Xylogics

Annex Communications Server R6.0

Installation Notes

READ BEFORE INSTALLING RELEASE R6.0

These Installation Notes support R6.0 of the Annex Communications Server software.

These Installation Notes contain information specific to this release. This information is not available elsewhere in the Annex documentation.

NOTICE

The information in this manual is subject to change without notice, and should not be construed as a commitment by Xylogics, Inc. Xylogics assumes no responsibility for any errors that may appear in this document.

Table of Contents

Introduction

These installation notes are for R6.0 of the Annex Communications Server operational software. An Annex downloads the operating code from a host running Annex file server software. The Annex software can interface with any UNIX host that supports the TCP/IP protocols.

Installing the Annex software requires access to the host selected as an Annex file server. At that host, you can load and compile the Annex file server software, edit certain configuration files, and start the erpcd daemon.

This document contains the following sections:

- Preparation
- Host installation
- Annex installation
- Host-specific installation instructions
- List of files

Use these installation instructions either to install new Annexes or to update the operational code with a new release of software. If you are installing a new Annex, read all sections. For updating software, you need to read only the *Host Installation* section, and *Boot Procedures for Upgrading Software* (in the *Annex Installation* section).

The *Annex Communications Server Network Administrator's Guide* provides instructions for installing optional software.

> **Note:** These instructions cover software installation for all models of the Annex, including the original Annex, the Annex II, the Annex IIe, the Annex-802.5, and the Annex 3 . The term Annex refers to all models unless otherwise noted.

Preparation

If you are installing a new Annex, perform the following steps:

1. Select the directory that will serve as the root of the Annex source tree. This directory requires approximately 2.5 megabytes of memory to read the tape, and approximately twice this amount to build the programs.

2. Select the directory that will contain the compiled Annex support programs. The Annex installation procedures uses the **/usr/annex** directory if it exists; otherwise, it uses the **/etc** directory. (See below to select another directory.) This directory requires approximately .5 megabytes of memory.

3. Allocate space in **/usr/spool** for the Annex operational code. The boot image requires between 1 and 2 megabytes in **/usr/spool/erpcd/bfs**, which is created by the installation script. Furthermore, dumps can require from 1 to 3 megabytes for each Annex serviced by this file server.

Host Installation

This section includes the following information on the installation of the Annex software at the host:

- Contents of the distribution media
- Step-by-step installation procedures
- Host set-up procedures

Distribution Contents

The distribution media contains the Annex operational code and host-resident Annex support utilities. These utilities are distributed in *C* language source form with appropriate make files. Annex operational code is compiled.

Host-resident Annex support utilities are:

aprint	The Annex printer utility (intended for sites that cannot modify the Berkeley spooler).
ch_passwd	The Annex user password change utility.
erpcd	The expedited remote procedure call daemon, which listens for Annex requests on a reserved UDP port (121). The daemon contains two programs: **bfs**, the block file server used by the Annex to access host-resident files; and **acp**, the Access Control Protocol (ACP) program that responds to host-based security requests.
ien116d	The IEN-116 name server.
mk_rotaries	The port rotaries file **/usr/spool/erpcd/bfs/rotaries.o** definition utility.
na	The Annex Network Administrator utility
rtelnet	A pseudo-tty device create utility, which allows such UNIX programs as **tip** and **uucp** to access serial devices attached to Annex ports.
timserver	The time server that updates the Annex's time-of-day clock.

The distribution tape provides *diffs* to the standard 4.2/4.3BSD sources for the Berkeley LPD spooling system. Applied to standard sources, these *diffs* create a version of the spooling system that understands remote Annex printers. The distribution tape also contains *diffs* to sources of the 4.2BSD network daemons, **rlogind** and **telnetd**. These *diffs* contain bug fixes to these daemons. Refer to the *Annex Communications Server Network Administrator's Guide* for instructions on applying these *diffs*.

Installation Procedure

After you have mounted the distribution media on the file server host, perform the following steps:

1. Create a directory to serve as the root of the Annex software tree if it does not already exist. For example:

   ```
   # mkdir /usr/annex
   ```

2. Change to this directory. For example:

   ```
   # cd /usr/annex
   ```

3. Extract the contents from the source distribution using the following command:

   ```
   # tar -xv
   ```

   If you are installing the operational code onto a PC running the XENIX operating system, use the above command to extract the file, **tar-image**, from the diskettes. Next, using the following command, change the tar-formatted file, **tar-image**, to the desired Annex directory structure:

   ```
   # tar -xvf tar-image
   ```

   You may want to save the list of files to verify that all files in the source distribution have been extracted. For example:

   ```
   # tar -xv > annex.files          # for tape media
   ```

   or

   ```
   # tar -xvf tar-image > annex.files          # for floppy diskettes
   ```

   A list of distributed files is provided at the end of this document.

4. Run the **install-annex** script to generate a configuration of your system, and to install the Annex host utilities and **man** pages. Any prompt that has a default displays it in brackets ([ ]). Pressing **Return** enters the default.

The configuration sequence of the installation script generates two files: **src/Makefile** and **src/make.config**. These files compile and install the Annex tools. To run the script, enter the following command:

```
# ./install-annex
```

The following dialogue is a typical installation script. What you see can vary, depending on the type of system on which you are installing the Annex software.

```
        Annex R6.0 Host Utilities Installation Script

This installation shell script will examine your system and possibly ask you
questions to generate the needed  configuration to allow you to compile the Annex
host utilities.

This script does not need to be run as root. In fact, its best that you not run it
as such. Only the installation process need to be run as root. You can abort the
script at any time and restart it without any ill effects. Once install-annex
completes the configuration section, you will only need to use
                "make all"    and "make install"
from the src directory to later rebuild or re-install the utilities.

At any question prompt you can escape to a shell by typing "!". When you exit that
shell you will bounce back to the question prompt. Many of the questions will have
default answers in square brackets; typing carriage return will give you the
default.


Type carriage return to continue. Your cursor should be here-->| RET |
```

At this point, press the **Return** key to continue the installation script.

```
        Beginning configuration.

        Locating common programs...
        ar is in /bin/ar.
        awk is in /bin/awk.
        cat is in /bin/cat.
        cc is in /bin/cc.
        chown is in /etc/chown.
        chmod is in /bin/chmod.
        cp is in /bin/cp.
        date is in /bin/date.
        echo is in /bin/echo.
        expr is in /bin/expr.
        grep is in /bin/grep.
        mkdir is in /bin/mkdir.
        mv is in /bin/mv.
        ranlib is in /usr/bin/ranlib.
        rm is in /bin/rm.
        sed is in /bin/sed.

        Don't worry if any of the following aren't found...
        inetd is in /etc/inetd.
        I don't see pack out there, offhand.
        strip is in /bin/strip.
```

Next, the script prompts for the machine type and operating system of the host:

```
Select the most appropriate machine type

1.   Encore Multimax
2.   Sun
3.   UNISYS
4.   Tahoe (CCI Power6/32 et al)
5.   Convergent
6.   NCR
7.   Pyramid
8.   Concurrent
9.   PC/AT
10. Sequent PTX

0.  Generic (no special code needed)

Enter Machine Type #:  2


Select the most appropriate OS

1. 4.[23]BSD Unix (& UMAX BSD)
2. SYS_V Unix (& UMAX V)
3. XENIX System V
4. MACH

Enter OS #: 1
```

After you respond to both questions, the script verifies that the desired functions are installed on your system. If a function is not found, either *not found, adding* or *not found, substituting* displays next to the missing function.

```
Checking your system for these functions...
#include <systypes.H> ok
#include <netinet/in.h> ok
bcopy() ok
bzero() ok
dup2() ok
gethostbyname() ok
gethostname() ok
htonl() ok
htons() ok
ntohl() ok
nthos () ok
inet_addr() ok
inet_ntoa() ok
recvfrom() ok
recvmsg() ok
sendmsg() ok
sendto() ok

index ok
rindex ok
select ok
u_char ok
u_short ok
u_long ok
hostent ok
servent ok
```

If your operating system is a UNIX System V, the script asks what type of networking software you have before it displays the install and build messages. It may ask additional questions, depending on your machine type and networking software. For assistance with installing the Annex software on some specific System V hosts, refer to the section *Host-specific Installation Instructions*.

Next, the script asks you where to install the Annex utilities and the Annex **man** pages.

```
Where do you want the annex utilities installed? [/usr/annex] RET

What kind of on-line manual page layout do you have?
Select the most appropriate on-line manual page layout:

1. v7/BSD (/usr/man/man?)
2. SystemV (/usr/{catman,man}/[apu]_man/man?)
3. None (don't install on-line manual pages automatically)

Enter on-line manual page type #: [1] RET
```

You have several location options for installation of **man** pages, based on the type of system you have; you also have the option of not installing **man** pages.

If you select number **1**, to install the **man** pages using a BSD page layout, you are queried for the directories in which these manual pages reside:

```
How should the on-line manual pages be installed?

1. One directory (i.e., all in /usr/man/man1)
2. Multiple directories (i.e., under /usr/man in man1,man5,man8)

    Enter on-line manual page layout type: [1]
```

If you select number **2** to install the **man** pages in multiple directories, the following query displays:

```
What base directory should be used? [/usr/man]
```

If you select to install the **man** pages in one directory, you are asked the following questions:

```
How are manual page extensions handled in /usr/man/man1?

1. They use a fixed extension like ".1"
2. They use the extension of the manual page
   (i.e., ".1", ".5", ".8" - note that this is not typical)

Enter manual page extension type: [1] RET

What is the extension used? [1] RET
```

However you elect to install the **man** pages, the following display indicates that the configuration portion of the script is complete:

```
Configuration complete.
```

The next step in the script is to compile and install the annex tools:

```
Shall I compile the tools for you? [y]  RET
```

If you answer **y**, the following is displayed:

```
Output being saved in "Build.out"

*** Building libannex ***
         .
         .
         .
*** Building timserver ***
```

Followed by the next question:

```
Shall I install the tools for you? [y]  RET
```

If you answer **y**, the following is displayed:

```
Output being saved in "Install.out"
*** Installing Annex images ***
         .
         .
         .
*** Installing timserver ***

Done.
```

If you did not compile and install the tools, the following is displayed:

```
Installation not completed.
To compile the tools use "cd src ; make"
To install the tools and manual pages use "cd src ; make install"

Done.
```

At this point the installation procedure is complete.

If you need to rebuild or reinstall the Annex software in the future, you need to run only **make** and **make install** from the **src** directory.

Host Server Set-up Procedures

A new Annex requires the following edits to host files on the file server before downloading software:

1. Add the following line to **/etc/services** to define a UDP port for the **erpcd** daemon:

   ```
   erpc        121/udp                          # rpc listener
   ```

   Note: Perform this step *only* if you have an **/etc/services** file on your system.

2. Verify that the following line, required by the Annex **aprint** utility, is in the **/etc/services** file:

   ```
   printer     515/tcp     spooler              # line printer spooler
   ```

3. If you are installing software on a host running Sun Microsystem's Yellow Pages, you must re-initialize the yellow pages database before any changes to **/etc/services** take effect.

4. Log in as **root** and run **erpcd**, which is located in the directory where you installed the Annex utilities. As a daemon, **erpcd** detaches itself and returns your terminal to the UNIX shell. You can verify that **erpcd** is running with the command:

   ```
   # ps -aux | grep erpcd
   ```

5. Update **/etc/rc.local** to start **erpcd** when the system is booted.

6. The security server (**acp**) is disabled with the distribution. If you wish to enable the security server, you must edit the **/etc/eservices** file and remove the pound sign (**#**) in front of the line defining **acp**. (Refer to the *Annex Communications Server Network Administrator's Guide* for more information.)

Additional host set-up procedures may involve a name server and a time server. Refer to the *Annex Communications Server Network Administrator's Guide* for discussions on these set-up procedures.

Annex Installation

After the Annex software is loaded on the file server host, install the operational code by booting the Annex.

Boot Procedures for Upgrading Software

If you are upgrading Annex software, you can boot your Annex after installing the operational software on a file server host. Enter the **boot** command from either the superuser CLI or **na**. After the Annex boots, it will be up and running the latest version of the operational software.

Boot Procedures for New Annexes

This section provides a brief summary of the steps required for powering up the Annex and downloading the operational code to it.

After you have installed the Annex software on the file server host:

1. Collect information for the Annex boot parameters. The Annex needs these parameters to perform an initial boot to load the software. Enter these parameters into the Annex's EEPROM using ROM monitor commands, which are accessed through the Annex Console.

2. Select the type of connection between the file server host and the Annex. The options are:

    • The Annex and host are on the same network and the same subnet.

    • The Annex and the host are on two different networks that communicate through a gateway.

    • The Annex and the host are connected by serial ports using the Serial Line Internet Protocol (SLIP).

The following manuals contain information on connecting a terminal for use as the Annex Console and details on the ROM Monitor commands:

• *Annex II Communications Server Hardware Guide*

• *Annex IIe Communications Server Hardware Installation Guide*

• *Annex-802.5 Communications Server Hardware Guide*

• *Annex 3 Communications Server Hardware Guide*

The boot procedures for a new Annex are:

1. Set the Annex's Internet address and other initial boot parameters with the ROM monitor. Refer to Chapter 3, *ROM Monitor Commands*, in the appropriate hardware guide.

2. For all Annex models *except* the Annex 3, set the NORMAL/DIAG switch on the rear panel of the Annex to the DIAG position.

    For the Annex 3, which automatically powers up in NORMAL mode, press the Test switch on the front panel *within 30 seconds of power up* to enter TEST mode.

3. At the Annex Console (the terminal attached to Port 1), enter the **boot** command:

   ```
   monitor:: boot
   ```

   If you do not enter a file name with the command, you are prompted for one (a default file name is displayed at the prompt). Press the **Return** key to enter the default file name.

   The terminal displays messages describing the boot request, followed by messages describing the downloading.

During a normal power-up and boot sequence, the Annex front panel indicators light. Refer to the *Power-up and Test* section in Chapter 2 of the appropriate hardware guide for a discussion of the sequence. (If the indicators do not light as documented, refer to Chapter 4, *Troubleshooting Procedures*, in that guide.)

The download takes between 30 and 60 seconds for an Annex booting over the local network. If you are booting over a SLIP interface, the speed of the serial line affects the amount of time needed for the download. The length of time can be anywhere from 5 minutes for a 19200 baud line, to 80 minutes for a 1200 baud line.

4. For all Annex models *except* the Annex 3:

   When you see EOF display on the terminal, set the NORMAL/DIAG switch to the NORMAL position. Now you are at the Annex CLI (the normal user command level). The Annex is up and running.

5. For the Annex 3:

   Do not press the **Test** switch when you see EOF display on the terminal.

   Note: The CLI does not appear on the console terminal. You can access it only from a terminal connected to one of the Annex 3's serial ports. Refer to Chapter 2 of *Annex 3 Communications Server Hardware Guide* for instructions on connecting serial devices.

After the Annex has loaded the operational code, the Command Line Interpreter (CLI) banner and prompt appear as:

```
Annex Command Line Interpreter * Copyright 1991 Xylogics, Inc.

annex:
```

Host-specific Installation Instructions

This section provides installation instructions for the following manufacturers:

- ARIX 825
- ARIX/CMC
- Bull
- HP 9000 (HP-UX)
- MIPS 1000 and Silicon Graphics
- Motorola
- SCO TCP/IP

The following instructions have been tested on the hardware platforms and operating system software listed under each manufacturer.

You also need to check whether your host has an alternate include directory for the network code and, if so, where it is located. You will need to locate the directory in which the include files reside.

ARIX 825

The following set of instructions was tested using the ARIX 825 hardware platform and the Arete 8000S 3.32 Vers 3.3.3 ARIX V3.1 operating system software running Excelan network software. These instructions are not guaranteed to work with all ARIX 825 systems, but should provide guidelines for other ARIX 825 machines and software releases.

To install the Annex distribution software on the above system, perform the following steps:

Step 1. Run the **install-annex** script. Answer the following questions as noted:

    Select the most appropriate machine type

Select: Generic (no special code needed)

    Select the most appropriate OS

Select: System V Unix (& UMAX V)

    What kind of Network Software does your OS have?

Select: 2.Excelan EXOS package

    Does your system have any auxiliary libraries for the network code?[n]

Select: *y*

> Enter any additional loader options for the network libraries

Enter: -lsocket -lnfs.c

> Does your system have an alternate include directory for network code[n]?

Select: *y*

> Where are the network include files located?

Enter: /usr/include/EXOS

Step 2. Do not compile the tools at the end of the installation script. That is, answer **n** to the following question:

> Shall I compile the tools for you?

Step 3. Copy:

> /usr/include/sys/time.h

to:

> src/inc/sys/time.h

Step 4. Copy:

> /usr/include/rpc/netdb.h

to:

> src/inc/rpc/netdb.h

Step 5. Then, edit the copy of the **netdb.h** file to comment out the definitions of **servent** and **hostent**.

Step 6. Compile and install the tools:

```
cd src
make
make install
```

ARIX/CMC

The following set of instructions is not guaranteed to work with all CMC systems, but should provide guidelines for other CMC machines and software releases.

To install the Annex distribution software on the above system, perform the following steps:

Step 1. Run the **install–annex** script. Answer the following questions as noted:

```
Select the most appropriate machine type
```

Select: Generic (no special code needed)

```
Select the most appropriate OS
```

Select: System V Unix (& UMAX V)

```
What kind of Network Software does your OS have?
```

Select: CMC Package

Step 2. Do not compile the tools at the end of the installation script. That is, answer **n** to the following question:

```
Shall I compile the tools for you?
```

Step 3. Add the following lines to the **src/inc/config.h** file:

```
#define big_endian
#define need_sendto
#define sendto _sendto
```

Also, remove the line **have_msghdr** if defined.

Step 4. Copy :

```
src/include/sys/socket.h
```

to:

```
src/inc/sys/socket.h
```

Next, edit the copy to remove (using **#ifdef**) the definition of the **msghdr** structure.

Step 5. Compile and install **erpcd**:

```
cd src
make
make install
```

Step 6. Edit **src/inc/config.h** file again to remove the following lines:

```
#define need_sendto
#define sendto _sendto
```

Step 7. Compile and install **na**:

```
# cd src/na
# make -f ../make.config -f Makefile install
```

Bull

The following set of instructions were testing using the Bull DPX-2 hardware platform and the BOS 68000 operating system. These instructions are not guaranteed to work with all Bull systems, but these instructions should provide guidelines for other Bull machines and software releases.

To install the Annex distribution software on the above system, perform the following steps:

Step 1. Run the **install-annex** script. Answer the following questions as noted:

```
Select the most appropriate machine type
```

Select: Generic (no special code needed)

```
Select the most appropriate OS
```

Select: System V Unix (& UMAX V)

```
What kind of Network Software does your OS have?
```

Select: 4.[234]BSD or compatible

```
Does your system have any auxiliary libraries for the network code?[n]
```

Select: y

```
Enter any additional loader options for the network libraries
```

Enter: -linet

```
Does your system have an alternate include directory for network code[n]?
```

Select: n

Step 2. Do not compile the tools at the end of the installation script. That is, answer **n** to the following question:

```
Shall I compile the tools for you?
```

Step 3. Edit the **src/inc/config.h** to add the following lines :

```
#define have_iovec
#define need_index
#define need_rindex

#include <port/libannex.h>
```

Step 4. Change the following line in the **src/make.config** file from:

```
CFLAG=
```

to:

```
CFLAG=-D_SYSV
```

Step 5. Compile and install the tools:

```
cd src
make
make install
```

HP 9000 (HP-UX)

The following set of instructions were tested using the Hewlett Packard HP9000 hardware platform and the HP-UX Rev 3.0 operating system software. These instructions are not guaranteed to work with all HP systems, but they should provide guidelines for other HP machines and software releases.

To install the Annex distribution software on the above system, perform the following steps:

Step 1. Run the **install–annex** script. Answer the following questions as noted:

```
Select the most appropriate machine type
```

Select: `Generic (no special code needed)`

```
Select the most appropriate OS
```

Select: `System V Unix (& UMAX V)`

```
What kind of Network Software does your OS have?
```

Select: `4.[234]BSD or compatible`

Step 2. Do not compile the tools at the end of the installation script. That is, answer **n** to the following question:

```
Shall I compile the tools for you?
```

Step 3. The header file **/usr/include/sys/socket.h** defines the structure **msghdr**, which is not used by the HP-UX package. However, the compiler flags the structure as multiply defined, because the Annex software defines it. To overcome the compiler error, copy:

```
/usr/include/sys/socket.h
```

to:

```
src/inc/sys/socket.h
```

Then edit the copy to remove (using **#ifdef**) the definition of the **msghdr** structure.

Step 4. Add the following lines to the **src/inc/config.h** file:

```
#define need_recvmsg
#define recvmsg xy_recvmsg
#define need_sendmsg
#define sendmsg xy_sendmsg
```

Step 5. Edit **src/libannex/sendrecv.c** to change the names of all occurrences of the following routines:

from:	_sendmsg	to:	xy_sendmsg
from:	_sendto	to:	xy_sendto
from:	_recvmsg	to:	xy_recvmsg
from:	_recvfrom	to:	xy_recvfrom

Step 6. There is a conflict between the names of the routines in the HP-UX TCP/IP library and some of those in the Annex library (which are not used in the HP-UX version). It is therefore necessary to change the names of the Annex versions to resolve this conflict. To do this, edit the **src/libannex/sendrecv.c** file to change the following line from:

```
_recvfrom(s,msg,len,flags,from,fromlen)
```

to:

```
_XYrecvfrom(s,msg,len,flags,from,fromlen)
```

and change the line:

```
_sendto(s,msg,len,flags,to,tolen)
```

to:

```
_XYsendto(s,msg,len,flags,to,tolen)
```

Step 7. Edit **src/make.config** to change the following line from:

```
LIBANX_C = erpc_subr.c srpc.c
LIBANX_O = erpc_subr.o srpc.o
```

to:

```
LIBANX_C = erpc_subr.c srpc.c sendrecv.c
LIBANX_O = erpc_subr.o srpc.o sendrecv.o
```

Step 8. Compile and install the tools:

```
cd src
make
make install
```

MIPS 1000 and Silicon Graphics

The following set of instructions were tested using the MIPS1000 hardware platform and the System V release 3_10C operating system software. These instructions are not guaranteed to work with all MIPS systems, but they should provide guidelines for other MIPS machines and software releases.

To install the Annex distribution software on the above system, perform the following steps:

Step 1. Run the **install-annex** script. Answer the following questions as noted:

```
Select the most appropriate machine type
```

Select: `Generic (no special code needed)`

```
Select the most appropriate OS
```

Select: `System V Unix (& UMAX V)`

```
What kind of Network Software does your OS have?
```

Select: `4.[234]BSD or compatible`

```
Does your system have any auxiliary libraries for the network code?[n]
```

Select: `y`

```
Enter any additional loader options for the network libraries
```

Enter: `-lbsd`

```
Does your system have an alternate include directory for network code[n]?
```

Select: `y`

```
Where are the network include files located?
```

Enter: `/usr/include/bsd`

Step 2. Complete the script but do not compile the tools at the end of the installation script. Answer **n** to the following question:

```
Shall I compile the tools for you?
```

Step 3. Change the following line in the **src/make.config** file from:

```
CFLAGS=
```

to:

```
CFLAGS=-signed
```

Note: There may already be other flags defined for CFLAGS. In this case, do not delete these flags; just add *–signed* to CFLAGS.

Step 4. Compile and install the tools:

```
cd src
make
make install
```

Motorola

The following set of instructions were tested using the Motorola Delta Series hardware platform and the System V/68 and System V/88 operating system software. These instructions are not guaranteed to work with all Motorola systems, but these instructions should provide guidelines for other Motorola machines and software releases.

Step 1. Run the **install-annex** script. Answer the following questions as noted:

```
Select the most appropriate machine type
```

Select: `Generic (no special code needed)`

```
Select the most appropriate OS
```

Select: `System V Unix (& UMAX V)`

```
What kind of Network Software does your OS have?
```

Select: `4.[23]BSD UNIX (& UMAX BSD)`

Step 2. Do not compile the tools at the end of the installation script. That is, answer **n** to the following question:

```
Shall I compile the tools for you?
```

Step 3. Edit **src/inc/config.h** to add the following lines:

```
#define need_sendmsg
#define sendmsg xy_sendmsg
#define need_recvmsg
#define recvmsg xy_recvmsg
#define have_msghdr
#define need_sendto
#define sendto xy_sendto
#define NEED_O_SENDTO
```

Step 4. Edit **src/libannex/sendrecv.c** to change the names of all occurrences of the following routines:

from:	`_sendmsg`	to:	`xy_sendmsg`
from:	`_sendto`	to:	`xy_sendto`
from:	`_recvmsg`	to:	`xy_recvmsg`
from:	`_recvfrom`	to:	`xy_recvfrom`

Step 5. Edit **src/libannex/sendrecv.c** changing the following **xy_sendto()** routine code segment from:

```
#ifdef EXOS
            nwritten = send(s, to, residmsg, nremain);
#else
            /* XXX to who? */
            if (debug)
                        fprintf(stderr, "NOT IMPLEMENTED!\n");
            nwritten = send(s, residmsg, nremain, flags);
#endif
```

to:

```
#ifdef EXOS
            nwritten = send(s, to, residmsg, nremain);
#else
#ifdef NEED_0_SENDTO
#undef sendto
            nwritten = sendto(s,msg,len,flags,to,tolen);
#define sendto _sendto
#else
            /* XXX to who? */
            if (debug)
                        fprintf(stderr, "NOT IMPLEMENTED!\n");
            nwritten = send(s, residmsg, nremain, flags);
#endif
#endif
```

Step 6. Edit **src/make.config** to change the following line from:

```
LIBANX_C = erpc_subr.c srpc.c
LIBANX_O = erpc_subr.o srpc.o
```

to:

```
LIBANX_C = erpc_subr.c srpc.c sendrecv.c
LIBANX_O = erpc_subr.o srpc.o sendrecv.o
```

Step 7. Compile and install the tools:

```
cd src
make
make install
```

SCO TCP/IP

The following set of instructions were tested using a Compaq 386/20 portable and the Xenix System V Version 2.3.2 (for Intel 80386) operating system software. These instructions are not guaranteed to work with all TCP/IP packages, refer to the accompanying manuals for the package you are using for more information.

Now, install the Annex distribution software by perform the following steps:

Step 1. Run the **install-annex** script. Answer the following questions as noted:

> `Select the most appropriate machine type`

Select: `Generic (no special code needed)`

> `Select the most appropriate OS`

Select: `System V Unix (& UMAX V)` (Do **not** use XENIX System V)

> `What kind of Network Software does your OS have?`

Select: `4.[234]BSD or compatible`

> `Does your system have any auxiliary libraries for the network code?[n]`

Select: `y`

> `Enter any additional loader options for the network libraries`

Enter: `-lsocket`

> `Does your system have an alternate include directory for network code[n]?`

Select: `n`

Step 2. Do not compile the tools at the end of the installation script. Answer **n** to the following question:

> `Shall I compile the tools for you?`

Step 3. The SCO TCP/IP package requires special compiler flags. To use these flags, change the following line in the **src/make.config** file from:

> `CFLAG= -DINETD -DGENERIC -DSYS_V  -I../inc`

to:

> `CFLAG= -DINETD -DGENERIC -DSYS_V  -DLAI_TCP -Di386 -pack -I../inc`

Step 4. Normally, SCO Xenix does not supply a **crypt** routine. The Annex security, Access Control Protocol (ACP), requires the routine to encrypt passwords. You can create the following dummy routine:

```
char *crypt(key,salt)
char *key;
char *salt;
{
        statuc char         buff[200];

        strcpy(buff,key);
        return buff;
}
```

Place this routine in the file **src/libannex/crypt.c.** In the file **src/make.config** change the following lines from:

```
LIBANX_C = erpc_subr.c srpc.c
LIBANX_O = erpc_subr.o srpc.o
```

to:

```
LIBANX_C = erpc_subr.c srpc.c crypt.c
LIBANX_O = erpc_subr.o srpc.o crypt.o
```

Step 5. When using the **–pack** option with the Xenix software, the Annex's **fprintf** routine does not work. To fix this, create a routine called **_fprintf** and have it call the original **fprintf** routine. The **_fprintf** routine should be:

```
#include   <stdio.h>

_fprintf(chan,f,ala,a2,a3,a4,a5,a6,a7,a8,a9,
          a10,a11,a12,a13,a14,a15,a16,a17,a18,a19,a20)
unsigned long  chan;
{
        unsigned long  newchan;

        newchan == chan;
        if(chan == ((unsigned int)stdout - 2)) newchan = stdout;
        if(chan == ((unsigned int)stderr -4 )) newchan = stderr;

        return fprintf(newchan,f,ala,a2,a3,a4,a5,a6,a7,a8,a9,a10,
        a11,a12,a13,a14,a15,a16,a17,a18,a19,a20);
}
```

Step 6. Place the above routine in the **src/libannex/fprintf.c** file. Change the following lines in the **src/make.config** file from:

```
LIBANX_C=erpc_subr.c srpc.c crypt.c
LIBANX_O=erpc_subr.o srpc.o crypt.o
```

to:

```
LIBANX_C=erpc_subr.c srpc.c crypt.c fprintf.c
LIBANX_O=erpc_subr.o srpc.o crypt.o fprintf
```

You must compile the **src/libannex/fprintf.c** file without the **–pack** flag, by editing the **src/libannex/Makefile** file. After these lines:

```
clean:
        ${RM} -f $(LIBANX_O)
        ${RM} -f libannex.a
```

add the following lines:

```
fprintf.o: fprintf.c
        ${CC} -c fprintf.c
```

Finally, you need to force the software to use the modified routine instead of the standard routine, by adding the following line to the head of the **src/inc/config.h** file:

```
#define     fprintf     _fprintf
```

Step 7. Copy the file **/usr/include/sys/socket.h** to **src/inc/sys/socket.h**. Edit this new file and change:

```
/*
 * Message header for recvmsg and sendmsg calls.
 */
struct msghdr {
    caddr_t         msg_name;           /* optional address */
    int             msg_namelen;        /* size of address */
    int             msg_base;           /* base address of data */
    int             msg_len;            /* size of data */
    caddr_t         msg_accrights;      /* access rights sent/received */
    int             msg_accrightslen;
};
```

to:

```
#ifdef          NOTDEF
/*
 * Message header for recvmsg and sendmsg calls.
 */
struct msghdr {
    caddr_t         msg_name;           /* optional address */
    int             msg_namelen;        /* size of address */
    int             msg_base;           /* base address of data */
    int             msg_len;            /* size of data */
    caddr_t         msg_accrights;      /* access rights sent/received */
    int             msg_accrightslen;
};
#endif
```

Step 8. Compile and install the tools:

```
cd src
make
make install
```

List of Files

The following files are distributed with the release media:

bfs/annex.13	920 blocks
bfs/oper.16.8025	1026 blocks
bfs/oper.16.enet	1124 blocks
bfs/oper.42.enet	1110 blocks
install–annex	75 blocks
man/cat/aprint.1	11 blocks
man/cat/erpcd.8	13 blocks
man/cat/eservices.5	4 blocks
man/cat/gateways.5	19 blocks
man/cat/mk_rotaries.8	3 blocks
man/cat/na.8	3 blocks
man/cat/rotaries.5	14 blocks
man/cat/rtelnet.8	13 blocks
man/cat/timserver.8	3 blocks
man/src/aprint.1	9 blocks
man/src/erpcd.8	10 blocks
man/src/eservices.5	4 blocks
man/src/gateways.5	13 blocks
man/src/mk_rotaries.8	2 blocks
man/src/na.8	3 blocks
man/src/rotaries.5	10 blocks
man/src/rtelnet.8	9 blocks
man/src/timserver.8	3 blocks
release	1 blocks
src/aprint/Makefile	4 blocks
src/aprint/aprint.c	28 blocks
src/aprint/aprint.h	4 blocks
src/aprint/parsecap.c	15 blocks
src/diffs/4.2/lpr/common.c.diff	3 blocks
src/diffs/4.2/lpr/displayq.c.diff	3 blocks
src/diffs/4.2/lpr/lp.h.diff	3 blocks
src/diffs/4.2/lpr/printjob.c.diff	9 blocks
src/diffs/4.2/lpr/rmjob.c.diff	3 blocks
src/diffs/4.2/rlogind/rlogind.c.diff	3 blocks
src/diffs/4.2/telnetd/telnetd.c.diff	8 blocks
src/diffs/4.3/lpr/common.c.diff	3 blocks
src/diffs/4.3/lpr/displayq.c.diff	3 blocks
src/diffs/4.3/lpr/lp.h.diff	3 blocks

src/diffs/4.3/lpr/printjob.c.diff	7 blocks
src/diffs/4.3/lpr/rmjob.c.diff	3 blocks
src/erpcd/Makefile	6 blocks
src/erpcd/acp.c	19 blocks
src/erpcd/acp.h	13 blocks
src/erpcd/acp_lib.c	27 blocks
src/erpcd/acp_policy.c	73 blocks
src/erpcd/acp_policy.h	10 blocks
src/erpcd/acp_rpc.c	19 blocks
src/erpcd/annex-initd	3 blocks
src/erpcd/bfs.c	66 blocks
src/erpcd/ch_passwd.c	12 blocks
src/erpcd/erpcd.c	33 blocks
src/erpcd/eservices	1 blocks
src/erpcd/getacppw.c	6 blocks
src/erpcd/policy.doc	44 blocks
src/ien-116/Makefile	3 blocks
src/ien-116/ien116d.c	11 blocks
src/inc/config.h	1 blocks
src/inc/courier/courier.h	9 blocks
src/inc/erpc/bfs.h	7 blocks
src/inc/erpc/erpc.h	8 blocks
src/inc/erpc/netadmp.h	28 blocks
src/inc/port/CMC.h	4 blocks
src/inc/port/EXOS.h	5 blocks
src/inc/port/SLIP.h	10 blocks
src/inc/port/SYS_V.h	3 blocks
src/inc/port/libannex.h	8 blocks
src/inc/port/xuio.h	5 blocks
src/inc/slip/BSDslip.h	7 blocks
src/inc/slip/slip_system.h	8 blocks
src/inc/slip/slip_user.h	6 blocks
src/inc/syslog.h	7 blocks
src/libannex/Makefile	5 blocks
src/libannex/api_if.c	53 blocks
src/libannex/api_if.h	6 blocks
src/libannex/bcopy.c	1 blocks
src/libannex/contents	1 blocks
src/libannex/dup2.c	3 blocks
src/libannex/erpc_subr.c	36 blocks
src/libannex/hostname.c	12 blocks
src/libannex/in_cksum.c	8 blocks
src/libannex/inet.c	14 blocks

src/libannex/ip_layer.c	11 blocks
src/libannex/sendrecv.c	14 blocks
src/libannex/servname.c	6 blocks
src/libannex/slip_layer.c	16 blocks
src/libannex/socket_layer.c	30 blocks
src/libannex/srpc.c	33 blocks
src/libannex/srpc.h	5 blocks
src/libannex/udp_layer.c	12 blocks
src/mk_rotaries/Makefile	4 blocks
src/mk_rotaries/example	1 blocks
src/mk_rotaries/mk_rotaries.c	40 blocks
src/mk_rotaries/rotary_rec.h	8 blocks
src/na/Makefile	5 blocks
src/na/cmd.c	161 blocks
src/na/conv.c	74 blocks
src/na/conv2.c	28 blocks
src/na/do.c	70 blocks
src/na/help.h	69 blocks
src/na/iftype.h	4 blocks
src/na/main.c	16 blocks
src/na/na.h	34 blocks
src/na/names.h	4 blocks
src/na/oem.h	1 blocks
src/na/parse.c	113 blocks
src/na/server.h	3 blocks
src/na/sub.c	22 blocks
src/na/vers.c	1 blocks
src/netadm/Makefile	10 blocks
src/netadm/boot.c	13 blocks
src/netadm/broadcast.c	8 blocks
src/netadm/dumpboot.c	7 blocks
src/netadm/get_anx_opt.c	5 blocks
src/netadm/get_anx_rev.c	6 blocks
src/netadm/get_da_param.c	7 blocks
src/netadm/get_da_stat.c	7 blocks
src/netadm/get_ln_param.c	8 blocks
src/netadm/get_ln_stat.c	8 blocks
src/netadm/get_port_cnt.c	6 blocks
src/netadm/get_port_eib.c	5 blocks
src/netadm/get_tr_cnt.c	5 blocks
src/netadm/init_socket.c	7 blocks
src/netadm/netadm.h	6 blocks
src/netadm/netadm_err.h	7 blocks
src/netadm/read_memory.c	9 blocks

src/netadm/reset_all.c	6 blocks
src/netadm/reset_annex.c	6 blocks
src/netadm/reset_line.c	7 blocks
src/netadm/return_param.c	13 blocks
src/netadm/rpc.c	38 blocks
src/netadm/set_da_param.c	13 blocks
src/netadm/set_inet_add.c	7 blocks
src/netadm/set_ln_param.c	11 blocks
src/netadm/verify_resp.c	7 blocks
src/rtelnet/Makefile	4 blocks
src/rtelnet/rtelnet.c	47 blocks
src/rtelnet/rtelnet.smns	60 blocks
src/slipd/Makefile	4 blocks
src/slipd/slipcfg	1 blocks
src/slipd/slipd.c	30 blocks
src/slipd/stty.c	11 blocks
src/snmp/annex-mib.txt	18 blocks
src/snmp/mib.txt	56 blocks
src/timserver/Makefile	4 blocks
src/timserver/timserver.c	11 blocks