HXEASY
Async Exerciser
User's Guide

James D. Miles

September 14, 1993

This is a preliminary version.
Forward comments to jmiles@jmiles.

IBM Internal Use Only

Contents

1 Overview

HXEASY is an HTX exerciser that conforms to the requirements of "HTX Hardware Exerciser Programmer's Guide." It is intended to exercise any standard UNIX tty port. Ports may be wrapped on themselves. Even better, the ports may be connected for interoperabilty between like and unlike types. It is designed to support the testing of asynchronous ports by novice testers while offering configurability for special tests that are likely to be needed.

The following are the key features of this exerciser:

- Configurability through the use of a rules file.
- Automatic configuration of the hardware flow control disciplines (xon, dtr, and rts).
- Automatic configuration of the POSIX disciplines.
- Automatic identification of the port-to-port connections eliminating complex setup procedures.
- Robust recovery from errors.
- Extensive communication and synchronization between processes that is used to identify abnormal/unexpected events.
- Support for debug and trace.

Note: This document is written for AIX version 4.x. When paths are different for AIX version 3.x, it will be noted as a footnote.

2 Input files

The following subsections refer to fields defined in Appendix B and to exit codes defined in Appendix A. If you are not familiar with rules file fields, read the above appendix sections first. The rules file "rules/reg/hxeasy/128port"[1] contains an example utilizing multiple stanzas. The rules file "rules/reg/hxeasy/termrule" contains an example utilizing the SPECIAL parameter.

[1] For AIX version 3.x the path is "rules/reg/asy."

2.1 Rules File

The rules file name is always specified as an input variable on the command line (see Section 4).

The *rules file* is used to specify the various customizable parameters. These parameters consist of 2 types: hardware parameters which can only be set once and software parameters which are reset with each stanza of the rules file. The *hardware parameters* consist of HW_PACING and 128P.

The rules file consist of stanza(s) separated by a blank line. Each stanza is made up of lines consisting of definitions of various parameters.

2.2 Pattern File

The *pattern file* specifies the stream of characters that is used to initialize the write buffer. The size of the file can be less than or greater than the write buffer size (see NUM_CHARS). The pattern is either repeated or truncated till it fits the write buffer. The pattern file is specified as a field PATTERN_ID in the rules file.

2.3 Input File Interactions

2.3.1 Equivalent Rules Files

If setting up rules files for cables, care must be taken to make sure that the rules files used for each port pair are equivalent. *Equivalent rules files* will always lead to the read port expecting to read what has been written by the write port. This would mean that parameters such as NUM_OPER, BUFSIZE, CBAUD, CHSIZE, CSTOPB, PARODD, PATTERN_ID, IXON, IXOFF, DIRECTION, and HW_PACING must be the same. IOCTL_SLEEP, RULE_ID, CLOCAL, and IHOG might be different depending on the purpose of the test.

2.3.2 Multiple Stanzas

If multiple stanzas are used, hardware parameters (HW_PACING and 128P) may only be defined once. If these parameters are defined multiple times, the last value is used.

2.3.3 Half-duplex on a Wrap Plug

If half-duplex is specified on a wrap plug, no testing will be performed. There will only be an error message if all BAUD rates are specified at half-duplex (see exit(13) and CBAUD).

2.3.4 BUFSIZE, Full-duplex, and Deadlocks

When testing full-duplex, each process writes BUFSIZE characters before reading. If the read side isn't able to buffer BUFSIZE characters, the write port won't be able to write all BUFSIZE characters — it will time out and terminate with an exit(18). This isn't a problem when testing half-duplex since there is always a reader.

2.3.5 Pattern File Name

Since all information in the rules file is converted to upper-case when read, it is not possible to locate a pattern file whose name contains any lower-case letters.

2.3.6 IOCTL_SLEEP Parameter Set Too Low

It is necessary to sleep after changing port characteristics, flushing data, etc. There is no indication available to the exerciser when a particular function has completed. This problem is handled by inserting sleeps after each call to the driver that changes a port's state. This sleep time must be worst case to guarantee that the function/command has been executed and not just buffered.

The default sleep time used by the exerciser has been found to be good for all test loads but does result in initial setup times in the minutes. The good news is that this time is constant and does not result in longer setup times as more ports are tested.

If IOCTL_SLEEP is made too low, many symptoms can/will occur. While it is reasonable to change the sleep time to a lower value for some test, the user must assure that new problems haven't been introduced because of an inappropriate sleep time.

2.3.7 Control-S and Control-Q with IXON and IXOFF

When testing with XON/XOFF flow control, the pattern file can not contain the Control-S and Control-Q characters. The file HEX255 contains these characters and will cause the process to generate read errors since the write process will be flow-controlled.

When using XON/XOFF flow control with pattern files that contain no control characters (such as pattern file "ASCII"), 5 bit words should not be used. When the characters are masked to 5 bits, control characters will be generated.

2.3.8 NUM_OPER and Low BAUD Rates

If using a large NUM_OPER, multiple stanzas may be necessary. A NUM_OPER of 10000 may be satisfactory for 38.4K BAUD but would take too long to complete a test cycle at 50 BAUD.

3 Environmental variables

HXEASY recognizes the following environmental variables: HTXPATTERNS and HTXRULES. If HXEASY can't locate the rules file or the pattern file by the specified path search (see Section 4.1), then it will try the path specified by the appropriate environmental variable.

Example: To run HXEASY on a machine that doesn't have the HTX lpp, set the environmental variables. If the rules file is in directory $HOME/rule and the pattern file is in directory $HOME/test then set the environmental variables as follows:

```
export HTXRULES=$HOME/rule
export HTXPATTERNS=$HOME/test
```

Once these variables are set, HXEASY can be invoked as described in Section 4.

4 Starting HXEASY

If HXEASY is invoked without any arguments it responds with:

```
usage: hxeasy /dev/ttyXX [REG EMC] rule_path
  or: hxeasy [-l] [-m seconds] /dev/ttyXX  OTH  rule_path
  or: hxeasy -c
  or: hxeasy -k
```

These are the 4 different ways to call HXEASY.

The first call must be started under HTX (see Section 4.2).

The second call is the normal startup when not running under HTX (see Section 4.3).

The third call can be used to assure the cleanup of shared memory and semaphores (see Section 4.4).

The fourth call can be used to terminate all running processes when running stand-alone (see Section 4.3).

4.1 Rules File Name Path Searching

HXEASY takes the rule_path it is given and searches for the file as follows:

1. try <<rule_path>>,
   IF the file is not found AND <<rule_path>> doesn't start with a '.' or a '/' try item 2 and if necessary try item 3.
2. IF type is EMC[2]
   try ../rules/emc/asy/<<rule_path>>
   ELSE
   try ../rules/reg/asy/<<rule_path>>
3. try $HTXRULES/<<rule_path>> where $HTXRULES is an exported environmental variable as explained in Section 3.

4.2 Starting under HTX

This is the first option referred to in Section 4 and is covered in "HTX User's Guide."

4.3 Starting Stand-alone

This is the second option referred to in Section 4. This mode can be used anytime it is desired to run HXEASY outside of the HTX environment. This mode provides all of the capability present when running under HTX.

4.3.1 Stand-alone Messages

All messages are written to the screen. Messages with severe errors that would normally be written to the htxerr log are written to stderr. All other messages are written to stdout. In addition to the usual messages generated under HTX, the exerciser writes additional information messages to stdout.

[2] This item is included for compatability with AIX version 3.x only.

4.3.2 Stand-alone Options

The default mode is for the exerciser to run through the rules file once and terminate. If it is desired to test in an endless loop then the **-l option** should be used.

The default time-out value is 180 seconds. If more than a few processes are running this may not be long enough. The default time used under HTX is max_run_tm=300 and may be more appropriate. To set the time-out value use the **-m option** — i.e. **-m 300**. If stand-alone cable processes are being started, this timeout value would also be the maximum amount of time available to get both processes started without the first process exiting with an error.

4.3.3 Killing Stand-alone Exercisers

The easiest method of stopping the exercisers is the fourth option in Section 4. "hxeasy -k" will kill all exercisers that are running in the stand-alone environment. This method cannot be used under the HTX environment.

If it is desired to only kill a few exercisers, it is recommended that each be sent a SIGTERM signal. This will allow the exerciser to cleanup properly. If the exerciser being killed by a SIGTERM is testing through a cable, the exerciser will automatically send a SIGTERM signal to the other exerciser. This can be accomplished as follows:

```
/home% export HTXRULES=/u/jmiles/test
/home% export HTXPATTERNS=/u/jmiles/test
/home% hxeasy /dev/tty2 OTH fast


/dev/tty2 Mar 15 12:06:08 1993 err=00200000 sev=7...
/dev/tty2 OTH /u/jmiles/test/fast


/dev/tty2 Mar 15 12:06:08 1993 err=00200000 sev=7...
Starting hardware configuration phase...
```

```
/home% ps -ef | grep tty2
build 13600 16085 4 12:06:47 pts/1 0:00 grep tty2
build 19998     1 0 12:06:08 pts/1 0:00 hxeasy /dev/tty2

/home% kill -TERM /dev/tty2 19998
/dev/tty2 Mar 15 12:06:59 1993 err=00200000 sev=7...
SIGTERM received.
/dev/tty2 Mar 15 12:06:59 1993 err=00200000 sev=7...
Exiting with exit code 0 ...

/home%
```

4.4 Memory Cleanup

Because of the way UNIX processes can be killed and the fact that the HTX supervisor sends SIGKILL signals, it is possible that the shared memory and semaphores do not get removed when the exercisers exit. Normally this should not be a problem in that the first exerciser on a restart will identify and correct the problem. If it is desired to make sure that all shared memory and semaphores have been removed, then "hxeasy -c" will do the job.

5 Premature Exits

If the an exerciser process terminates prematurely, note the exit number. If running stand-alone, a formatted message containing the exit number will be sent to stderr. If running under HTX it will momentarily be sent to the screen but it will also be recorded in the htxerr log. While looking up the exit number, be sure to note what was happening prior to the exit. This includes the error messages for this port and any connected cable port. In most cases this information will be sufficient to resolve the problem. If not, possible causes are listed in Appendix A.

6 Design/Functionality of HXEASY

This section contains an overview of the basic design. For more a more detailed look, review the source code.

6.1 Hardware Configuration Phase

The hardware configuration phase consists of:

1. Non-blocking open of port.
2. Setup the discipline stack with the type of pacing desired.
3. Any hardware specific setup that is required.
4. Close of port.

6.2 Software Configuration Phase

The software configuration phase consists of:

1. Non-Blocking open of port.
2. Termios parameters are set to default values.
3. Close of port.

6.3 Port Identification Phase

Port identification phase consists of:

1. Set/make-sure exerciser process is session leader for port.
2. Blocking open of port.

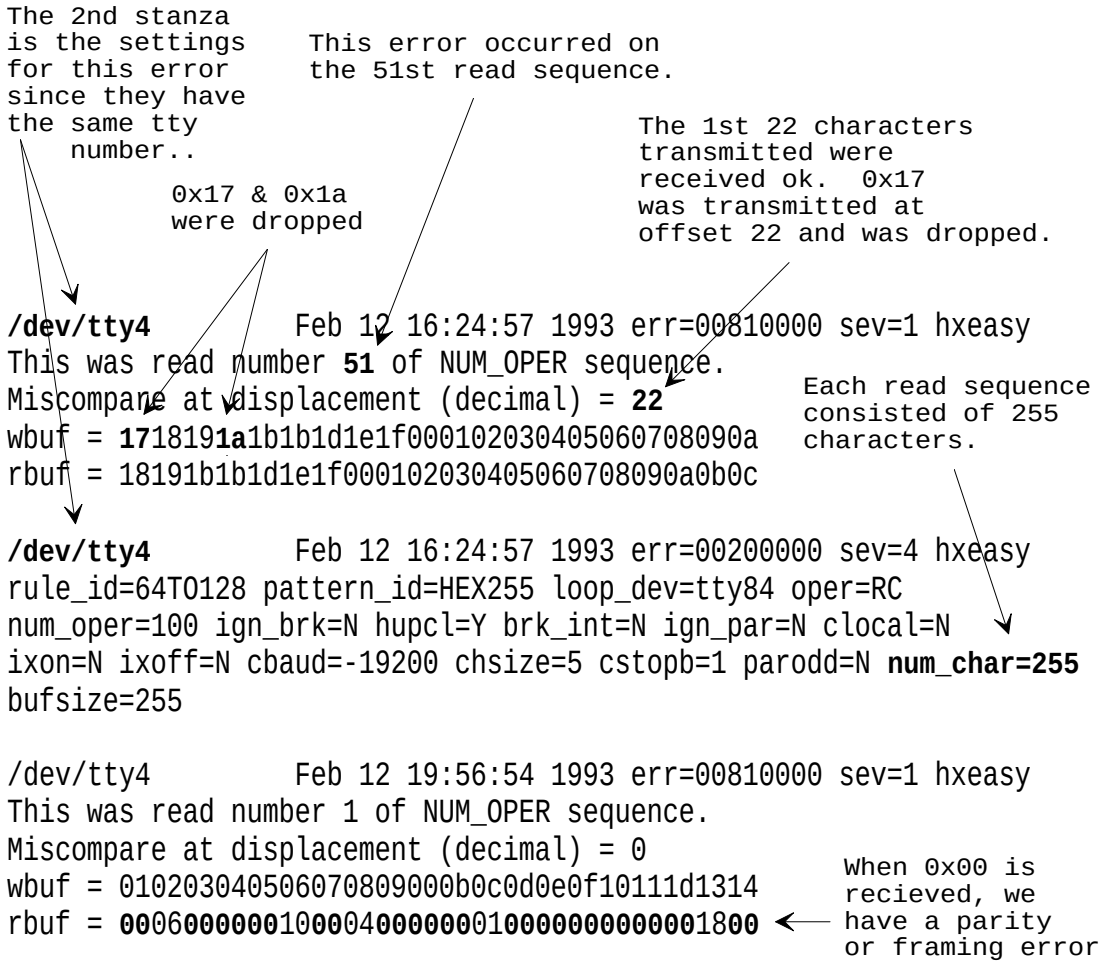
3. Send breaks out on port and look for a response back from other side. This is a crude sync mechanism and the first check that signals work.
4. Write out the port the following: hostname, port tty name, process ID, and shared memory IDs. Read in the same information. This can be used to identify if we are a wrap plug, a cable, or a remote cable.
5. Write configuration information to /tmp/tty.doc

6.4 Test Phase

Test phase consists of:

1. Setup all termios parameters to the values specified in the rules file.
2. Allocate memory for the pattern buffers.
3. Loop on BAUD rate, character size, stop bits, and parity.
4. Flush all port buffers and set parameters.
5. Determine operation to perform, i.e. full-duplex, half-duplex read, or half-duplex write.
6. Perform I/O operation
7. Recover if errors occurred.
8. Goto item 3

6.5 How to Read Miscompare data



```
Notice that the characters not read on read sequence 59
are read on read sequence 60.  This is an indication that
the time-out value is too short.  If the time-out value
is increased, this error will disappear.

/dev/tty4         Feb 12 16:24:57 1993 err=00810000 sev=1 hxeasy
This was read number 59 of NUM_OPER sequence.
Miscompare at displacement (decimal) = 22
wbuf = 1718191a1b1b1d1e1f000102030405060708090a
rbuf = 1718191a2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e

/dev/tty4         Feb 12 16:25:57 1993 err=00810000 sev=1 hxeasy
This was read number 60 of NUM_OPER sequence.
Miscompare at displacement (decimal) = 0
wbuf = 0b0c0d0e0f101112131415161718191a1b1c1d1e
rbuf = 1b1b1d1e1f000102030405060708090a0b0c0d0e
```

The character '2e' is a placeholder for nothing when the read times out.

7 Limitations

Host-to-host testing has not been implemented yet.

8 Using Debuggers

9 Using Trace Functions

A Exit Codes

The following has been pulled from the header file "exit.h." It contains pointers to possible solutions when an exerciser terminates prematurely. To locate the proper paragraph match the exit number with the #define number.

```
/* When the htxerr file is referred to below and you are testing in          */
/* standalone mode, we are refering to stderr.                               */

/*****************************************************************************/
/* This is the most likely exit.  A shutdown of the supervisor using the     */
/* fast or slow shutdown method will generate this exit code.  If a cable    */
/* process exits with an error, it will send a SIGTERM signal to the other   */
/* cable process terminating it.                                             */
#define EX_SIGTERM      0
/*****************************************************************************/

/*****************************************************************************/
/* hxeasy with type=OTH will terminate with this return code if no errors.   */
#define EX_NORMAL       0
/*****************************************************************************/

/*****************************************************************************/
/* An invalid number of arguments were passed at the invocation of "hxeasy". */
/* Correct invocation of hxeasy can be displayed by executing hxeasy without */
/* any arguments.                                                            */
#define EX_INV_ARGV     1
/*****************************************************************************/

/*****************************************************************************/
/* There was an error reading the rules file.                                */
/* Specific information should have been printed in the htxerr log file.     */
#define EX_RULEREAD1    2
/*****************************************************************************/

/*****************************************************************************/
/* The exerciser may have been unable to generate a valid pattern file name. */
/* Check the following:                                                      */
/* htxerr message  - There are more than a few failure modes and this will   */
/*                   help isolate problem.                                   */
/* The file name   - Specified in rule file and should be a valid file in the */
/*                   pattern directory. The name must be all upper-case      */
/*                   letters.  If the file name contains some lower-case     */
/*                   letters, rename the file.                               */
/* The path      1 - Path will be assumed to be part of the file name.       */
```

```
/*               2 - If file name begins with a '.' or a '/'. no other paths  */
/*                   will be tried (This is hard to do with a limit of 8      */
/*                   characters for the filename.)                            */
/*            or 2 - Next a hardwired path '../pattern' will be tried.        */
/*               3 - Last, path HTXPATTERNS from environment will be tried.   */
#define EX_RULEREAD2    3
/*****************************************************************************/

/*****************************************************************************/
/* While trying to allocate memory for the pattern buffers, the system       */
/* returned an error.  Additional information should be in the htxerr log.   */
/* The total memory needed is NUM_CHARS(from rules file) * 3 for each        */
/* exerciser.  If all exercisers are using the same rules file, then the     */
/* total memory needed is NUM_CHARS * number_of_exercisers * 3.              */
#define EX_FMALLOC1     4
/* This is an error during reallocation of buffer size.  There are at least  */
/* two stanzas in the rules file with different NUM_CHAR sizes.              */
#define EX_FMALLOC2     5
/*****************************************************************************/

/*****************************************************************************/
/* An error was generated while trying to determine the type of port         */
/* connected (wrap plug, cable, or remote cable).                            */
/* Since this is the first test that actually reads and writes, check common */
/* setup problems first.   Bad cables, port that is "Defined", port not      */
/* connected, using an mdt file generated for a different configuration, etc.*/
/* A probable cause is a device that is not responding in the way documented */
/* in the POSIX 1003 manual.  Since this is not a stressful test and is      */
/* highly redundant, this is potentially a serious error.  Contact the person*/
/* responsible for supporting this exerciser if the cause can't be determined.*/
/* If testing with cables, try starting this port(s) on a wrap plug.         */
#define EX_WRAPREAD     6
/*****************************************************************************/

/*****************************************************************************/
/* Either the paging volume is too small, not enough memory, or too many     */
/* processes are running on this machine.                                    */
#define EX_SIGDANGER    7
/*****************************************************************************/

/*****************************************************************************/
/* This is an impossible exit code.  If hxeasy exits with this code, call the*/
/* person responible for supporting this exerciser.                          */
#define EX_FASTPATH     8
/*****************************************************************************/

/*****************************************************************************/
```

```
/* We timed out before shared memory was initialized.  Either the shared      */
/* memory routines had errors or the time-out interval MIN_WAIT is too short. */
#define EX_SHMWAIT      9
/*****************************************************************************/

/*****************************************************************************/
/* This is a normal exit after executing the command "hxeasy -c".            */
/* This command cleans up after an abnormal termination of the hxeasy process.*/
#define EX_CLEANUP      10
/*****************************************************************************/

/*****************************************************************************/
/* This code can only happen from a programming error.                       */
#define EX_ARG          12
/*****************************************************************************/

/*****************************************************************************/
/* This is an error in setting up the rules file.  You have specified all    */
/* test permutations at half-duplex on a port with a wrap-plug.  Since       */
/* half-duplex can't be run on a wrap-plug, there is nothing to do but       */
/* exit and report the condition.  If 3 half-duplex test and one full-duplex  */
/* test are specified, the test will continue for the one full-duplex test.  */
/* In that case no error message is generated.                              */
#define EX_BAUD         13
/*****************************************************************************/

/*****************************************************************************/
/* Read returned EIO - I/O error occurred while reading from the port.       */
#define EX_REIO         14
/*****************************************************************************/

/*****************************************************************************/
/* Write returned EIO - I/O error occurred while writing to the port.        */
#define EX_WEIO         15
/*****************************************************************************/

/*****************************************************************************/
/* The write process was unable to write any characters and no data should be */
/* in any system write buffers.  This implies that the read port should not  */
/* be in a flow controlled state.  It may mean that there is a problem on the */
/* read port.                                                                */
#define EX_WRITEZERO1   16
/*****************************************************************************/

/*****************************************************************************/
/* The write timed-out after not writing any characters.                     */
#define EX_WRITEZERO2   17
```

```
/*****************************************************************************/

/*****************************************************************************/
/* The write process timed-out without writing any characters.  But it had   */
/* previously written some characters during this write sequence. If this was */
/* the first test or if the BUFSIZE had just increased from a previous rules  */
/* stanza, check the following:                                               */
/*    - If it was a full-duplex test, BUFSIZE may be larger than the amount   */
/*      of characters that can be buffered by the system.  Reduce BUFSIZE     */
/*      or increase IHOG.  The IHOG value should only be increased for cases  */
/*      like interoperabilty between 8 port, 16 port to a concentrator that   */
/*      has the ability to buffer large amounts of data.  In this case        */
/*      increase the IHOG value on the 8 port or 16 port only.                */
#define EX_WRITEPART    18
/*****************************************************************************/

/*****************************************************************************/
/* If any of these exits occur, be sure that the message in the htxerr log is */
/* saved for the person responsible for maintaining the exerciser.            */
/* If a series of these messages happened i.e. on more than one port being    */
/* tested, then it is likely that somehow the semaphores/shared memory were   */
/* removed while a test was in progress i.e. "hxeasy -c" was executed.        */
#define EX_SEMDWN1      20
#define EX_SEMDWN2      21
#define EX_SEMDWN3      38
#define EX_SEMCREAT1    22
#define EX_SEMCREAT2    23
#define EX_SEMCREAT3    120
#define EX_SEMUP1       24
#define EX_SEMUP2       25
#define EX_SEMINIT1     26
#define EX_SEMINIT2     27
#define EX_SEMGET       36
/*****************************************************************************/

/*****************************************************************************/
/* If the HTX supervisor hasn't been changed, then there is a problem with    */
/* the setpgrp() function call.  Save the message in the htxerr log.  Try     */
/* running the failing port in standalone mode to rule out HTX supervisor     */
/* problems.                                                                  */
#define EX_SPGRP        29
/*****************************************************************************/

/*****************************************************************************/
/* You have tried to run more that the maximum allowable hxeasy processes or  */
/* shared memory has not been cleaned up between runs.  HXEASY should         */
/* eliminate the second possability.  If you need to run more exercisers than */
```

```
/* is allowed (see message in htxerr log) request the exerciser to be         */
/* recompiled with a larger EXER_MAX in global.h                              */
#define EX_MAXEXER      30
/****************************************************************************/

/****************************************************************************/
/* Save htxerr log for analysis.                                              */
#define EX_SHMSRCH      28
#define EX_SHMGET       31
#define EX_SHMAT1       32
#define EX_SHMAT2       110
#define EX_SHMAT3       111
#define EX_SHMCTL       33
#define EX_DGERR        34
#define EX_DSERR        35
/****************************************************************************/

/****************************************************************************/
/* You have attempted to connect ports on different host.  This version       */
/* doesn't support multiple host.                                             */
#define EX_REMOTE       37
/****************************************************************************/

/****************************************************************************/
/* The exerciser attempted to do a tcgetattr() call.  The htxerr log should   */
/* provide additional information.  If it is an interrupted system call then  */
/* the timeout value may be too low.                                          */
#define EX_TCGET        40
/****************************************************************************/

/****************************************************************************/
/* The exerciser was attempting to do a close() on a port.  "Close()" will    */
/* hang and return an interuppted system call if the port isn't able to flush */
/* its data.  Check time-out values being used.  Additional information s/b   */
/* in the htxerr log.  If possible also record the status of the control      */
/* lines on the port and what it is connected to.                             */
#define EX_CLOSE        41
/****************************************************************************/

/****************************************************************************/
/* If you get this exit code.  Check that port is available, etc.             */
#define EX_OPEN1        44
/****************************************************************************/

/****************************************************************************/
#define EX_OPEN2        45
/****************************************************************************/
```

```
/*****************************************************************************/
/* The port previously opened twice using a nonblocking open.  This open was  */
/* a blocking open(if CLOCAL=N).  The port will hang if DCD is not high       */
/* (normally set by other side raising DTR).  Check the following:            */
/*    - Check that the port is available.  lsdev -C may show "Available" but  */
/*      is the concentrator connected, etc?                                   */
/*    - Has the other side exited with an error.  This is would be a normal   */
/*      termination for this port and not an error.                           */
/*    - Check that the plug/cable is connected to the port properly.          */
/*    - Investigate max_run_tm.  If there is a heavy load, the ports may need */
/*      more time to sync.                                                    */
#define EX_OPEN3        46
/*****************************************************************************/

/*****************************************************************************/
/* This would be a normal exit if the SIGINT_2_hdl has been activated and     */
/* a hxeasy process is in the foreground and a ^C is given at the console.    */
/* This signal handler is currently not used.                                 */
#define  EX_SIGINT      50
/*****************************************************************************/

/*****************************************************************************/
/* This port initiated a SIGHUP on the other side by lowering its DTR line.   */
/* It timed out waiting for the loop port to respond.                         */
#define  EX_HUPWAIT1     51
/*****************************************************************************/

/*****************************************************************************/
/* This port sent a SIGHUP signal to the other side.                          */
/* It timed out waiting for the loop port to respond.                         */
#define  EX_HUPWAIT2     52
/*****************************************************************************/

/*****************************************************************************/
/* If you get this error, then the exerciser being used has been              */
/* compiled with trace capability enabled.  Check the htxerr log for          */
/* additional information.  Most likely cause is use of a bad trace argument. */
/* Check syntax of trace argument.  Is the trace dameon already running?      */
#define EX_TSTART       60
/*****************************************************************************/

/*****************************************************************************/
/* If you get these errors, then the exerciser being used has been            */
/* compiled with trace capability enabled.  Check the htxerr log for          */
/* additional information.                                                    */
#define EX_TOPEN        61
```

```
#define EX_TON          62
#define EX_TOFF         63
#define EX_TSTOP        64
/*****************************************************************************/

/*****************************************************************************/
/* This exerciser is requesting a trace argument that is different from one  */
/* previously requested.  If running standalone execute "hxeasy -c", "trcstop"*/
/* and try again.                                                            */
#define EX_TRACE_OPEN   65
/*****************************************************************************/

/*****************************************************************************/
/* Trace argument is longer than TRACE_ARG_LEN in global.h  If this is a valid*/
/* trace argument, request that TRACE_ARG_LEN be modified.                   */
#define EX_TRACE_ARG    66
/*****************************************************************************/

/*****************************************************************************/
/* These are errors while working with the stack disciplines.  Check the error*/
/* messages in the htxerr log.  Try setting and deleting the particular      */
/* disciplines using "stty add XXX" and "stty del XXX"                       */
#define EX_TXGETCD      80
#define EX_TXDELCD      81
#define EX_TXSETLD      82
#define EX_ADDXON       83
#define EX_ADDRTS       84
#define EX_ADDDTR       85

/*****************************************************************************/
/* Memory available for increasing the IHOG value are limited.  Check that   */
/* the amount of memory requested is really needed.  Check messages in htxerr */
/* log.                                                                      */
#define EX_IHOG         86
/*****************************************************************************/

/*****************************************************************************/
/* This error is caused by improper use of "hxeasy -c."  See exerciser manual */
/* on their proper use.  Generally, only use  "hxeasy -k" to stop(kill) all  */
/* standalone exercisers.  Use "hxeasy -c" ONLY when it is necessary to make  */
/* sure that no shared memory/semaphores remain after running the exerciser.  */
#define EX_CERR         90
/*****************************************************************************/

/*****************************************************************************/
/* This is a normal exit when "hxeasy -k" is typed and no standalone         */
/* exercisers are still running.                                             */
```

```
#define EX_KILL         91
/*****************************************************************************/

/*****************************************************************************/
/* Save htxerr log and any aditional information.                            */
#define EX_FORK         92
/*****************************************************************************/

/*****************************************************************************/
/* The exerciser was unable to generate a valid rule file pathname.  Check   */
/* the following:                                                            */
/* htxerr message  - There are more than a few failure modes and this will   */
/*                   help isolate problem.                                   */
/* The file name   - If specified in mdt test file, it should be a valid file */
/*                   in the rule directory.  If running standalone, the rule */
/*                   file name is specified as the 3rd argument on the       */
/*                   command line.                                           */
/* The path      1 - Path will be assumed to be part of the file name.       */
/*               2 - If file name begins with a '.' or a '/'. no other paths */
/*                   will be tried.                                          */
/*            or 2 - Next, a hardwired path '../rules/reg/asy' will be tried */
/*                   if in "REG" mode or "OTH" mode.  If in "EMC" mode,      */
/*                   '../rules/emc/asy' will be tried.                       */
/*               3 - Last, path HTXRULES from environment will be tried.     */
#define EX_FNAME        93
/*****************************************************************************/

/*****************************************************************************/
/* Old shared memory existed when processes were started.  These errors      */
/* occurred while removing the old memory.  Additional information should be */
/* in the htxerr log.                                                        */
#define EX_BACK1        100     /* Unable to get shared memory ID            */
#define EX_BACK2        101     /* Unable to get status of shared memory.    */
#define EX_BACK3        102     /* Unable to remove old shared memory.       */
/*****************************************************************************/
```

B Rules

The following has been extracted from the default rule file.

```
************************************************************************
* Rules1 file for async devices using hxeasy exerciser
************************************************************************
* HXEASY tests asynchronous ports by transferring data through
* the targeted port(s).  It can change port characteristics
* such as baud rate and parity dynamically during the test. It can
* also change test characteristics including the data packet size and
* and direction of data flow. It automatically configures the hardware
* and software disciplines of a port.  Ports that are interconnected
* with an appropriate cable are tested as easily as a port with a wrap
* plug. (Restrictions: both ports must have equivalent rules files and
* both ports must be on the same host.  Equivalent is defined as having
* the same NUM_OPER, NUM_CHARS, BUFSIZE, CBAUD, CHSIZE, CSTOPB, PARODD,
* DIRECTION, etc.  Other parameters such as IHOG may be different.)
*
* The HXEASY rule file specifies the test characteristics for a port.
* This rule file is written in the standard HTX format -- "*" denoting
* a comment, keywords or rule identifiers are followed by "=" and
* and their assigned values.  All entries in the rules file are
* converted to upper case before use.  All characteristics not specified
* in the rule file will use the default values specified below.
*
* When specifying the rule_file_name as an "hxeasy" argument, the file
* is searched for in the following order:
*    1. <<rule_file_name>> (local directory, relative or absolute path),
*    IF name doesn't start with a '.' or a '/' :
*    2. IF type is EMC
*           ../rules/emc/asy/<<rule_file_name>>
*       ELSE
*           ../rules/reg/asy/<<rule_file_name>>
*    3. $HTXRULES/<<rule_file_name>> where $HTXRULES
*          is an exported environmental variable.
*
* This algorithm is included to help clarify the relationship between
* NUM_OPER, NUM_CHARS, BUFSIZE, and IOCTL_SLEEP when testing.
*    LoopForever
*        Setup test permutation and IOCTL_SLEEP as necessary
*        DoUntil NUM_OPER cycles are complete
*            DoUntil NUM_CHARS are I/O'd
*                I/O Min(BUFSIZE, Characters left)
*            EndLoop
*        EndLoop
```

```
*    EndLoop
*
* The following list is the complete set of rule identifiers recognized
* by HXEASY:
*
* RULE_ID
* String of 8 characters or less. No default value. This string denotes
* the rule identification which can be used by the tester to identify
* the test.  This RULE_ID will be displayed by the HTX main exerciser
* and will be written to the error log with the errors stanzas.
*
* PATTERN_ID
* String of 8 upper-case characters or less. There is no default value.
* The pattern file contains data patterns that are written/read from the
* target port(s).  The pattern file will be copied multiple times or it
* will be truncated as required to fill the buffer (size specified by
* NUM_CHARS).  The pattern file is searched for in the following order:
*    1. ./<<pattern_id>>,
*    2. ../pattern/<<pattern_id>>,
*    3. $HTXPATTERNS/<<pattern_id>> where $HTXPATTERNS is an exported
*       environmental variable.
*
* NUM_OPER
* Integer value, must be greater than 0.  This represents the number of
* I/O operations to be performed at each setting of baud, word size,
* stop bits, and parity.  A large number will generally yield a more
* stressful test but will also result in taking more time at each
* permutation of the rules file settings.
*
* BUFSIZE
* Integer value, must be 1 or greater (default 10).  BUFSIZE specifies
* the write/read buffer size in bytes. This is maximum number of bytes
* read/written during each IO operation.  That is, each read/write
* performed on the ports receives/sends a maximum of BUFSIZE bytes. This
* is different from the NUM_CHARS rule which specifies the total number of
* characters to be read/written during one I/O test sequence.
*
* IOCTL_SLEEP
* Integer. Default value is 10000000 (10 seconds).
* Sleep time is given in microseconds and is performed after ioctls.
* Sleeps are required for ALL port testing.  A value less than 10 seconds
* may be used under light loads.  With the proper use of NUM_OPER,
* a smaller value of IOCTL_SLEEP should not be necessary.
*
* SPECIAL
* Sequence of up to 3 characters.  They can be any permutation of "WRC".
* This mode is available for when it is desired to write/read from a
```

```
* port without any synchronization with another port.  By using
* SPECIAL=W, CLOCAL=Y, and HW_PACING=NONE, it is possible to write
* data out of a port not connected to anything.  DO NOT ATTEMPT USING
* THIS MODE FOR NORMAL CABLE/WRAP TESTS.
*
* CLOCAL
* One character, Y|N (default is N). (Y) Yes ignore modem line condition.
* (N) No do not ignore modem line condition.
*
* IGN_BRK
* One character, value must be Y|N (default is N). (Y) Yes ignore break
* condition.  Break characters not put on input queue and thus not read.
* (N) do not ignore break condition.
*
* HUPCL
* One character, value must be Y|N (default is Y).  (Y) Disconnect when
* last process closes open port. (N) do not disconnect on last close.
*
* BRK_INT
* One character, value must be Y|N (default is N). (Y) break condition
* generates an interrupt and flushes both input and output queues.
* (N) break does not generate interrupts.  Appropriate value will be set
* in termios structure but actions on receipt of a break are not supported.
*
* IGN_PAR
* One character, value must be Y|N (default is N). (Y) Characters with
* framing and parity errors are ignored. (N) Do not ignore framing or
* parity errors.
*
* PAR_MARK
* One character, value must be Y|N (default is N). (Y) characters with
* framing or parity error are read as 3-character sequences 0377, 0, x,
* where x is the character received in error.  (N) Character with
* framing or parity error is read as the character NULL. Appropriate values
* will be set in the termios structure, however character sequences
* inserted into the read stream will be treated the same as all other
* characters by the exerciser.  It is up to the user to identify this
* marking when the exerciser marks a miscompare.
*
* IXON
* One character, value must be Y|N (default N).  (Y) enable start and stop
* output control. (N) disable start and stop output control.  Care must be
* taken to not have ^S and ^Q in the pattern file.  When using CHSIZE=5,
* the pattern file ASCII will be masked to 5 bits and ^S and ^Q characters
* will be transmitted and you get an exit code 18.  See the exerciser
* manual for additional details.
*
```

```
* IXOFF
* One character, value must be Y|N (default N). (Y) transmit start or stop
* characters when input queue nears low or high water marks.
* (N) do not transmit start and stop characters.
*
* CBAUD
* Sequence of 1 to 13 integers. Values must be  50| 75| 110| 134| 150|
* 200| 300| 600| 1200|  1800| 2400| 4800| 9600| 19200| 38400| 57600| 76800|
* 115200.  9600 is the default. The 3 high rates are for ibm7318 server only.
* A negative number represents data in  half-duplex mode.
* If a baud rate is not specified as negative, testing is full-duplex.
* The exerciser will not perform half-duplex on wrap_plugs.  If a negative
* baud rate is specified on a wrap-plug, that setting is skipped.  When
* using full-duplex, there is a danger of blocking and exiting with
* an exit code of 18.  See the exerciser manual for additional
* information.
*
* CHSIZE
* Sequence of 1 to 4 integers. Values must be  5|6|7|8 (default is 7).
* Specifies character bit sizes.
*
* CSTOPB
* Sequence of 1 to 2 integers. Values must be 1|2 (default is 2).
* Specifies 1 one stop bit or 2 two stop bits.
*
* PARODD
* Sequence of 1 to 3 characters. Values must be 'O'|'E'|'N' (default is N).
* Parity check. (O) generate parity bit for odd check. (E) generate parity
* bit for even check. (N) no parity bit generation.
*
* NUM_CHARS
* Integer value, between 1 and 32767 (default is 255). Number of characters
* to be read/written during each I/O cycle. See the entry BUFSIZE.
*
* HW_PACING
* Default is RTS line discipline.  Acceptable values are RTS, DTR, or NONE.
* If NONE is specified and IXON, IXOFF are set to 'N' (the default),
* errors can be expected due to the lack of handshaking/flow control.
* DTR line discipline only supports output flow control and thus can't be
* used for normal wrap testing.  DTR line discipline can be used for
* testing on the 128 port if DTRPACE_ON is set as explained under 128P
* below.
*
* 128P
* This is used only for 128 port ttys.  It has no effect on other async
* devices.  The default is ALTPIN_ON, FASTCOOK_ON, DTRPACE_OFF mode.
* Other acceptable values are ALTPIN_OFF, FASTCOOK_OFF, DTRPACE_ON.
```

```
* DTRPACE_ON sets dtrpace in the cxma discipline.  It is intended to be
* set to on only when checking DTR line discipline.  This will allow
* using cable test with the DTR discipline.  DTR line discipline only
* sets up dcdpace and will drop data without setting DTRPACE_ON.
*
* DIRECTION
* When testing half-duplex, the direction of the first write is from the
* low tty port number to the higher tty number.  If it is desired to
* start the writes from the opposite port, set DIRECTION to REVERSED.
* The acceptable values are DEFAULT and REVERSED (default is DEFAULT).
*
* IHOG
* IHOG value may be increase to accommodate interoperability testing.
* If IHOG is used it should be greater than 512.
*
* TRACE
* One character, value must be Y|N (default is N).  If HXEASY has been
* compiled with trace capability, then this port will be traced if
* TRACE is set equal to Y.  See the user manual for additional details.
* This field has no effect if the exerciser hasn't been compiled with
* trace capability.
*
* SLEWRATE
* This applies to the network terminal server only..... ibm7318
* Character string with value of "SLOW", "MEDIUM" or "FAST".
* Default is "FAST", becuase fast will work with all baud rates although it
* does get inefficient for lower baud rates. "SLOW" will only work up to
* 2400 buad, "MEDIUM is for 4800-38400 and "FAST" is for the higher rates.
*
*
* How to obtain high utilization on the adapter-to-concentrator link or
* a concentrator:  To obtain high throughputs use a large NUM_OPER.  There
* will be a very significant overhead in setting up a new set of parameters
* but the exerciser will perform NUM_OPER I/Os at each setting before
* setting up the next parameters.  If there are no errors while testing,
* NUM_OPER I/Os will have occurred without any sleep times inserted
* between the read and writes.
```