- ◆ Specify Include and a service type to include only messages which contain that service type.
- ◆ Specify Exclude and a service type to ignore messages which include that service type.

For example, if you specify Exclude and the type 3270, none of the messages logged by the 3270 emulation program will be displayed.

Lowest Severity

Specifies the lowest severity level to display, as follows:

- 6 Detailed audit information, for example the initialization sequence for an SDLC connection or log data supplied by an APPC TP.
- 8 User and data link audit level, covering such events as user logon, APPC TP program start/end, and connection establishment.
- 10 Significant system events. These include events such as starting a network service, or successfully updating a configuration file. This is the default audit level set during configuration.

12 Warnings/recoverable errors. Level 12 errors will adversely affect the operation of a LINKix component, but will not cause its termination.

16 Fatal errors. A level 16 error will cause termination of a LINKix component.

Choose level 6 to display all messages.

After you have changed the settings, choosing < OK > will reformat the display according to the options you selected. Choosing < Cancel > abandons the operation and returns you to the log file.

Viewing Trace Files

This section explains how to use the Browse program to view trace files. It includes the following information:

- ◆ How to open and display the file
- ◆ How to select information from the file to be displayed or printed

For information on interpreting the contents of trace files, see chapter 4, LINKix Trace Diagnostics.

Opening a Trace File

To open and view a trace file:

1. Select Trace file from the File menu. The Open Trace File dialog is displayed.

File Display View Log File : /usr/lib/linkix/com.aud Fl=Help

<--- Open Trace File

==== Path name/usr/lib/linkix.....]

==== File name

SNA1 [User Defined] ^ [comfile1.trc]

==== comfile1.trc

==== comfile2.trc

====

SNA1 #

SNA2 #

SNA2 #

SNA0 #

SNA2 #

SNA0 v

SNA0 lp

SNA0 lp

==== < OK > <Cancel>

==== New run at Thu Mar 7 09:50:59 1994 =====

The Open Trace File dialog contains the following fields.

Path name

The path name for the last opened trace file. If you specify a different path, the File name list box is updated to show all the trace files in the directory you have just specified. This updating of the File name list box occurs when you tab out of the Path Name edit field.

File name

The File name edit box contains the name of the trace file last opened. The list box contains the names of all trace files (files with extension .trc) in the directory specified by the Path name.

2. If the file you want to open is in a different directory, type in the file's path name.
3. To open the trace file, select the file in the list box or select [User Defined] and enter a different file name in the File name edit box.
4. Choose < OK > to open and view the trace file.
Choose < Cancel > to abandon the operation and return to the main screen.

Displaying a Trace File

When you open a trace file, the file contents appear on the screen.

```

File Display                               Fl=Help
View Trace File : /usr/lib/linkix/comfile1.trc
<--- Start of File --->
--- Initialized at 12:44:14 on 03/22/94 ---
105.49 FMI ----- 12:44:13.00
105.49 FMI 00120001->00000000 OPEN  SSCP REQUEST
105.49 FMI FMI
105.49 FMI
105.49 FMI ----- Header at address 433508, 1 elements -----
105.49 FMI 00000101 02000001 00000000 00000000 <.....>
105.49 FMI 0000 <..>
105.49 FMI
105.49 FMI ----- Element at address 433f00, start 1, end 34 -----
105.49 FMI 00000000 00000000 0000434F 4C504F4F <.....COLPOO>
105.49 FMI 4C200000 01010101 01000101 01010101 <L .....>
105.49 FMI 0101 <..>
105.49 FMI ----- 12:44:13.00
105.49 FMI 00120001->02000000 OPEN  SSCP REQUEST
105.49 FMI FMI
105.49 FMI
105.49 FMI ----- Header at address 435830, 1 elements -----
105.49 FMI 00000101 02000001 00000000 00000000 <.....>
v
Press Return to select a menu

```

Using Browse

To examine the contents of the trace file, use the **UP** and **DOWN** cursor keys, or the **PAGEUP**, **PAGEDOWN**, **HOME**, and **END** keys.

The example above shows an uncustomized trace file that contains all the entries in the file. Each message in this open log file includes the process ID, the trace type, a text message and the time the message was logged. For tracing on user-space LINKix components, the UNIX process ID is shown in two parts: the first three digits are shown as “process ID,” and the last two digits are shown as “thread ID.” For tracing on kernel-space components and the network management daemon, which have no UNIX process ID, the process ID and thread ID are as follows:

KER.PU	indicates tracing on a local node
KER.SD	indicates tracing on SDLC links; this is replaced by SDL.nn when tracing messages specific to a particular link, where <i>nn</i> is an identifier for each link which is assigned when the link is activated
KER.X2	indicates tracing on X.25 links; this is replaced by X25.nn for a particular link (as for SDLC above)
KER.QL	indicates tracing on Vendor X.25 (QLLC) links; this is replaced by QLL.nn for a particular link (as for SDLC above)
KER.TR	indicates tracing on Token Ring links; this is replaced by TKR.nn for a particular link (as for SDLC above)
USR.NM	indicates tracing on the network management daemon.

Customizing a Trace File

Sometimes you may not need to view all the trace information in the file. For example, assume that you need to generate customized reports for certain process IDs. You can tailor the printouts to show only those particular IDs. Customizing diagnostic reports lets you efficiently review and use the file information.

To customize diagnostic trace reports or become familiar with the information in the trace files, choose the Display menu. This provides the following options:

- ◆ Display options - to select the details that appear in each trace file line
- ◆ Filter options -to select which trace entries appear in the trace file

Selecting Trace File Display Options

To access the trace file display options, select Display options from the Display menu. The Trace File Display Options dialog is displayed.

File Display		Trace file display options		F1=Help
<--- Start		[X] Process ID	2:44:13.00	^
105.49 FMI		[X] Thread ID		#
105.49 FMI		[X] Trace type		#
105.49 FMI		[X] Continuous update	>	#
105.49 FMI		< OK > <Cancel>		#
105.49 FMI	00000000 00000000 0000434F 4C504F4F	<.....COLPOO>		#
105.49 FMI	4C200000 01010101 01000101 01010101	<L		#
105.49 FMI	0101	<..>		#
105.49 FMI	----- 12:44:13.00			#
105.49 FMI	00120001->02000000 OPEN SSCP REQUEST			#
105.49 FMI	FMI			#
105.49 FMI	---- Header at address 435830, 1 elements ----			#
105.49 FMI	00000101 02000001 00000000 00000000	<.....>		v

You can control output for an opened trace file by changing which display options are selected. The following paragraphs describe what each option in the Trace File Display Options dialog performs.

ProcessID

Displays the process ID, for each entry. For user-space components, Process ID indicates the first 3 digits of the UNIX process ID. For kernel components, see the explanation in **Displaying a Trace File** (p. 2-13).

Thread ID

Displays the thread ID, for each entry. For user-space components, Thread ID indicates the last two digits of the UNIX process ID. For kernel components, see the explanation in **Displaying a Trace File** (p. 2-13).

Trace Type

Displays the trace type for each entry; for example APPC, SNAP (LINKix internal control messages), SNA (SNA message tracing) or L2 (link tracing).

Using Browse

Continuous Update

Displays new entries as they are added to the trace file. You will only see these new entries if you are at the end of the file. If this option is not selected, you can only update the display by using the **DOWN**, **PAGEDOWN**, or **END** keys to see the latest entry.

Selecting Trace File Filter Options

Filter options determine which trace file entries are selected for trace file displays and printouts. They allow you to include or exclude information from diagnostic reports. To access these options, select Filter options from the Display menu. The Trace File Filter Options dialog is displayed.

File Display View Trace File : /usr/lib/linkix/comfile1.trc F1=Help

<--- S Trace file filter options

105.49 Process ID: Thread ID:

105.49 (o) Include (o) Include

105.49 () Exclude () Exclude

105.49 [.....] [.....]

105.49 Trace Type:

105.49 (o) Include

105.49 () Exclude

ALL

HLL

APPC

CPIC

CSV

LUA

< OK > <Cancel>

105.49 FMI 00000101 02000001 00000000 00000000 <.....>

The following paragraphs explain how the options in the Trace File Filter Options dialog operate.

Process ID

- ◆ If you specify Include and leave the edit box empty, all entries from all processes will be included.
- ◆ If you specify Include and a process ID, only entries from the specified process will be included.

- ◆ If you specify Exclude and a process ID, all entries from the specified process will be excluded.

Thread ID

- ◆ If you specify Include and leave the edit box empty, all entries from all processes will be included.
- ◆ If you specify Include and a thread ID, only entries from the specified thread will be included.
- ◆ If you specify Exclude and a thread ID, all entries from the specified thread will be excluded.

Trace Type

- ◆ If you specify Include and select ALL from the list box, all entries for all trace types will be included.
- ◆ If you specify Include and an entry from the list box other than ALL, the entries for that particular trace type only will be included. Type includes such entries as APPC, SNAP (LINKix internal message tracing), SNA (SNA message tracing), L2 (link tracing), DLPI (the interface between Token Ring components) and LLI (Token Ring adapter interface).
- ◆ If you specify Exclude and an entry for the list box other than ALL, the entries for that particular trace will be excluded. Type includes such entries as APPC, SNAP (LINKix internal message tracing), SNA (SNA message tracing), L2 (link tracing), DLPI (the interface between Token Ring components) and LLI (Token Ring adapter interface).

Resetting the Audit and Error Log Files

After a time, the information in log files becomes outdated. Error and audit log files also grow in size and consume disk resources.

To ensure that the files do not become too large, LINKix automatically backs up and resets a log file when it reaches one megabyte; the contents are copied to a file bak.aud (for the audit log file) or bak.err (for the error log file), and the file is then reset. A message is also logged in the error log file to warn you that this has happened.

The Browse program also provides a Reset Log/Trace File option for clearing the diagnostic file of all information. You can choose to save the diagnostic information under a different file name or discard it.

Note: You will only be able to reset log and trace files if the following are true:

- ◆ You have permission to use LINKix's system administrator functions (such as the configuration and manage programs).
- ◆ The LINKix software is started.

If either of these is not true, an error message will be displayed when you attempt to reset the file.

To reset the size of a file:

1. Select the Reset Log/Trace File option from the File menu. The Copy and Reset File dialog is displayed.

Using Browse

```
File Display      View Log File : /usr/lib/linkix/com.aud      F1=Help
<--- Start      Reset log/trace file
=====
File to be reset:
=====
Path name . . . . . [/usr/lib/linkix.....]
File name . . . . . [com.aud.....]
=====
SNA0445: 05      Copy to file before resetting:
3
SNA0167: 05      Path name . . . . . [.....]
I      File name . . . . . [.....]
SNA0165: 05      Copy to file before resetting:
E      Path name . . . . . [.....]
SNA0188: 05      File name . . . . . [.....]
W      Copy to file before resetting:
t      Path name . . . . . [.....]
b      File name . . . . . [.....]
SNA0182: 05      Copy to file before resetting:
C      Path name . . . . . [.....]
link role = 0, remote node type = 0
SNA0184: 05/30 10:19:12 SDLC01 SDLC 06
Press Return to select a menu
```

3. Specify the path and file name of the file you wish to reset.
4. To save the diagnostic information, specify a file and path to copy it to.
5. Choose < OK > to reset the file, or select < Cancel > to abandon the operation.

An acknowledgment message is displayed that tells you if the file was reset or reset and copied successfully; choose < Exit Message > to acknowledge this message and return to the main screen.

Printing Diagnostic Reports

The Browse program lets you print the contents of diagnostic files. The Print Current option is one of the File menu options; it prints the currently opened file as it appears on the screen. Printing maybe to a printer or print spooler, or to a file.

To print a file:

1. Open a log or trace file.
2. Choose any desired display options to customize the file.
3. Choose any desired filter options to customize the file.
4. Select the Print Current option from the File menu. This choice displays the Print Current File dialog.

```

File Display View Log File : /usr/lib/linkix/com.aud F1=Help
<--- Start of File --->

=====
=====
=====
SNA0445: 05/30
3270
SNA0167: 05/30
Init
SNA0165: 05/30
Esta
SNA0188: 05/30
Warn
thro
been
SNA0182: 05/30
Conn
link role = 0, remote node type = 0
SNA0184: 05/30 10:19:12 SDLC01 SDLC 06

=====
=====
=====
Print Current File
=====
=====
Print to:
(o) Printer/Program
( ) File
Name . . . [lp.....]
Path . . . [/usr/bin.....]
Parameters [.....]
[ ] Append (not overwrite) file
< OK > <Cancel>
=====
=====
=====
h device driver#
Please dial
n LINK1
r may have
F019 ,
=====
=====
=====
Press Return to select a menu

```

Using Browse

5. Complete the fields in the dialog, as follows:

Print to

Select Printer/Program to send the output to a printer, print spooler, or shell script, or File to send it to a UNIX text file. The default is Printer/Program.

Name

Specify the name of the file to hold the output, or the name of the printer or shell script to which the output is sent.

If Printer/Program is selected in “Print to” above, the default is **lp**; if File is selected, there is no default.

Path

Specify the full directory path of the file or program to which the output is sent.

If Printer is selected in “Print to” above, the default is **/usr/bin**; if no path is specified, the PATH environment variable is used to search for the program.

If File is selected in “Print to” above, there is no default path; if no path is specified, LINKix writes to the current directory.

Parameters

Specify any command-line parameters required for the printer or print spooler. This field is disabled (grayed) if you are printing to a file instead of a printer.

Append (Not Overwrite) File

If you are sending output to a file, select this check box if you want the output to be added to the end of an existing file. If you do not select this check box, the output will replace any existing data in the file. This field is disabled (grayed) if you are sending output to a printer.

6. With the options set as you want, choose < OK > to print the file or < Cancel > to cancel the print request.

Exiting the Browse Program

To end the Browse program and return to the UNIX command prompt, choose Exit from the File menu.

Alternatively, you can press **F3** from any point in the menu interface. The Browse program closes any displayed dialogs and exits to the UNIX command prompt. Note that this will abandon any action awaiting conflation (for example, printing or resetting a file).

Using Browse

4. Additional Utilities

This chapter describes additional command line utilities that allow you to:

- ◆ manage connections
- ◆ view the status of LUs

Managing Connections

You can activate and deactivate connections using the Manage program or with the *comconn* utility. This section explains how to manage connections with *comconn*; see chapter 2 of this manual for information on managing connections using the Manage program.

Viewing the Status of a Connection

To view the status of a connection, enter the following command at the UNIX command prompt:

```
comconn conname
```

where *conname* identifies which connection is being managed. This must match the name of a connection that is configured for LINKix. It is not case sensitive.

The *comconn* utility returns an exit code of zero if it completes successfully. If an error occurs, the exit code is non-zero. For explanations of the exit codes which may be returned, see the LINKix *Diagnostics Guide*.

In addition, LINKix writes a message to standard output to indicate whether the command was issued successfully. This is normally as follows:

```
Connection details are:  
  Activation state   = state  
  Link name         = linkname
```

This message indicates that the command was issued successfully. *state* is the current state of the connection: "inactive," "active," "starting," "stopping," or "outage" (a link error has occurred). *linkname* is the name of the link used by the connection; this line appears only if the connection is active.

Any other message indicates an error condition, and begins either with "Syntax error" or "Error," followed by information on the cause of the error. "Syntax error" messages indicate that you entered an invalid option or parameter, or that you did not supply a required one; check the correct syntax and retry. "Error" messages indicate that the command syntax was correct, but that the command could not be issued (for example, because a required LINKix component was not started or because an operating system error occurred). Take whatever action is appropriate and retry the command.

Activating a Connection

Before activating a connection from the command line, the control daemon must be started. See chapter 11 of the *LINKix Administration Guide* for details.

To activate a connection, enter the following command at the UMX command prompt:

```
comconn conname -a
```

where *conname* identifies which connection is being managed. This must match the name of a connection that is configured for LINKix. It is not case sensitive.

The `-a` option indicates that the connection is to be activated.

The *comconn* utility returns an exit code of zero if it completes successfully. If an error occurs, the exit code is non-zero. For explanations of the exit codes which may be returned, see the *LINKix Diagnostics Guide*.

In addition, LINKix writes a message to standard output to indicate whether the command was issued successfully. This is normally one of the following:

Activate connection request sent to Node *nodename*.

This message indicates that the command was issued successfully. *nodename* is the name of the local node that owns the connection. (Note that this message does not necessarily indicate that the connection has been activated.)

Connection is already active.

The connection cannot be activated because it is already active. No action is needed,

Connection is already in pending state.

The connection cannot be activated because it is already in the process of being activated. No action is necessary.

Any other message indicates an error condition, and begins either with “Syntax error” or “Error,” followed by information on the cause of the error. “Syntax error” messages indicate that you entered an invalid option or parameter, or that you did not supply a required one; check the correct syntax and retry. “Error” messages indicate that the command syntax was correct, but that the command could not be issued (for example, because a required LINKix component was not started or because an operating system error occurred). Take whatever action is appropriate and retry the command.

Deactivating a Connection

To deactivate a connection, enter one of the following commands at the UNIX command prompt:

```
comconn  conname  -d
comconn  conname  -f
```

where *conname* identifies which connection is being managed. This must match the name of a connection that is configured for LINKix. It is not case sensitive.

The *-d* option specifies a normal deactivation (do not deactivate the connection until all sessions using it have ended).

The *-f* option specifies a forced deactivation (deactivate immediately, without waiting for sessions to end).

The *comconn* utility returns an exit code of zero if it completes successfully. If an error occurs, the exit code is non-zero. For explanations of the exit codes which may be returned, see the *LINKix Diagnostics Guide*.

In addition, LINKix writes a message to standard output to indicate whether the command was issued successfully. This is normally one of the following:

Deactivate connection request sent to Node *nodename*.

This message indicates that the command was issued successfully. *nodename* is the name of the local node that owns the connection. (Note that this message does not necessarily indicate that the connection has been deactivated. In particular, if you specified a normal deactivation, it may take some time for all sessions to end before the connection can be deactivated.)

Connection is already inactive.

The connection cannot be deactivated because it is not active. No action is needed.

Any other message indicates an error condition, and begins either with “Syntax error” or “Error,” followed by information on the cause of the error. “Syntax error” messages indicate that you entered an invalid option or parameter, or that you did not supply a required one; check the correct syntax and retry. “Error” messages indicate that the command syntax was correct, but that the command could not be issued (for example, because a required LINKix component was not started or because an operating system error occurred). Take whatever action is appropriate and retry the command.

Viewing Status of LUs

You can view status information on LINKix LUs using the Manage program or with the *comlu* utility. This section explains how to view LU status with *comlu*; for information on viewing LU status using the Manage program, see chapter 2 of this manual.

To view the status of LUs, enter the following command at the UNIX command prompt:

```
comlu -n nodename [-3][-A][-L][-R][-X]
```

where *nodename* identifies the node on which you want to view LU status. Only one node can be viewed at a time.

Note: The parameters and options (explained in detail below) must be entered in the order shown. Also, the options are case sensitive.

-3 list the status of all 3270 LUs on the selected node.

-A list the status of all APPC local LUs on the selected node.

-L list the status of all LUA LUs on the selected node.

-R list the status of all RJE LUs on the selected node.

-x suppress column headings in the output. This option is intended to allow the output from *comlu* to be processed by a program or shell script.

The *comlu* command returns an exit code of zero if it completes successfully. If an error occurs, the exit code is non-zero. For explanations of the exit codes which may be returned, see the LINKix *Diagnostics Guide*.

Output Format

The following example shows the output for each LU type. The first line of the output is a set of column headings for the data; this can be suppressed using the -x option (see above).

Additional Utilities

LU Name	Type	Connectn	LU#	LU State	User/Wkst/#sessions	Started	Descript.
COL01	3270	NEWCONN	001	LU-LU	steve	Jun 01 14:27	LU 1 at A
TESTLU	APPC			No_Session	0		TESTTP LU
OLDLU5	LUA	OLDCONN	055	SSCP	andy	Jun 01 15:03	At HQ hos
RJELU2	RJE		102	LU-LU	R427	Jun 01 15:04	LUs for R

The data is displayed in the following columns:

LU Name	The name of the LU as specified in the configuration.
Type	The LU type - 3270, APPC, LUA or RJE.
Connectn	The connection used by the LU. This column is blank if the LU is not active.
LU#	The LU number as specified in the configuration. This column is blank for APPC independent LUs.
LU State	The activation state of the LU - LU-LU (in session), SSCP (communicating with the host SSCP) or No_Session (not in use).
User/Wkst/ #sessions	the Information in this column depends on the LU tupe. 3270 the name of the 3270 user, or <FREE> if the LU is not in use APPC the number of sessions using the LU (normally 1, unless the LU supports parallel sessions), or 0 if the LU is not in use LUA the login ID with which the LUA application using this LU is running, or <FREE> if the LU is not in use RJE the name of the RJE workstation using this LU, or <FREE> if the LU is not in use
Started	The date and time at which the user, application, or workstation started using the LU. This column is blank if the LU is not in use.
Descript.	The first 9 characters of the description of this LU, as specified in the configuration file.

Any other message indicates an error condition, and begins either with “Syntax error” or “Error,” followed by information on the cause of the error. “Syntax error” messages indicate that you entered an invalid option or parameter, or that you did not supply a required one; check the correct syntax and retry. “Error” messages indicate that the command syntax was correct, but that the command could not be issued (for example, because a required LINKix component was not started or because an operating system error occurred). Take whatever action is appropriate and retry the command.

Additional Utilities

Index

3 2 7 0

- 3270 LUs
 - managing, 2-40
 - viewing status, 2-40, 2-42, 4-5
- 3270 sessions
 - terminating, 2-73
- 3270 users
 - forcing off, 2-73
 - managing, 2-72

A

- Activating
 - APPC sessions, 2-47
 - connections, 2-64
 - links, 2-12, 2-13
 - local nodes, 2-8, 2-9
 - SDLC links, 2-14
 - Token Ring links, 2-17
 - VX25 links, 2-16
 - X.25 links, 2-14
- APPC
 - queued TPs
 - starting, 2-20
 - starting with tracing, 2-21
 - terminating, 2-28
 - tracing, 2-23
 - queued TPs (active)
 - managing, 2-36
 - tracing, 2-37
- APPC applications
 - managing, 2-75
 - terminating, 2-76
- APPC LUs
 - activating sessions, 2-47, 2-49
 - managing, 2-45

- session details, 2-47
 - viewing status, 4-5
- APPC sessions
 - activating, 2-47, 2-49
 - deactivating, 2-46, 2-47, 2-48
 - stopping, 2-52
 - APPC TPs
 - managing, 2-4

B

- Browse (*combrowse*), 3-1
 - exiting, 3-23
 - menu
 - display, 3-3
 - file, 3-3
 - starting, 3-2
 - starting Motif version, 3-2

C

- combrowse*, 2-3
- comconn*, 4-2
- comlu*, 4-5
- commanage*, 3-2
- Connections
 - activating, 2-64, 4-3
 - deactivating, 2-64, 4-4
 - managing, 2-63, 4-2
 - tracing
 - SDLC, 2-15, 2-31
 - Token Ring, 2-18, 2-35
 - VX25, 2-17, 2-33
 - X.25, 2-31
- CPI-C
 - queued TPs
 - starting, 2-20
 - starting with tracing, 2-21

- terminating, 2-28
- tracing, 2-23
- queued TPs (active)
 - managing, 2-36
 - tracing, 2-37
- CPI-C applications
 - managing, 2-75
 - terminating, 2-76
- CPI-C TPs
 - managing, 2-4
- Customizing
 - log file view, 3-6
 - trace file view, 3-15

D

- Deactivating
 - APPC sessions, 2-47, 2-48
 - connections, 2-64
- Dialogs
 - 3270 LU, 2-42
 - 3270 LUs, 2-40
 - 3270 Users, 2-72
 - Activate APPC Sessions, 2-49, 2-50, 2-51
 - APPC and CPI-C Applications, 2-75
 - APPC LU, 2-47
 - APPC LUs, 2-45
 - Audit Log/Trace, 2-66
 - Audit Log/Trace for Link (SDLC), 2-30
 - Audit Log/Trace for Link (Token Ring), 2-34
 - Audit Log/Trace for Link (Vendor X.25), 2-32
 - Audit Log/Trace for NETMAN, 2-69
 - Audit Log/Trace for TP, 2-36
 - Connections, 2-63
 - Copy and Reset File, 3-19
 - Deactivate APPC Sessions, 2-52
 - Deactivate Connection, 2-64
 - Log file display options, 3-7

- Log file filter options, 3-8
- LUA Applications, 2-78
- LUA LU, 2-57
- LUA LUs, 2-55
- Open Log files, 3-4
- Open Trace File, 3-12
- Print Current File, 3-21
- RJE LU, 2-61
- RJE LUs, 2-59
- RJE Workstations, 2-80
- Select Resource Type for Zoom, 2-39
- Sessions with Partner LU, 2-54
- Start Link, 2-12
- Start Link with Trace (SDLC), 2-14
- Start Link with Trace (Token Ring), 2-17
- Start Link with Trace (VX25), 2-16
- Start NETMAN, 2-24, 2-26
- Start Node, 2-8, 2-10
- start TP, 2-20, 2-22
- Start/Manage Services, 2-5
- Terminate Service, 2-28
- Trace File Display Options, 3-15
- Trace File Filter Options, 3-16
- Display options
 - log files, 3-7
 - trace files, 3-15
- Displaying
 - log files, 3-4
 - trace files, 3-13

F

- Filter options
 - log files, 3-8
 - trace files, 3-16
- Forcing off
 - 3270 users, 2-73
 - APPC applications, 2-76
 - CPI-C applications, 2-76
 - LUA applications, 2-79
 - RJE workstations, 2-81

I

Internal messages
tracing, 2-27

L

Links, 2-5
managing, 2-30
SDLC, 2-30
VX25, 2-32
X.25, 2-30
starting, 2-12
starting with tracing, 2-13
terminating, 2-28
tracing, 2-15
SDLC, 2-31
X.25, 2-31
Local nodes, 2-5, 2-8
managing resources, 2-38
starting, 2-8
starting with tracing, 2-9
Local services, 2-5
Log files
changing log level, 2-66
customizing view, 3-6
display options, 3-7
filter options, 3-8
opening, 3-4
printing reports, 3-21
resetting, 3-19
viewing, 3-4
navigation keys, 3-6
LUA applications
managing, 2-78
terminating, 2-79
LUA LUs
managing, 2-55
viewing status, 2-55, 2-57
viewing status, 4-5

M

Manage (*commanage*), 2-1
main screen, 2-3
starting, 2-3
Managing
3270 LUs, 2-40
3270 users, 2-72
APPC applications, 2-75
APPC LUs, 2-45
APPC TPs, 2-4
connections, 2-63
CPI-C applications, 2-75
CPI-C TPs, 2-4
links, 2-30
local node resources, 2-38
LUA applications, 2-78
LUA LUs, 2-55
queued TPs (active), 2-36
RJE workstations, 2-80
SDLC links, 2-30
services, 2-2
VX25 links, 2-32
X.25 links, 2-30

N

Network management daemon, 2-24,
2-69
starting, 2-24
starting with tracing, 2-26

O

Opening
log files, 3-4
trace files, 3-12

P

Printing reports
log files, 3-21
trace files, 3-21

Q

- Queued TPs
 - starting, 2-20
 - starting with tracing, 2-21
 - terminating, 2-28
 - tracing, 2-23
- Queued TPs (active)
 - managing, 2-36
 - tracing, 2-37

R

- Resetting
 - log files, 3-19
 - trace files, 3-19
- RJE LUs
 - viewing status, 2-59, 2-61
 - viewing status, 4-5
- RJE workstations
 - forcing off, 2-81
 - managing, 2-80

S

- SDLC connections
 - tracing, 2-15, 2-31
- SDLC links
 - managing, 2-30
 - starting with tracing, 2-14
 - tracing, 2-15, 2-31
- Services
 - starting, 2-5
 - status, 2-6, 2-7
 - stopping, 2-5

Starting

- links, 2-12
- local nodes, 2-8
- Manage, 2-3
- network management daemon, 2-24
- queued TPs, 2-20

Starting with tracing

- links, 2-13
- local nodes, 2-9
- network management daemon, 2-26
- queued TPs, 2-21
- SDLC links, 2-14
- Token Ring links, 2-17
- VX25 links, 2-16
- X.25 links, 2-14

Status (viewing)

- 3270 LUs, 2-42
- LUA LUs, 2-55, 2-57
- RJE LUs, 2-59, 2-61

Stopping

- APPC sessions, 2-52

T

Terminating

- links, 2-28
- queued TPs, 2-28

Token Ring adapter interface

- tracing, 2-18, 2-35

Token Ring connections

- tracing, 2-18, 2-35

Token Ring DLPI

- tracing, 2-18, 2-35

Token Ring links

- starting with tracing, 2-17

Trace files

- customizing display, 3-15
- display options, 3-15
- displaying, 3-13
- filter options, 3-16
- opening, 3-12

- printing reports, 3-21
- resetting, 3-19
- viewing, 3-12
 - navigation keys, 3-14

Tracing

- changing options, 2-66
- internal messages, 2-27
- links, 2-13
- local nodes, 2-9
- network management daemon, 2-26
- queued TPs, 2-23
- queued TPs (active), 2-37
- SDLC connections, 2-15, 2-31
- SDLC links, 2-14, 2-15, 2-31
- Token Ring adapter interface, 2-18, 2-35
- Token Ring connections, 2-18, 2-35
- Token Ring DLPI, 2-18, 2-35
- Token Ring links, 2-17
- VX25 connections, 2-17, 2-33
- VX25 links, 2-16
- X.25 connections, 2-31
- X.25 links, 2-14, 2-31

- managing, 2-30
- starting with tracing, 2-14
- tracing, 2-31

V

Viewing

- log files, 3-4
 - navigation keys, 3-6
- trace files, 3-12
 - navigation keys, 3-14

- VX25 connections
 - tracing, 2-17, 2-33

- VX25 links
 - managing, 2-32
 - starting with tracing, 2-16

X

- X.25 connections
 - tracing, 2-31
- X.25 links

