SunNet X.25 Application Guide

The PAD (User) and X.29 (Server) Programs

Sun Microsystems, Inc. • 2550 Garcia Avenue • Mountain View, CA 94043 • 415-960-1300

Part No: 800-5012-05
pre-FCS version of 12 January 1991
SunNet 7.0

This software and documentation is based in part on the Fourth Berkeley Software Distribution under license from the Regents of the University of California. We acknowledge the following individuals and institutions for their role in its development: The Regents of the University of California, the Electrical Engineering and Computer Sciences Department at the Berkeley Campus of the University of California, and Other Contributors.

Preface

Purpose and Audience

This manual tells Sun™ workstation users how to use the PAD and X.29 server applications that are shipped with SunNet X.25. This User's Guide is intended for users who are familiar with SunOS™.

This manual assumes that a system administrator or other personnel has installed and configured SunNet X.25 according to the instructions in the *SunNet X.25 System Administration Manual* or *SunNet OSI Installation Guide*.

Summary of Contents

Chapter 1, "Overview," presents an overview of the SunNet X.25 applications.

Chapter 2, "The PAD Application," describes the PAD application, tells how to make an X.29 connection to a remote server, explains profile files, describes the PAD executive, and lists error messages.

Chapter 3, "Configuring and Running the X.29 Server Application," describes the X.29 application and tells how to determine access for remote machines to the machine running the `x29` program.

Appendix A, "The PAD Parameter Set," provides a list of the PAD (X.3) parameters with their default values.

Appendix B, "Glossary," lists terms and acronyms used in this manual, with their definitions.

Conventions Used in this Manual

This manual uses the following conventions:

- We use the acronym PSDN to refer to any public or private packet-switching network that makes available X.25 services to its users.
- We use the acronym HDLC (High-level Data Link Control) to refer the Link Access Protocol, Balanced (LAPB), which is a subset of HDLC that is used under the X.25 Packet Layer Protocol.

We use the following typographic conventions:

`Typewriter font`
: Represents what the system prints on your workstation screen and is used for program and file names.

`Boldface typewriter font`
: Indicates literal user input, typically commands and responses to prompts that you can type in exactly as printed in this manual.

Boldface font
Emphasizes information within the text.

Italic font
Indicates variables or parameters that you replace with an appropriate word or string. Also used for emphasis.

`hostname%`
Represents your system's prompt for a non-privileged user's account.

`hostname#`
Represents your system's prompt for the `root` (super-user) account.

Gray-shaded
Contain text that represents interactive sessions. User input is indicated by **`boldface typewriter font`**.

Non-shaded
Contain text that represents part of a configuration file or program output.

1

Overview

This chapter briefly describes the PAD and X.29 server applications that are shipped and automatically installed with SunNet X.25. The information in this chapter is background in nature; it is not required for the use of the applications.

Appendix B contains a list of the terms and acronyms used in this manual, with their definitions.

1.1. The X.3 and X.29 Standards

The X.3 standard grew out of users' need to establish sessions between their asynchronous, character-oriented terminals and synchronous, packet-mode hosts, across a PSDN. X.3 defines the characteristics of a PAD, which stands for packet assembler/disassembler. A PAD is a translation device that assembles packets from the characters it receives from an asynchronous terminal (character-mode DTE or DTE-C, in CCITT terms) and forwards those packets on a virtual circuit across a PSDN, usually to a remote host. In the reverse direction, a PAD receives packets from the remote host, across a PSDN, and disassembles those packets into a character stream that can be read by a connected asynchronous terminal. X.3 defines characteristics such as data rate, character echo, editing, and flow control of the PAD for a given DTE-C. The set of terminal characteristics controlled by a PAD are referred to as "The PAD Parameters." These are listed in Appendix A.

X.29 defines how the remote host and the PAD exchange and negotiate the settings of the PAD parameters to allow for communication between host and terminal. There are times when the host must tell the PAD how to operate with a particular application.

1.2. Implementation of the X.3 and X.29 Standards

The PAD and X.29 applications implement the X.3 and X.29 standards in the following ways:

- The PAD application enables a Sun workstation to act as a PAD, controlling the characteristics of the virtual terminal, which is created by the `pad` program.
- The X.29 server application enables a Sun workstation, acting as a packet-mode host, to control the parameters of a remote PAD.
- Both the X.29 and PAD applications support all of the PAD parameters, except parameter 6 (PAD service signals).

1.3. PAD and X.29 Server Applications

The SunNet X.25 PAD and X.29 applications are described as follows:

- The PAD application allows Sun workstations to connect to a remote host (in CCITT terms, a packet-mode DTE) across a PSDN. In this function, the Sun machine plays the role of a character-mode terminal (in CCITT terms, a DTE-C). In addition to providing a virtual terminal function, the software also functions as a PAD. As a PAD, the application allows you to set the characteristics of your virtual terminal and can respond to the character-display requirements of applications running on the remote host.
- The X.29 server application allows a Sun workstation to act as a host to character-mode terminals (virtual or real) that are connected to the Sun machine across a PSDN. In its role as a packet-mode DTE, the Sun can respond to remote PADs and can have virtual terminal sessions with a variety of Sun and non-Sun machines. As a remote host, the X.29 server application can support applications running on the X.25 gateway machine or on other machines on the TCP/IP network.

Both PAD and X.29 applications assume a correctly installed and configured X.25 gateway that can establish and maintain the required virtual circuits. You can use these two applications independently of one another.

Figure 1-1 *PAD and X.29 Applications*

1326

The value of the PAD and X.29 applications is multiplied by the fact that the X.25 gateway, on top of which the programs run, is available to any workstation on the same TCP/IP internetwork. This internetwork availability has the following benefits:

- Using the PAD application, any workstation user on the internetwork can `rlogin` to the X.25 gateway machine and establish a virtual terminal session across a PSDN with a remote host.

- Using the X.29 server application, any device acting as a remote DTE-C can establish a terminal session with the X.25 gateway machine. With this session established, the remote terminal user can use `rlogin`, `telnet`, or `dnalogin` to access a variety of Sun and non-Sun resources on the TCP/IP internetwork.

The PAD and X.29 applications support multiple, simultaneous virtual terminal sessions, in both PAD-to-host (PAD) and host-to-terminal (X.29) directions.

2

The PAD Application

This chapter is intended for Sun workstation users who want to access a remote packet-mode host across a PSDN or LAN. Specifically, this chapter explains how to use the `pad` program, stored in `/usr/sunlink/x25`. The `pad` program is installed when you install SunNet X.25. We assume you or your system administrator have correctly installed and configured SunNet X.25, according to the procedures in the *SunNet X.25 System Administration Manual* or *SunNet OSI Installation Guide*. Also, you or your system administrator must successfully start X.25 on each link you intend to use before invoking the PAD application.

In addition to instructions on running `pad`, this chapter introduces the PAD application's profile feature. Profiles allow you to associate a name with a destination address and a set of PAD parameter settings. You can then specify this name when you invoke the `pad` program. The chapter concludes with an explanation of the PAD executive.

2.1. Setting Terminal Type

If the remote host to which you are connecting with `pad` is a UNIX† (including SunOS) host, you must set the local terminal type after you log in to the host. This is true whether you are working at the X.25 gateway machine itself, or are logged in to the gateway from a networked workstation. To set the terminal type correctly, perform the following steps:

1. Before invoking `pad`, enter the following command:

```
user% set | grep term
```

   This command responds with output such as `term sun-cmd` or `term vt100`. The second word (`sun-cmd` or `vt100` in our examples) is the terminal type. Record this type so that you can enter it after you log in to the X.29 server.

2. Later, after you have used `pad` to connect to the remote host, enter the `set` and `tset` commands shown below to set the local terminal type on the remote host. For example, if your local terminal type is *sun-cmd*, enter the following commands at the remote host's prompt:

† UNIX is a registered trademark of UNIX Systems Laboratory, Inc.

```
remotehost% set term=sun-cmd
remotehost% tset
```

The `set` command, as used above, sets the built-in C shell variable `term` to your terminal type. The second command, `tset`, determines the type of terminal involved, then performs necessary initializations and mode settings. See `tset(1)` for further information.

The commands shown above are required because the X.29 protocol does not provide for the transferring of terminal type and the number of rows to the remote host.

2.2. Invoking the `pad` Program

The general form of a `pad` command is:

```
user% pad [<options>] [<X.121_addr>]
```

The argument *<X.121_addr>* is the remote X.121 address. Note that this argument is optional. You need not specify an X.121 address if one of the *<options>* is the called AEF and this AEF matches an AEF route set up by your SunNet X.25 administrator. When you specify an AEF and no X.121 address, if the administrator has configured routing correctly, the SunNet X.25 routing function supplies the X.121 address and, if a LAN interface is selected, the remote LSAP address. The SunNet X.25 routing function is described in the *SunNet X.25 System Administration Manual*.

For example, to connect to the remote host over a PSDN at address 476507206100 and bill that host for the cost of the call, enter the following command:

```
user% pad –r 476507206100
```

The link over which you make your `pad` call can be determined by the SunNet X.25 routing function, as set up by your administrator. Alternatively, you can specify the link using `pad`'s `-l` option.

If you use the `-l` option, you must supply all of the parameters required for the call. If an X.121 address is required, you must enter it in exactly the form required by the PSDN. For example, if the PSDN expects a prefix digit, you must include such a digit in your X.121 address. For a call over a LAN, use of `-l` means that you have to supply at least a called LSAP address. For example, to make a call over a LAN using `-l`, you enter:

```
user% pad -l2 -no4922222200 -m8:0:20:0:49:af,7e
```

The LSEL above (`7e`, at the end of the MAC address) is optional. SunNet X.25 supplies the LSEL configured for the specified interface if you do not supply it.

Another form of the pad command makes use of the profile feature, described in section 2.3, below. Using this feature, you can specify a name to stand for an address and a set of parameters. For example, in the following command,

```
user% pad surya
```

the name surya is an entry in the file /etc/sunlink/x25/x29profile, which is also described in the section "Profiles," below.

When you invoke pad with a profile entry, pad automatically looks for a matching entry in the file /etc/sunlink/x25/x29profile, regardless of the directory from which you invoke pad.

Command Line Options

Table 2-1 lists the options you can use when invoking the pad program.

Table 2-1 pad *Command Line Options*

Option	*Description*
–b*n*	Sets the break key to the ASCII character designated by *n*. For example, to set the break key to Ctrl-B, specify -b2 in the pad command line. By default the break key is set to Ctrl-C.
–c*n*	Specify the –c option to allow the use of the closed user group (CUG) specified by the closed user group index, *n*. Omit this option to indicate that no closed user group is to be used. The closed user group facility allows DTEs belonging to the group to communicate with each other, but precludes communication with other DTEs. Each closed user group is identified by a number, the CUG index, which is furnished by the PSDN. The value of the CUG index can range from 0 to 99. With the CUG option in effect, all incoming and outgoing calls are limited to those between members of the CUG.
–d	Specifies the debug option, so that all X.29 control messages are displayed on your terminal or window. This option is for troubleshooting purposes and is intended only for those who are familiar with the X.29 protocol.
–l*n*	Specify the -l*n* option to attempt a connection over the link specified by *n*. If you omit this parameter, SunNet X.25 automatically routes your call, using the method described in the *SunNet X.25 System Administration Manual*. If you omit this parameter and specify an X.121 address, it must be a full one—one that has a DNIC, NTN, and, optionally, a subaddress. If you do not specify a full X.121 address and omit this parameter, SunNet X.25 cannot perform proper routing. If you use -l to select a LAN interface, you must also use the -m option to specify the remote LSAP address.
–m<*dest_LSAP_addr*>	If you use the -l option to select a LAN interface, you must use this option to specify the remote LSAP address, which consists of a MAC address, such as an 802.*x*-type address, and an optional LSEL. You can specify the LSAP address in the standard form used for Ethernet addresses, for example, -m8:0:20:0:46:9e. Alternatively, you can specify the address as a hexadecimal string, for example, -m08002000469e. Using either method, if you enter an LSEL, separate it from the MAC address with a comma, for example, -m08002000469e,fe. The LSEL must be an even, hexadecimal number.

Table 2-1 `pad` *Command Line Options— Continued*

Option	*Description*
`–n`*<dest_AEF>*	Specify the called AEF as a string of up to 40 decimal and/or hexadecimal digits, for example, `-n491111111100`. By default, `pad` interprets this string as a non-OSI AEF. Precede the string with the letter `o` to indicate that the AEF is an OSI address, for example, `-no491111111100`. Precede the string with the letter `p` to indicate that the AEF is an partial OSI address, for example, `-np49`.
`–p`*n*	Proposes the call with a send and receive packet size set to *n* bytes. Legal values of *n* are `16`, `32`, `64`, `128`, `256`, `512`, `1024`, `2048`, and `4096`. The default value is `128`. This value is usually dictated by the requirements of the PSDN.
`–r`	Specify the `–r` option to request a reverse-charged connection. A reverse-charging connection request is one in which the remote host, rather than you, the caller, is billed for the cost of the call. If you want the remote host to be billed for the call, specify the `–r` option. The calling and called DTEs must agree on reverse-charging.
`–t`*[1|0]*	Set eight-bit mode with `-t1`; return to the default, seven-bit mode with `-t0`. In eight-bit mode, characters have eight bits with no parity. In seven-bit mode, characters have seven bits and parity is filtered. Do not use eight-bit mode (`-t1`) from a character-oriented terminal (i.e., a character-mode DTE).
`–u`*xyz*	Specify the option `–u`*xyz* to pass on the string *xyz* as call user data in the call request to the remote host. Often, hosts require that calling devices insert a host name or password in the call user data field. For example, if you must pass the call user data field *tranquebar* in your call request to a host at address 476507206100, enter the following command: `user%` **`pad –utranquebar 476507206100`** If you specify more than 12 bytes for the call user data field, the `pad` program makes the call using X.25's fast-select option, to carry the extra data. The fast-select option gives the remote host the choice of clearing the call, if the specified user data is sufficient to complete the transaction, or accepting the call, if the user data is insufficient.
`–w`*n*	Proposes the call with a send and receive window size set to *n* packets. The window size specifies the number of unacknowledged I-frames extant at any time. The value of *n* normally ranges from `1` to `7`. On networks that support extended mode, `n` can range from `1` to `127`. The default value is `2`. This value is usually dictated by the requirements of the remote host or the PSDN.

Table 2-1 pad *Command Line Options— Continued*

Option	*Description*
−z*xyz*	Specifies the source subaddress *xyz*. The pad program inserts the subaddress you specify with -z in the calling address field in the Call Request packet. As an example, consider the case where you have a DNIC of 6240 and are calling a remote SunNet X.25 gateway running the X.29 server application. The remote Sun has the following x29-auth entry: `sa=6240*01:ex=/bin/utopia` Assuming your PSDN supports a two-digit subaddress, you specify -z01 in your pad command line to reach utopia. (Note that the above x29-auth entry matches any address that starts with 6240 and ends with 01.)

2.3. Profiles

Use of the file /etc/sunlink/x25/x29profile allows you to associate a single name with the following:

- the pad command line options, including AEFs
- PAD parameter settings
- a remote server X.121 address

For example, assume you routinely want to connect to a remote host at X.121 address 650407610720, use “bengal” as your call user data, and set PAD parameter 1 (escape from data transfer) to 1, which allows you to use the PAD executive (described in section 2.4). Also, you want to charge the call to the remote host. You make the following entry in x29profile:

```
surya:a=650407610720:u=bengal:r=1:p1#1::
```

This line associates the name surya with the numeric X.121 address 650407610720, along with the option and parameter settings described above. Do not confuse the PAD parameter (p1#1, above) with the -p option, which would be expressed as, for example, -p=512. You can now enter the following command to connect to the remote host at 650407610720 :

```
user% pad surya
```

Your system administrator can set up an x29profile file on your network’s X.25 gateway machine, for use by all pad users who rlogin to that machine.

When you invoke pad with a profile entry, pad automatically looks for a matching entry in /etc/sunlink/x25/x29profile, regardless of the directory from which you invoke pad.

The following table lists the conventions used in `x29profile`.

Table 2-2 `x29profile` *File Conventions*

Convention	Description
`::`	Indicates the end of a profile entry.
`:a=`<*addr*>`:`	Sets the X.29 server address.
`:b=`*n*`:`	Sets the break key to character whose ASCII integer value is *n*.
`:c=`*n*`:`	Specifies that a closed user group call is being made and that the closed user group index is *n*.
`:f#`*n*:	Fixes the value of PAD parameter *n* so that the remote host cannot change it. `pad` will use the parameter setting specified in the `p` option (below) or the default value if the `p` option is not specified.
`:l=`*n*:	Specifies the link to be used for the outgoing call, where *n* specifies the link number.
`:m=`*n*`:`	Specifies the destination LSAP address, a six-byte MAC address, plus an optional one-byte LSEL, expressed as a string of hexadecimal digits. Do not use colons within the LSAP address. For example, `m=08009200bacf`. If you specify an LSEL, use a comma to separate the LSEL from the MAC address, for example, `m=08009200bacf,7e`. This parameter is mandatory if you use the `l` parameter to select a LAN interface.
`:n=`*n*`:`	Specifies the called AEF, expressed as a string of decimal and/or hexadecimal digits. The AEF can be an OSI NSAP address, a partial-OSI address, or a non-OSI address. Precede OSI addresses with the letter `o`, for example, `:n=o5711112035551212555`. Precede partial-OSI addresses with the letter `p`, for example, `:n=p57`. Non-OSI AEFs are not preceded by a letter, for example, `:n=81234`.
`:p`*n*#*m*`:`	Sets PAD parameter *n* to have the value *m*. Entries of this type allow you to preset PAD parameter profiles.
`:r=`*n*`:`	Controls whether the call being made is reverse charged. If *n* is zero, reverse charging is *not* used. If *n* is non-zero, a reverse-charged call is made. The default value of *n* is `0`.

Table 2-2 `x29profile` *File Conventions— Continued*

Convention	Description
`:s=`*n*`:`	Sets the default send and receive packet size to *n* bytes. Legal values of *n* are `16`, `32`, `64`, `128`, `256`, `512`, `1024`, `2048`, `4096`. The default value is determined by the value for the `def_packet_size` parameter in the link configuration file.
`:t=`*[0|1]*`:`	If `t` is `0`, the parity bit on incoming and outgoing characters is filtered out (seven-bit mode). If `t` is `1`, the parity bit is not filtered (eight-bit mode). The default value is `0`.
`:u=`*xyz*	Places the string *xyz* in the call user data field in the call request.
`:w=`*n*`:`	Sets the send window size to *n* packets. The value of *n* may range from `1` to `127`. The default send window size is determined by the value of the `def_window_size` parameter in the link configuration file.
`:z=`*xyz*`:`	Sets your source subaddress to *xyz*.

Note that you cannot have white-space characters within a profile line. Use colons to separate the fields of the profile entry and double colons to terminate the entry. Comment lines begin with the # character.

A profile entry must have either an X.121 address or a called AEF. Use of the AEF is valid only in a 1984 X.25 environment. If you use the `l` parameter to explicitly select an interface (thereby overriding routing) and specify an X.121 address (for the `a` parameter), this address must be in exactly the form required by the PSDN. If you use the `l` parameter to specify a LAN interface, you must also specify a destination LSAP address.

The following is an example of an `x29profile` file:

```
# @(#)x29profile 1.6 90/09/04
#
# This is a sample X.29 PAD profile specification.  Each profile is
# specified by a string.  There may be no whitespace characters within
# a string that specifies a profile.

dowjones:a=650465437860::
# The above specifies the abbreviation 'dowjones' for a service
# obtained by calling the address 650465437860.  The abreviation, which
# names the service in a suitable way, should appear first.  The 'a'
# parameter specifies the address, which should usually be a full
# X.121 address (ie, has DNIC, NTN, and optionally, subaddress).  The
# end of the profile is indicated by '::'.

news:a=639761209830:u=asdfgh:l=1::
# 'u=asdfgh' indicates that a the data string 'asdfgh' will be sent as
# user data following the X.29 protocol id of 01 00 00 00.
# 'l=1' indicates that the call is to be always made over link 1.

sunmicro:a=4042383100:p2#0:b=02:z=12:p1#1:f#1::
# 'p2#0' sets parameter 2 to 0
# 'b=02' sets the break character to ascii 02 (ctrl-B)
# 'z=12' sets the source subaddress in call request to 12
# 'p1#1' sets parameter 1 (pad recall) to 1
# 'f#1' fixes the value of parameter 1, so host cannot change it - thus
#       pad recall cannot be turned off in this case

somehost:a=86456742658:r=1:c=12:s=128:w=5:t=1::
# 'r=1' specifies that reverse charging should be used
# 'c=12' specifies that a closed user group request is being made and
# that the closed user group index is 12.
# 's=128' sets packet size to 128
# 'w=5' sets window size to 5
# 't=1' sets transparent mode on (to pass 8-bit data)

lanhost:n=34675432167:m=080020096f02::
# 'n=34675432167' specifies the called address extension for lanhost
# 'm=080020096f02' specifies its MAC address on the LAN

lanhost2:n=8764577675:m=080020096f02,7e::
# 'n=8764577675' specifies the called address extension for lanhost2 -
#       this is a non-OSI AEF.
# 'm=080020096f02,7e' specifies its MAC address and l-selector

lanhost3:n=o57788744:m=080020096f02::
# 'n=o57788744' specifies the called address extension for lanhost3 -
#       this is an OSI NSAP.
# 'm=080020096f02' specifies its MAC address on the LAN.
```

2.4. PAD Executive

Most often, you do not even need to be aware of, much less change, any of the local PAD parameter settings. These settings are usually taken care of by the remote system running the application that you are using. However, if you must change a setting, you can do so through the PAD executive. For example, if you need to change the line delete character from Ctrl-U to Ctrl-X, you make the change by means of a command to the PAD executive.

To enter the PAD executive, type Ctrl-P (`^p`) and Return, for example:

```
user% ^P <CR>
PAD>
```

Note that the PAD executive has its own prompt, shown above. To leave the PAD executive, type Ctrl-P. At this point, the operating system accepts your keyboard input. Press Return, to redisplay the system prompt. To complete the example mentioned above, to change the line delete character from Ctrl-U to Ctrl-X, enter the following command:

```
PAD> set bufferdelete 24
```

Table 2.4, below, lists the PAD executive commands. Table 2-3, immediately following, contains a list of parameter names and values recognized by the PAD executive.

Table 2-3 *PAD Executive Commands*

Command	Description
`?`	Display a list of the entire PAD parameter set and the current values of each parameter.
`^P`	Return to data transfer state.
`help`	Provide a list of commands and their formats.
`set` *parametername parametervalue*	Sets the parameter named in *parametername* to the numeric value specified in *parametervalue*. See Table 2-4 for a list of valid parameter names and values.
`read` *parametername*	Display the current value of the parameter *parametername*.
`reset`	Reset the virtual circuit.
`clear`	Clear the virtual circuit and transmit a Clear Indication.
`break`	Send break.
`interrupt`	Transmit an Interrupt packet.

Table 2-4 *PAD Executive Parameters and Values*

Number	Name	Values
1	escape	0, 1
2	echo	0, 1
3	forward	0-255
4	timeout	0-255
5	padflowcontrol	0, 1
6	service	--
7	breakaction	0, 1, 5, 8, 21
8	datadelivery	0, 1
9	crpad	0, 1
10	linefold	0-255
11	baudrate	3
12	terminalflowcontrol	0, 1
13	linefeedinsert	0,1,4,5,6,7
14	lfpad	0, 1
15	editcontrol	0, 1
16	chardelete	0-127
17	bufferdelete	0-127
18	displayline	0-127

The following is an example of an interactive session with the PAD executive:

```
user% ^P
PAD> read breakaction
breakaction = 5
PAD> set padflowcontrol 2
Value out of range
PAD> set padflowcontrol 1
PAD> ^P
<CR>
user%
```

2.5. Troubleshooting PAD Problems

A problem with the PAD functions might be a result of a problem communicating with the PSDN or a problem with the remote host. When you encounter a problem that causes your call to be cleared, you receive a diagnostic code in the form of two pairs of digits. The first pair indicates why the call was cleared; the second pair gives additional diagnostic information. The following tables specify meanings for these numeric codes. Table 2-5 specifies reasons for call-clearing. Table 2-6 specifies diagnostic information in addition to the call-clearing cause.

Use these tables as follows: when your connection is cleared unexpectedly and you receive a four-digit code, look up the first pair of digits in the first table; look up the second pair in the second table. A zero (00) call-clearing code indicates that the call-clearing request came from the remote host. A non-zero call-clearing code indicates the call-clearing request came from with the PSDN.

For example, your call is cleared with a code `0342`. As specified in Table 2-5, `03` indicates an invalid facility request. Table 2-6 tells you that `42` (hex) means that a facility parameter (that you used) is not allowed. This combination of messages tells you that:

1. The problem was detected by the PSDN, not by the remote host.
2. The parameter settings in the `x25params` file for the SunNet X.25 gateway are unacceptable to the PSDN.

Table 2-5 *Causes for Call Clearing*

Hex Code	Decimal Code	Meaning
00	00	Originated by remote DTE
01	01	Number busy
09	09	Out of order
11	17	Remote procedure error
19	25	Reverse charging acceptance not subscribed
21	33	Incompatible destination
29	41	Fast select acceptance not subscribed
03	03	Invalid facility request
0B	11	Access barred
13	19	Local procedure error
05	05	Network congestion
0D	13	Not obtainable
15	21	RPOA out of order

In Table 2-5, note the four categories of codes, separated by the double horizontal lines:

- The first category, code 00, indicates that the call-clearing request originated with the remote host. The remaining codes, in all other categories, indicate that the call-clearing request originated within the PSDN.
- Codes in the second category indicate problems between the PSDN and the remote host.
- Codes in the third category indicate problems between the local host and the PSDN.
- Codes in the fourth category indicate problems within the PSDN.

Table 2-6 *X.25 Diagnostic Codes*

Hex Code	Decimal Code	Meaning
00	**00**	**No additional information**
01	01	Invalid P(S) (send-packet sequence number)
02	02	Invalid P(R) (receive-packet sequence number)
10-1F	**16-31**	**Invalid packet type**
20	**32**	**Packet not allowed**
21	33	Unidentifiable packet
22	34	Call on one-way logical channel
23	35	Invalid packet type on a permanent virtual circuit
24	36	Packet on an unassigned logical channel
25	37	*Reject* not subscribed to
26	38	Packet too short
27	39	Packet too long
28	40	Invalid general format identifier
29	41	Restart with nonzero in bits 1-4, 9-16
2A	42	Packet type not compatible with facility
2B	43	Unauthorized interrupt confirmation
2C	44	Unauthorized interrupt
30	**48**	**Timer expired**
31	49	For incoming call
32	50	For clear indication
33	51	For reset indication
34	52	For restart indication
40	**64**	**Call setup problem**
41	65	Facility code not allowed
42	66	Facility parameter not allowed
43	67	Invalid called address
44	68	Invalid calling address

Note in the table above that hexadecimal codes 0, 20, 30, and 40 (decimal 0, 32, 48, and 64) stand for general messages that include the specific conditions indicated by the codes that follow them. Your PSDN might return only the codes that stand for the general messages.

For both its X.29 and PAD applications, SunNet X.25 creates a file, `/var/adm/x29debug,` to which it sends error messages. These messages are intended for the use of Sun technical personnel. If you experience problems while using the X.29 or PAD applications, check this file for messages. If messages are present, forward them to your local Sun customer service representative.

2.6. PAD Log File

The PAD application creates a log file named `/usr/tmp/x29userlog`. This file contains time stamps recording the beginning and end of PAD sessions, as well as a count of total segments (64Kb units) and characters sent and received. The file also record all unsuccessful connection attempts.

The following is an example of user sessions as they might appear in `/usr/tmp/x29userlog`:

```
Mar 10 09:55:29 Pid 2980: Call connected to 9878766675, uid=3843
Mar 10 09:55:31 Pid 2980: Call cleared - segs: sent=1 rcvd=4, chars:
sent=1 rcvd=83
Mar 10 09:58:22 Pid 2996: Call connected to 4042383100, uid=3843
Mar 10 09:59:18 Pid 3005: Call connected to 311040800133, uid=769
Mar 10 09:59:22 Pid 3005: Call cleared - segs: sent=1 rcvd=4, chars:
sent=1 rcvd=83
Mar 10 10:01:01 Pid 2996: Call cleared - segs: sent=27 rcvd=64, chars:
sent=135 rcvd=988
```

Note that you can use the process id (`Pid`, in the examples above) to associate the connection and clearing of the same call.

3

Configuring and Running the X.29 Server Application

This chapter tells you how to configure and run the X.29 server application that is shipped with SunNet X.25.

The X.29 server application allows Sun workstations to act as a host to character-mode terminals (virtual or real) that are connected to the Sun across a PSDN. In its role as a packet-mode DTE, the Sun can respond to remote PADs and can have virtual terminal sessions with a variety of Sun and non-Sun machines. As a remote host, the X.29 server application can support applications running on the X.25 gateway machine or on other machines on a TCP/IP network.

Before starting the configuration described in this chapter, we assume that you have correctly installed SunNet X.25, as instructed in the *SunNet X.25 System Administration Manual* or *SunNet OSI Installation Guide*.

You invoke the X.29 server application as you would any SunOS command. To allow you ready access to the `x29` program, put the directory `/usr/sunlink/x25` in your path.

3.1. Preparation for Using the X.29 Server Application

To run X.29 server application, you must set up password protection for the gateway machine, put appropriate entries in the `/etc/sunlink/x25/x29-auth` file, then invoke the `x29` program. The following subsections describe these steps.

All persons intending to log in to the X.29 server machine over a PSDN must have a valid username entry for the server machine, either in the directory services password file or in the local `/etc/passwd` file.

You can find further details about creating user accounts in the SunOS manual *System and Network Administration*.

3.2. Setting up the `x29-auth` File

For a Sun machine that is playing the role of a packet-mode host on a PSDN, use of the `/etc/sunlink/x25/x29-auth` file allows you to

- Restrict access to the machine, based on source address, destination address, or both.
- Automatically start programs you select.
- Select whether character echoing will be done by the PAD (local echo) or the host (remote echo).

- Set PAD parameters.
- Display a banner.

SunNet X.25 is shipped with a template `x29-auth` file in the `/etc/sunlink/x25` directory. We suggest that you modify this file to suit your own needs. There is nothing you need to do to cause the `x29` program to use the file. If `/etc/sunlink/x25/x29-auth` exists, the `x29` program will use it. If the file does not exist, `x29` allows connections from any DTE that can access the SunNet X.25 machine across the PSDN.

Entry Syntax and Meaning

Most of the parameters for `x29-auth` have both uppercase and lowercase versions. Uppercase parameters apply globally; lowercase parameters apply to specific sets of users. There are parameters for source and destination addresses (in incoming packets). You select sets of users based on either or both of these addresses. A given set of lowercase parameters must be associated with a source and/or destination address parameter.

For parameters that have both uppercase and lowercase versions, place the uppercase parameters before the lowercase parameters. The lowercase parameters override the uppercase parameters for the set of users to whom the lowercase parameters apply. For example, the `BN` parameter sets the banner for all users who connect to your host. If you set the `bn` parameter, users who match the source and/or destination address parameters in the same line see the banner specified by `bn`, instead of the banner specified by `BN`.

You specify a given set of parameters on a single logical line, using the *parameter*=*value* model. Separate multiple parameters on a single line with colons. If the lowercase parameters you want to associate with a specific address parameters do not fit on a single line, use the backslash (\) character to continue the parameters on second and subsequent lines. Spaces and tabs are not allowed, except in the banner specification. Comments are allowed; precede comments with a hash mark (#) in column 1.

`x29-auth` Parameters

The following are the `x29-auth` parameters. Recall that uppercase parameters apply to all users; lowercase parameters apply only to a subset of users who are selected by source and/or destination address parameters. These address parameters are part of the following list.

`BN, bn`
: Specifies a banner that `x29` displays when users connect to the local host. For example, `BN=Welcome to the Fun House!` causes `x29` to display “Welcome to the Fun House!” to connecting users. The X.29 server application appends a newline when it displays the string you specify. By default there is no banner.

`EX, ex`
: Specifies a program that `x29` runs for the connected user. For example, `EX=/usr/bin/special` causes `x29` to run the program in `/usr/bin/special` when users make their connections to the local host. By default, `x29` runs `/bin/login`. Users who write their own programs must note that `x29` invokes programs with the arguments `-h` *xyz* where *xyz*

is the number in the calling address field in the received Incoming Call packet.

`MODE, mode`
By default, `x29` sets PAD parameters 1, 13, 14, 15, 16, 17, and 18. If you set the `MODE` (or `mode`) parameter, `x29` will not set those PAD parameters. This parameter accepts only a value of 80, so `MODE=80` means "do not set the (subset above) PAD parameters."

`N,n`
Specify `on` for native mode and `off` for non-native mode. In native mode, the `x29` host echoes the characters the user types at her terminal. In non-native mode, the PAD echoes the characters. The default is non-native mode (PAD echoing).

`PC`*n*`, pc`*n*
(*n* is the PAD parameter number.) Specifies PAD parameter values that `x29` uses in non-native mode. For example, `PC3=126` sets the forwarding character to the set of characters specified by the value 126. This setting has effect only in non-native mode. If a user runs an application, such as `vi`, that puts the terminal in native mode, or if the terminal always operates in native mode, this parameter will not be used. Instead, `x29` uses the value specified for the `PN` parameter (see following) or the default value for native mode. You cannot use this parameter to set values for PAD parameters 2, 5, 6, and 11.

`PN`*n*`, pn`*n*
(*n* is the PAD parameter number.) Specifies PAD parameter values that `x29` uses in native mode. For example, `PN4=4` sets the idle timer delay to 0.2 seconds (4 x 0.05 seconds) when you are in native mode (host echo). This setting has no effect on PAD parameter 4 when the terminal is in non-native mode. You cannot use this parameter to set values for PAD parameters 5, 6, 9, 10, 11, and 14-18.

`PTY=`*qrst*
Specifies additional or alternate `pty`s to use. `pty`s have names of the form `/dev/pty`<*letter*><*hex_number*>, where *letter* is, by default, in the set p, q, r, and s and *hex_number* is a hexadecimal number in the range 0-f. For this parameter, you specify the letters that distinguish `pty`s. If you use this parameter, you must specify *all* `pty`s, not just those in addition to the default set.

`sa` Specifies the template for source (calling) X.121 addresses in incoming call packets. Use the wildcard `?` to match any single character in an address; use `*` to match a set of characters. For example, `sa=211??8*02` designates users who have source X.121 addresses that begin with 211, with any character in the fourth and fifth positions, an 8 in the sixth position, and end with 02.

`da` Specifies the template for destination (called) X.121 addresses in incoming call packets. You can use the same wildcards that you use for `sa`. For example, `da=*04` designates users who have destination X.121 addresses that end with 04.

`saef`

Specifies the template for source (calling) AEFs in incoming call packets. Use the wildcard `?` to match any single character in an address; use `*` to match a set of characters. For example, `saef=370000222????777*00` designates users who have source (calling) AEFs that begin with 370000222, with any character in the tenth through thirteenth positions, 777 in the fourteenth through sixteenth positions, and end with 00.

`daef`

Specifies the template for destination (called) AEFs in incoming call packets. You can use the same wildcards that you use for `saef`. For example, `daef=57*00` designates users who have destination addresses that begin with 57 and end with 00.

NOTE: *You can use the calling address parameters (*`da` *and* `daef`*) in combination with the called address parameters (*`sa` *and* `saef`*). If you use both parameters for a given address type (X.121 or AEF), the settings for both parameters must match the addresses in an incoming Call Request.*

The following is an example `x29-auth` file that illustrates the use of the parameters described above.

```
# @(#)x29-auth 1.4 90/09/04
#
# This is a sample X.29 configuration file.  This illustrates how you
# can use the various configuration parameters to advantage in providing
# X.29 services to different users.
#
# Set global parameters.  The following global parameters specify a
# banner and executable.  Non-native mode is used.  In 'cooked' mode,
# forwarding character set specified by 126 is used.  In 'raw' mode
# a forwarding timeout of 3/20 (0.15) sec is used.  Ptys ptyp0 through
# ptytf are used.

BN=This is our latest and greatest X.29 service
EX=/bin/login:N=off:PN4=3:PC3=126
PTY=pqrst

# Now specify parameter values for specifc users.

# The following line is for an old pad that cannot handle parameters
# 1 and 13-18.  This one is only interested in login.
# NOTE: in general, all special cases must appear before the general
# cases that match the special cases.

sa=567843256*:mode=80:bn=You weirdo!

# The following specifies that for any source address, if the destination
# address ends with 01 (the subaddress), execute /bin/database instead of
# /bin/login, and use native mode.

da=*01:ex=/bin/database:n=on

# The following specifes that for an incoming call whose source address
# has DNIC 4042 and subaddress 01, and whose destination address has at
# least three digits followed by a subaddress of 02, /bin/special is
# started up.  Also, it has a special banner.  Note the use of the '´
# character for continuation.  In cooked mode, pad recall is on, while in
# raw mode, pad recall is off.

sa=4042*01:da=???*02:pc1=1:pn1=0:ex=/bin/special:bn=This is a specialapplication just for you!

# The following entry makes use of the saef and daef parameters.

saef=571111444555*:daef=37*8800:ex=/bin/rate_retrieve:\
bn=This is for our Lancashire users.

# The following is a catch-all that handles all the users that don't
# match the above specifications.  Although both sa and da are specified,
# it is necessary only to specify one of either sa or da.  The default
# parameter values specified with global parameters are used.

sa=*:da=*
```

3.3. Starting the X.29 Server Application

To invoke the X.29 server application, simply enter the following command:

```
server# /usr/sunlink/x25/x29
```

The program runs in the background, "letting go" of the console from which you invoked it.

You can set the following options when invoking the `x29` process:

`–a`*xyz* When you specify the `-a`*xyz* option, the X.29 server application accepts calls only for the X.121 address specified by *xyz*. This option has precedence over the `-s` option, so that if you use `-a`, you do not need to use the -s option.

`–l`*n* Specify the `-l`*n* option to restrict incoming calls to those originating from the link identified by *n*.

`–r` Specify the `–r` option to permit reverse-charging connection requests. Omit this option to disallow reverse charging. A reverse charging connection request is one in which the X.29 server, rather than the calling DTE, is billed for the cost of the call. If you do not want to accept reverse-charged calls, do not use the `–r` option. Note that some PSDNs always reverse charge.

`–s`*xyz* Specify the `-s`*xyz* option to restrict incoming calls to those calls with a called address that has a subaddress of *xyz*.

The SunNet X.25 installation script, `/usr/sunlink/x25/install.x25` will, with your consent, put a command to start `x29` each time you reboot your machine in your X.25 startup file, `/etc/sunlink/x25/rc`. If you do not want this to occur, use your favorite editor to comment out the `x29` command line in your `x25/rc` file.

3.4. Running `x29` over Multiple Links

The parameter settings you make in `/etc/sunlink/x25/x29-auth` apply to all of the links—including HDLC and LLC2 links—on a given SunNet X.25 gateway machine. The `x29-auth` settings also apply to all invocations of the `x29` program on a machine.

Normally, you need to run only a single instance of the `x29` program, regardless of the number of links you support. A situation in which you might want to run multiple instances of `x29` occurs when you want one set of `x29` command-line parameters to apply to one link and not another. For example, if you want to permit reverse-charging requests over one link and not over another, you can enter commands such as the following:

```
hostname# x29 -l0 -r
hostname# x29 -l1
```

As a result of these commands, collect calls will be allowed only over link 0.

3.5. X.29 Log File

The X.29 server application creates a log file named `/var/adm/x29serverlog`. This file contains time stamps recording the beginning and end of server sessions, as well as a count of total segments (64Kb units) and characters sent and received. The file also record all unsuccessful connection attempts.

The following is an example of server sessions as they might appear in `/var/adm/x29serverlog`:

```
Feb 24 17:08:55 Pid 537: Call connected - caller(4042383100),
called(765765564), lcn(20), link(1)
Feb 24 17:19:52 Pid 537: Call cleared - segs: 52 sent, 29 rcvd; chars:
1305 sent, 629 rcvd
Feb 24 18:15:12 Pid 192: Call connected - caller(4042383100),
called(65765765765), lcn(20), link(1)
Feb 24 18:16:57 Pid 192: Call cleared - segs: 40 sent, 7 rcvd; chars:
1629 sent, 76 rcvd
Feb 27 11:30:48 Pid 220: Call connected - caller(4042383100),
called(756675765657), lcn(20), link(1)
Feb 27 11:30:49 Pid 220: Call cleared - segs: 2 sent, 1 rcvd; chars: 95
sent, 30 rcvd
```

Note that you can use the process id (`Pid`, in the examples above) to associate the connection and clearing of the same call.

A

The PAD Parameter Set

Recommendation X.3 defines the PAD operating characteristics for asynchronous DTEs, such as operating speed, character echoing, editing, and flow control. X.3 is referred to as the PAD parameters.

The following is a table of the X.3 parameter set. An asterisk indicates the default value where applicable.

Table A-1 *The X.3 Parameter Set*

Parameter	Name	Description
1	Escape from data transfer state	Parameter 1 specifies whether the terminal user is allowed to escape from the data transfer state to the waiting-for-command (PAD-executive) state. The PAD executive user is not allowed to change the setting of this parameter. Valid settings: 0 Escape not permitted. 1 Escape permitted.*
2	Echoing	Parameter 2 controls whether typed characters are echoed to standard output. The SunOS equivalent of setting parameter 2 to `0` is `stty –echo`, while `stty echo` simulates setting parameter 2 to `1`. Valid settings: 0 Don't echo. 1 Echo.*

Table A-1 *The X.3 Parameter Set— Continued*

Parameter	Name	Description
3	Data forwarding character	Parameter 3 controls the conditions under which typed characters are forwarded to the remote host. The value of parameter 3 is a sum of some set of the numbers 0, 1, 2, 4, 8, 16, 32, 64, and 128. For example, if it was desired to forward typed characters whenever a linefeed (LF), CTRL-C, (ETX), or carriage return was typed, parameter 3 would be set to 50 (50 = 32 + 16 + 2). There is no SunOS parallel that supports multiple input delimiting characters. Valid settings: 0 129th character (packet full). 1 A-Z, a-z, 0-9. 2 Carriage return* 4 ESC, BEL, ENQ, ACK* 8 DEL, CAN, DC2 16 ETX, EOT* 32 HT, LF, VT, FF* 64 All characters with values between `0x0` and `0x31`. 128 Graphic characters. The value of parameter 3 may be a sum of any of the above values.
4	Data forwarding timeout	Parameter 4 controls a parameter forwarding condition. If parameter 4 is `0`, then characters are forwarded from the user terminal to the remote host whenever the character forwarding conditions in parameter 3 are met. If parameter 4 is a non-zero number *n*, then characters in the input buffer will be transmitted every [*n* * 0.05] seconds. There is no SunOS parallel to this function. Valid settings: 0 No data forwarding timeout. 1-255: Number of 0.05 second ticks to wait before transmitting a character.

Table A-1 *The X.3 Parameter Set— Continued*

Parameter	Name	Description
5	Flow control of user's terminal by PAD	Parameter 5 controls whether the PAD is permitted to control the flow of incoming data. That is, if parameter 5 is set to `1` and the PAD is not able to accept additional data, it will transmit `X-OFF`. The PAD will transmit `X-ON` when it is once again able to accept additional data. If parameter 5 is `0`, then it is not permitted to control data flow by means of `X-ON` and `X-OFF`. There is no SunOS equivalent for this parameter. Valid settings: 0 No control.* 1 Control permitted.
6	Suppression of PAD service signals	Not supported.
7	PAD's action when break signal received	Parameter 7 specifies the type of action to be taken when the terminal user types the break character (whose default value is CTRL-C). Depending on the parameter's value, some combination of INTERRUPT, RESET, or INDICATION OF BREAK packets will be transmitted to the host. There is no equivalent function in SunOS Valid settings: 0 Do nothing. 1 Send an INTERRUPT packet to server. 2 Send a RESET packet to server. 5 Send an INTERRUPT and an INDICATION OF BREAK to server. 8 Leave data transfer state and enter waiting-for-command state. 21 As for (5) above but also set parameter 8 to 1.*
8	Suppression of data delivery to user's terminal	Parameter 8 controls whether data received from the remote host should be delivered to the user terminal. If parameter 8 is set to `1`, all data arriving from the remote host is discarded. The default value of this parameter is always `0` as no terminal profiles require a setting of `1`. Receiving or transmitting a RESET packet always has the effect of setting parameter 8 to `0`. There is no equivalent function in SunOS. Valid settings: 0 Suppression off.* 1 Suppression on.

Table A-1 *The X.3 Parameter Set— Continued*

Parameter	Name	Description
9	Padding characters after carriage return	Parameter 9 controls the number of padding characters inserted after carriage returns in the character stream. A padding character is defined to be a time interval required for one character to pass some point in space. The SunOS equivalent of manipulating parameter 9 is `stty cr0`, or `stty cr1`, etc. Valid settings: 0 No padding after carriage return.* 1-7 Number of padding characters inserted after carriage return.
10	Line folding	Parameter 10 specifies the size of the input line, that is the number of characters that will be accepted as input before a carriage return or linefeed is printed on standard output. No line folding is done when parameter 10 is set to `0`. Valid settings: 0 No line folding.* 1-255 Length of line.
11	Line speed	Parameter 11 specifies the user terminal's baud rate. The value of parameter 11 is immutable because the physical line speed cannot be changed. Valid settings: 0 110 bps. 1 134.5 bps. 2 300 bps. 3 1200/1200 bps.* 4 600 bps. 5 75 bps. 6 150 bps. 7 1800 bps. 8 200 bps. 9 100 bps. 10 50 bps. 11 1200/75 bps. 12 2400 bps. 13 4800 bps. 14 9600 bps. 15 19200 bps. 16 48000 bps. 17 56000 bps. 18 64000 bps.

Table A-1 *The X.3 Parameter Set— Continued*

Parameter	Name	Description
12	Flow control of PAD by user's terminal	Parameter 12 controls whether the terminal user is permitted to control the flow of characters from the PAD to the terminal. That is, if the user wishes to stop receiving characters from the PAD, the user transmits `X-OFF` to the PAD. To resume receiving characters from the PAD, the user transmits `X-ON` to the PAD. The SunOS equivalent of setting parameter 12 to `1` is to enable the start and stop characters with `stty start ^Q` and `stty stop ^S`. To simulate a value of `0` for parameter 12, it is necessary to disable SunOS's start and stop characters with `stty start \\377` and `stty stop \\377`. Valid settings: 0 No flow control. 1 Flow control permitted.*
13	Line feed insertion after carriage return	Parameter 13 controls the linefeed insertion conditions after carriage returns in the character stream. Six separate actions are possible depending on the value of parameter 13. There is no SunOS function equivalent to parameter 13. Valid settings: 0 No line feed insertion. 1 Insert line feed after every carriage return received from server. 4 Insert line feed after every carriage return received from user's terminal.* 5 Both (1) & (4) above. 6 Insert line feed after every carriage return written to user's terminal and transmitted to server. 7 Both (1) & (6) above.
14	Padding after line feed	Parameter 14 controls the number of padding characters inserted after linefeeds in the character stream. A padding character is defined to be a time interval required for one character to pass some point in space. The SunOS equivalent of manipulating parameter 14 is `stty` *n*10, or `stty` *n*11, etc. Valid settings: 0 Don't pad after line feed.* 1-7 Number of padding characters inserted after line feed.

Table A-1 *The X.3 Parameter Set— Continued*

Parameter	Name	Description
15	Editing	Parameter 15 controls whether typed characters can be edited by the terminal user. A value of `0` specifies no input editing and is equivalent to the SunOS `stty raw`. A value of `1` allows input editing in conjunction with parameters 16-18. Valid settings: 0 No input editing in data transfer state. 1 Editing allowed in conjunction with parameters 16-18.*
16	Character delete character	Parameter 16 specifies the character that acts as the character-delete character. A parameter 16 value of `0` disallows character deletions. In SunOS the default erase character is `^?`. Other values for the erase character may be set by `stty erase` *char* where *char* represents the new erase character. Valid settings: 0 Character delete off. 1-127 Ordinal value of delete character.
17	Buffer delete character	Parameter 17 specifies the character that acts as the buffer-delete character. A parameter 17 value of `0` disallows buffer deletions. In SunOS, the default buffer delete (kill) character is CTRL-U. Other values for the kill character may be set by `stty kill` *char* where *char* represents the new kill character. Valid settings: 0 Buffer delete off. 1-127 Ordinal value of delete character.
18	Buffer display character	Parameter 18 specifies the character that acts as the buffer display character. A parameter 18 value of `0` turns off buffer display. In SunOS, the default buffer display (`rprnt`) character is CTRL-R. Other values for the `rprnt` character may be set by `stty rprnt` *char* where *char* represents the new `rprnt` character. Valid settings: 0 Buffer display off. 1-127 Ordinal value of display character.

B

Glossary

CCITT — Consultative Committee for International Telephony and Telegraphy
An international organization of communication carriers, most of which are government telephone and telegraph agencies. The CCITT develops telecommunication standards through the use of their recommendations. The X.25, X.3, and X.29 standards originated as CCITT recommendations.

CUG — Closed User Group
A closed user group permits the DTEs belonging to the group to communicate with each other, but precludes communication with other DTEs. A single DTE can belong to multiple CUGs.

DCE — Data Circuit-terminating Equipment
The network side of the user-to-network interface.

DNIC — Data Network Identification Code
A (usually) four-digit number that identifies a specific PSDN. The DNIC is the first component of a complete X.121 address and is comparable to the exchange portion of a telephone number used for switched telephone service.

DTE — Data Terminal Equipment
The device at the user's side of a user-to-network interface. This may be a computer system (packet-mode DTE) or a character-mode terminal (DTE-C), and is both a source and destination for data.

DTE-C — Data Terminal Equipment-Character Mode
A character-mode (asynchronous) terminal that uses a PAD to connect to a remote host across a PSDN.

NTN — Network Terminal Number
The component of a complete X.121 address that identifies a specific DTE.

PAD — Packet Assembler/Disassembler
An interface device that buffers data sent to or from character mode devices, and assembles and disassembles the packets needed for X.25 operation.

PDN — Public Data Network
: A data communications network whose services are available to any user willing to pay for them. Also used to mean an X.25-compatible Private Data Network. PDN is not used in this manual.

PSDN — Packet-Switched Data Network
: A more general term than PDN, refers to any public or private packet-switching network that provides X.25-compliant interfaces to its users.

PVC — Permanent Virtual Circuit
: A permanent logical association between two physically separate DTEs that does not require call set-up or clearing procedures.

RPOA — Recognized Private Operating Agency
: An X.25 user facility that provides for user specification of a particular RPOA transit network through which a call is to be routed internationally when more than one RPOA transit network exists at an international gateway.

subaddress
: The component of an X.121 address that identifies a specific application on a DTE. The subaddress is an optional component of a complete X.121 address.

SVC — Switched Virtual Circuit
: A temporary logical association between two physically separate DTEs that exists only for the duration of the data transfer. Call set-up and call clearing procedures are required with a switched virtual circuit.

X.21bis
: A set of CCITT recommendations that define the physical interface between a DTE and a DCE of a public data network. Access to the DCE is through synchronous modems and voice-band lines. Equivalent to RS-232-C and V.24/V.28.

X.25 client
: An application layered above X.25.

X.25 gateway
: A Sun workstation on which SunNet X.25 is installed and made available to remote terminal or workstation users.

X.29
: Defines the interface between PADs and packet-mode DTEs or other PADs.

X.3
: Describes the functions of the PAD and the various parameters used to specify its mode of operation.