```
nb>help
                    COMMAND LINE INTERPRETER (CLI)
             incrs: Lists all available increments (sets of related commands)
            <incr>: Enters an increment
            <comm>: Executes a command in an increment already entered
     <incr> <comm>: Executes a command in the specified increment
       help <incr>: Lists the commands available in the specified increment
       help <comm>: Provides help for a command in an increment already entered
help <incr> <comm>: Provides help for a command in the specified increment

Adding "full" at the end of a >help command displays any detailed help
file for the specified increment or command.  >help full is equivalent to
>help <incr> full for the current increment.  The system starts up in the
>nb increment, so >help nb lists the commands that are always available.

The following are used when displaying and entering a command's parameters:
   (...)  denote alternatives for a mandatory parameter
   [...]  denote an optional parameter
       ~  skips an optional parameter
       =  tags an optional parameter
       &  obtains a symbol's value
   "..."  delimits a string with embedded blanks (treated as one parameter)
       /  precedes a comment (rest of input line is ignored)
       \  overrides any special meaning for the next character
Optional parameters must be entered in the same order that they are displayed
by >help <comm>.
nb>incrs
          nb : NodeBase Increment
          nt : NodeBase Tools and Tests
          ct : CodeTools increment
          nw : Network Increment
          sb : SessionBase Increment
          st : SessionBase Tools and Tests
        pots : POTS Increment
          sn : Service Node Increment
          an : Access Node Increment

nb>help full
help              : Provides help for an increment or command.
  [<str>]         : name of increment
  [<str>]         : name of command ('full' = all commands)
  [full]          : displays full documentation

quit              : Exits the most recent (or all) increments.
  [all]           : exits all increments

incrs             : Lists all available increments.

send              : Sends CLI output to the console or a file.
```

```
(                 : where to send CLI output
  cout            : to the console
  prev            : to the previous location
  <str>           : to the file specified
    [t|f]         : append if file already exists? (default=f)
)

read              : Reads commands from a file.
  <str>           : read input from <str>.txt

print             : Writes a string to the console.
  <str>           : the string to be written to the console

cfgparms          : Supports configuration parameters.
(                 : subcommand...
  list            : lists all configuration parameters
  expl            : explains a configuration parameter
    <str>         : name of configuration parameter
  get             : returns a configuration parameter's value
    <str>         : name of configuration parameter
  set             : sets a configuration parameter's value
    <str>         : name of configuration parameter
    <str>         : value of configuration parameter
)

symbols           : Supports symbols.
(                 : subcommand...
  list            : lists symbols
    [<str>]       : symbol's name (lists all if omitted)
  set             : sets a symbol's value
    <str>         : symbol's name
    [<str>]       : symbol's value (symbol deleted if omitted)
  assign          : sets a symbol's value to a configuration parameter's
    <str>         : symbol's name
    <str>         : name of configuration parameter
)

stats             : Supports performance statistics.
(                 : subcommand...
  groups          : lists available groups
  show            : displays statistics
    [0:255]       : group number (default=all)
    [0:65535]     : member number (group specific; default=all)
    [<str>]       : filename for output (default=console)
  rollover        : starts a new interval
    [t|f]         : end of first interval? (default=f)
)

modules           : Displays modules.
```

```
  [0:4000]        : ModuleId (default=all)
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

pools             : Displays object pools.
  [0:63]          : ObjectPoolId (default=all)
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

audit             : Controls the object pool audit.
(                 : subcommand...
  interval        : sets the audit's frequency
    (0:60)        : seconds between audits (0 = disabled)
  force           : forces the audit to run immediately
)

sched             : Provides scheduler information.
(                 : subcommand...
  show            : displays thread statistics
    [<str>]       : filename for output (default=console)
  start           : starts tracing context switches
  stop            : stops tracing context switches
  kill            : kills a thread
    (0:99)        : ThreadId
)

threads           : Counts or displays threads.
  [0:99]          : ThreadId (default=all)
  [c|b|v]         : 'c'=count 'b'=brief 'v'=verbose (default='b')

buffers           : Counts or displays message buffers.
  [c|b|v]         : 'c'=count 'b'=brief 'v'=verbose (default='b')

psignals          : Displays POSIX signals.
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

singletons        : Displays the singletons registry.
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

heaps             : Lists all heaps.

status            : Displays system statistics.

tools             : Lists available debugging tools.

set               : Controls trace tool settings.
(                 : what to set...
  tools           : trace tools: see >tools command for abbrevations
    <str>         : tools to set: string of tool abbrevations
  (               : setting...
    on            : on
```

```
    off           : off
  )
  buffsize        : size of event trace buffer
    (128:65536)   : buffer size in KBs
)

include           : Specifies what should be captured by trace tools.
(                 : what to include...
  all             : all activity
  (               : setting...
    on            : on
    off           : off
  )
  faction         : threads in a specific faction
    (0:7)         : faction
  thread          : a specific thread's activity
    (0:99)        : ThreadId
)

exclude           : Specifies what should not be captured by trace tools.
(                 : what to exclude...
  faction         : threads in a specific faction
    (0:7)         : faction
  thread          : a specific thread's activity
    (0:99)        : ThreadId
)

query             : Shows the status of trace tools.
(                 : what to query...
  buffer          : trace buffer
  tools           : trace tools
  selections      : all items included/excluded by trace tools
)

clear             : Clears the trace buffer, tools, or selections.
(                 : what to clear...
  buffer          : trace buffer
  tools           : trace tools
  selections      : all items included/excluded by trace tools
  faction         : threads in a specific faction
    (0:7)         : faction
  factions        : all included/excluded factions
  thread          : a specific thread's activity
    (0:99)        : ThreadId
  threads         : all included/excluded threads
)

start             : Starts tracing.
  [t|f]           : immediate tracing? (default=f)
```

```
stop              : Stops tracing.

save              : Saves what was captured by trace tools.
(                 : what to save...
  trace           : events captured by tools that are currently ON
    <str>         : filename for output
)

if                : Executes rest of input line if condition is true.
  <int>           : symbol for an integer (e.g. &cli.result)
(                 : relational operator...
  <               : less than
  <=              : less than or equal to
  ==              : equal to
  !=              : not equal to
  >               : greater than
  >=              : greater than or equal to
)
  <int>           : value for comparison

delay             : Pauses before executing the next command.
  (0:90)          : time (secs)

display           : Displays an object.
  <hex>           : pointer to an Object
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

dump              : Displays memory in hex.
  <hex>           : pointer to an Object
  (1:1024)        : number of bytes to display

restart           : Shuts down the system.
(                 : type of shutdown...
  warm            : exits and recreates threads
  cold            : deletes sessions (plus warm actions)
  reload          : reloads data (plus cold and warm actions)
  reboot          : exits and restarts the entire system
  exit            : exits and does not restart the system
)

  OK.
nb>nt
nt>help full
set               : Controls trace tool settings.
(                 : what to set...
  tools           : trace tools: see >tools command for abbrevations
    <str>         : tools to set: string of tool abbrevations
  (               : setting...
```

```
    on            : on
    off           : off
  )
  buffsize        : size of event trace buffer
    (128:65536)   : buffer size in KBs
  scope           : scope for function tracing
  (               : how to trace function invocations
    full          : full trace of invocations
    counts        : count invocations per function
  )
)

save              : Saves what was captured by trace tools.
(                 : what to save...
  trace           : events captured by tools that are currently ON
    <str>         : filename for output
  funcs           : function call statistics
    <str>         : filename for output
  [               : how to sort (default=calls)
    calls         : by number of invocations
    times         : by net time in function
    names         : by function name
  ]
)

testcase          : Configures or executes testcases.
(                 : subcommand...
  prolog          : file to read before executing a testcase
    [<str>]       : filename (none if omitted)
  epilog          : file to read after a testcase passes
    [<str>]       : filename (none if omitted)
  recover         : file to read after a testcase fails
    [<str>]       : filename (epilog if omitted)
  begin           : executes a testcase (and concludes any previous one)
    <str>         : testcase filename
  end             : concludes a testcase
  failed          : records that the current testcase failed
    <int>         : failure code
    [<str>]       : explanation for failure
  query           : shows the counts of passed/failed testcases
  reset           : clears the counts of passed/failed testcases
)

swflags           : Supports flags used to control branching
(                 : subcommand...
  set             : modifies a flag's setting
    (0:31)        : flag identifier
  (               : setting...
    on            : on
```

```
    off           : off
  )
  clear           : clears all flags
  query           : lists flags that are on
)

sizes             : Displays class sizes.
  [t|f]           : display sizes in base classes? (default=f)

corrupt           : Corrupts a data structure for testing purposes.
(                 : what to corrupt...
  pool            : object pool
    (0:63)        : ObjectPoolId
    (0:1024)      : offset into free queue (0 = head)
)

lbc               : Tests a LeakyBucketCounter function.
  init            : Initializes the counter.
    (1:3600)      : capacity of bucket (limit)
    (1:3600)      : time to empty bucket (seconds)
  event           : Updates the counter when an event occurs.

q1                : Tests a Q1Way function.
  enq             : Adds an item to the end of the queue.
    (0:8)         : item number (0 = nullptr)
  henq            : Adds an item to the front of the queue.
    (0:8)         : item number (0 = nullptr)
  insert          : Inserts item#2 after item#1.
    (0:8)         : item number (0 = nullptr)
    (0:8)         : item number (0 = nullptr)
  deq             : Removes the item at the front of the queue.
  exq             : Removes an item from anywhere in the queue.
    (0:8)         : item number (0 = nullptr)
  first           : Returns the first item in the queue.
  next            : Returns the next item in the queue.
    (0:8)         : item number (0 = nullptr)
  count           : Returns the number of items in the queue.
  empty           : Returns true if the queue is empty.
  purge           : Deletes all the items in the queue.

q2                : Tests a Q2Way function.
  enq             : Adds an item to the end of the queue.
    (0:8)         : item number (0 = nullptr)
  henq            : Adds an item to the front of the queue.
    (0:8)         : item number (0 = nullptr)
  deq             : Removes the item at the front of the queue.
  exq             : Removes an item from anywhere in the queue.
    (0:8)         : item number (0 = nullptr)
  first           : Returns the first item in the queue.
```

```
  next            : Returns the next item in the queue.
    (0:8)         : item number (0 = nullptr)
  last            : Returns the last item in the queue.
  prev            : Returns the previous item.
    (0:8)         : item number (0 = nullptr)
  count           : Returns the number of items in the queue.
  empty           : Returns true if the queue is empty.
  purge           : Deletes all the items in the queue.

reg               : Tests a Registry function.
  init            : Initializes the registry.
    (0:8)         : maximum number of items in registry
  insert          : Adds an item to the registry.
    (0:8)         : item number (0 = nullptr)
    [0:31]        : registrant id
  remove          : Removes an item from the registry.
    (0:8)         : item number (0 = nullptr)
    [0:31]        : registrant id
  at              : Accesses an item in the registry.
    (0:31)        : registrant id
  first           : Returns the first item in the registry.
    [0:31]        : registrant id
  next            : Returns the next item in the registry.
    (0:8)         : item number (0 = nullptr)
  last            : Returns the last item in the registry.
  prev            : Returns the previous item in the registry.
    (0:8)         : item number (0 = nullptr)
  count           : Returns the number of items in the registry.

time              : Tests a SysTime function.
  ctor1           : Constructs the current time.
    (1:3)         : item number
  ctor2           : Constructs a specified time.
    (1:3)         : item number
    (1900:2100)   : year
    (1:12)        : month (Jan=1, Dec=12)
    (1:31)        : day of month
    (0:23)        : hours (24-hour clock)
    (0:59)        : minutes
    (0:59)        : seconds
    (0:999)       : milliseconds
  dayofweek       : Returns the time's day of the week.
    (1:3)         : item number
  dayofyear       : Returns the time's day of the year.
    (1:3)         : item number
  isleapyear      : Returns true if a year is a leap year.
    (1900:2100)   : year
  truncate        : Truncates the time at a specified field.
    (1:3)         : item number
```

```
  (               : time field
    year          : year field
    month         : month field
    day           : day field
    hour          : hours field
    min           : minutes field
    sec           : seconds field
    msec          : milliseconds field
  )
  round           : Rounds off the time at a specified field.
    (1:3)         : item number
  (               : time field
    year          : year field
    month         : month field
    day           : day field
    hour          : hours field
    min           : minutes field
    sec           : seconds field
    msec          : milliseconds field
  )
    (1:500)       : interval (must evenly divide the field's range)
  addmsecs        : Adds milliseconds to the time.
    (1:3)         : item number
    <int>         : number of milliseconds
  submsecs        : Subtracts milliseconds from the time.
    (1:3)         : item number
    <int>         : number of milliseconds
  msecsfromnow    : Returns the milliseconds from now to a time.
    (1:3)         : item number
  msecsuntil      : Returns the milliseconds from one time to another.
    (1:3)         : item number
    (1:3)         : item number
  adddays         : Adds days to the time.
    (1:3)         : item number
    <int>         : number of days
  subdays         : Subtracts days from the time.
    (1:3)         : item number
    <int>         : number of days
  strtime         : Displays the time in various formats.
    (1:3)         : item number

recover           : Tests thread recovery.
(                 : what to recover from...
  abort           : call abort()
  delete          : delete a thread
    (t|f)         : perform by 'this' (t) or by another thread (f)
  badptr          : dereference an invalid pointer
  divide          : divide by zero
  loop            : enter an infinite loop
```

```
  stack           : cause a stack overflow
  raise           : raise a signal
    <str>         : signal's name ('SIG...')
  return          : return from a thread
  swerr           : cause a software abort log
  terminate       : call terminate()
  trap            : cause a trap
    (t|f)         : perform by 'this' (t) or by another thread (f)
    <str>         : signal's name ('SIG...')
)

  OK.
nt>quit
nb>nw
nw>help full
ip                : Executes IP functions.
(                 : function to execute...
  hostname        : returns the name of this element
  nametoaddr      : maps a host name/service name to an IP address
    <str>         : name of host
    [<str>]       : name of IP service (or port number)
  addrtoname      : maps an IP address to a host name/service name
    <str>         : IP address and optional port: n.n.n.n[:p]
)

ipports           : Displays IP ports with input handlers.
  [0:65535]       : ipport_t
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

status            : Displays system statistics.

include           : Specifies what should be captured by trace tools.
(                 : what to include...
  all             : all activity
  (               : setting...
    on            : on
    off           : off
  )
  faction         : threads in a specific faction
    (0:7)         : faction
  thread          : a specific thread's activity
    (0:99)        : ThreadId
  peer            : messages to/from from a specific peer address/port
    <str>         : IP address and optional port: n.n.n.n[:p]
  port            : messages received by a specific IP port
    (0:65535)     : ipport_t
)

exclude           : Specifies what should not be captured by trace tools.
```

```
(                 : what to exclude...
  faction         : threads in a specific faction
    (0:7)         : faction
  thread          : a specific thread's activity
    (0:99)        : ThreadId
  peer            : messages to/from from a specific peer address/port
    <str>         : IP address and optional port: n.n.n.n[:p]
  port            : messages received by a specific IP port
    (0:65535)     : ipport_t
)

query             : Shows the status of trace tools.
(                 : what to query...
  buffer          : trace buffer
  tools           : trace tools
  selections      : all items included/excluded by trace tools
)

clear             : Clears the trace buffer, tools, or selections.
(                 : what to clear...
  buffer          : trace buffer
  tools           : trace tools
  selections      : all items included/excluded by trace tools
  faction         : threads in a specific faction
    (0:7)         : faction
  factions        : all included/excluded factions
  thread          : a specific thread's activity
    (0:99)        : ThreadId
  threads         : all included/excluded threads
  peer            : messages to/from from a specific peer address/port
    <str>         : IP address and optional port: n.n.n.n[:p]
  peers           : all included/excluded peers
  port            : messages received by a specific IP port
    (0:65535)     : ipport_t
  ports           : all included/excluded IP ports
)

  OK.
nw>quit
nb>sb
sb>help full
services          : Displays services.
  [0:511]         : ServiceId (default=all)
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

states            : Displays a service's states.
  (0:511)         : ServiceId
  [0:63]          : State::Id (default=all)
  [b|v]           : 'b'=brief 'v'=verbose (default='b')
```

```
events            : Displays a service's event names.
  (0:511)         : ServiceId
  [0:127]         : EventId (default=all)
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

handlers          : Displays a service's event handlers.
  (0:511)         : ServiceId
  [0:255]         : EventHandlerId (default=all)
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

triggers          : Displays a service's triggers.
  (0:511)         : ServiceId
  [0:127]         : TriggerId (default=all)
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

factories         : Displays factories.
  [0:255]         : FactoryId (default=all)
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

protocols         : Displays protocols.
  [0:255]         : ProtocolId (default=all)
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

signals           : Displays a protocol's signals.
  (0:255)         : ProtocolId
  [0:63]          : SignalId (default=all)
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

parameters        : Displays a protocol's parameters.
  (0:255)         : ProtocolId
  [0:63]          : ParameterId (default=all)
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

contexts          : Counts or displays contexts.
  [c|b|v]         : 'c'=count 'b'=brief 'v'=verbose (default='b')

ssms              : Counts or displays service state machines.
  [0:511]         : ServiceId (default=all)
  [c|b|v]         : 'c'=count 'b'=brief 'v'=verbose (default='b')

psms              : Counts or displays protocol state machines.
  [0:255]         : FactoryId (default=all)
  [c|b|v]         : 'c'=count 'b'=brief 'v'=verbose (default='b')

msgports          : Counts or displays message ports.
  [0:255]         : FactoryId (default=all)
  [c|b|v]         : 'c'=count 'b'=brief 'v'=verbose (default='b')
```

```
messages          : Counts or displays messages.
  [0:255]         : ProtocolId (default=all)
  [0:63]          : SignalId (default=all)
  [c|b|v]         : 'c'=count 'b'=brief 'v'=verbose (default='b')

timers            : Counts or displays timers.
  [0:255]         : FactoryId (default=all)
  [c|b|v]         : 'c'=count 'b'=brief 'v'=verbose (default='b')

invpools          : Displays invoker pools.
  [0:7]           : faction (default=all)
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

status            : Displays system statistics.

include           : Specifies what should be captured by trace tools.
(                 : what to include...
  all             : all activity
  (               : setting...
    on            : on
    off           : off
  )
  faction         : threads in a specific faction
    (0:7)         : faction
  thread          : a specific thread's activity
    (0:99)        : ThreadId
  peer            : messages to/from from a specific peer address/port
    <str>         : IP address and optional port: n.n.n.n[:p]
  port            : messages received by a specific IP port
    (0:65535)     : ipport_t
  factory         : messages received by a specific factory
    (0:255)       : FactoryId
  protocol        : messages in a specific protocol
    (0:255)       : ProtocolId
  signal          : messages with a specific protocol and signal
    (0:255)       : ProtocolId
    (0:63)        : SignalId
  service         : contexts in which a specific service is running
    (0:511)       : ServiceId
  timers          : timer registry work
)

exclude           : Specifies what should not be captured by trace tools.
(                 : what to exclude...
  faction         : threads in a specific faction
    (0:7)         : faction
  thread          : a specific thread's activity
    (0:99)        : ThreadId
  peer            : messages to/from from a specific peer address/port
```

```
    <str>          : IP address and optional port: n.n.n.n[:p]
  port             : messages received by a specific IP port
    (0:65535)      : ipport_t
  factory          : messages received by a specific factory
    (0:255)        : FactoryId
  protocol         : messages in a specific protocol
    (0:255)        : ProtocolId
  signal           : messages with a specific protocol and signal
    (0:255)        : ProtocolId
    (0:63)         : SignalId
  service          : contexts in which a specific service is running
    (0:511)        : ServiceId
  timers           : timer registry work
)

query              : Shows the status of trace tools.
(                  : what to query...
  buffer           : trace buffer
  tools            : trace tools
  selections       : all items included/excluded by trace tools
)

clear              : Clears the trace buffer, tools, or selections.
(                  : what to clear...
  buffer           : trace buffer
  tools            : trace tools
  selections       : all items included/excluded by trace tools
  faction          : threads in a specific faction
    (0:7)          : faction
  factions         : all included/excluded factions
  thread           : a specific thread's activity
    (0:99)         : ThreadId
  threads          : all included/excluded threads
  peer             : messages to/from from a specific peer address/port
    <str>          : IP address and optional port: n.n.n.n[:p]
  peers            : all included/excluded peers
  port             : messages received by a specific IP port
    (0:65535)      : ipport_t
  ports            : all included/excluded IP ports
  factory          : messages received by a specific factory
    (0:255)        : FactoryId
  factories        : all included/excluded factories
  protocol         : messages in a specific protocol
    (0:255)        : ProtocolId
  protocols        : all included/excluded protocols
  signal           : messages with a specific protocol and signal
    (0:255)        : ProtocolId
    (0:63)         : SignalId
  signals          : all included/excluded signals
```

```
  service          : contexts in which a specific service is running
    (0:511)        : ServiceId
  services         : all included/excluded services
  timers           : timer registry work
)

kill               : Kills a PSM's context.
  <hex>            : pointer to a PSM

  OK.
sb>quit
nb>st
st>help full
save               : Saves what was captured by trace tools.
(                  : what to save...
  trace            : events captured by tools that are currently ON
    <str>          : filename for output
  funcs            : function call statistics
    <str>          : filename for output
  [                : how to sort (default=calls)
    calls          : by number of invocations
    times          : by net time in function
    names          : by function name
  ]
  msc              : message sequence chart
    <str>          : filename for output
    [t|f]          : include internal data structures? (default=f)
)

testcase           : Configures or executes testcases.
(                  : subcommand...
  prolog           : file to read before executing a testcase
    [<str>]        : filename (none if omitted)
  epilog           : file to read after a testcase passes
    [<str>]        : filename (none if omitted)
  recover          : file to read after a testcase fails
    [<str>]        : filename (epilog if omitted)
  begin            : executes a testcase (and concludes any previous one)
    <str>          : testcase filename
  end              : concludes a testcase
  failed           : records that the current testcase failed
    <int>          : failure code
    [<str>]        : explanation for failure
  query            : shows the counts of passed/failed testcases
  reset            : clears the counts of passed/failed testcases
  verify           : enables or disables the >verify command
  (                : setting...
    on             : on
    off            : off
```

```
  )
)

sizes             : Displays class sizes.
  [t|f]           : display sizes in base classes? (default=f)

corrupt           : Corrupts a data structure for testing purposes.
(                 : what to corrupt...
  pool            : object pool
    (0:63)        : ObjectPoolId
    (0:1024)      : offset into free queue (0 = head)
  context         : first in-use context
)

inject            : Sends a message FROM a factory or one of its PSMs.
(                 : factory abbreviation...
  CO              : CIP Originator (network side)
    (1:16)        : TestSessionId
  (               : signal abbreviation...
    I             : IAM
      r           : RouteResult
        (0:255)   : selector (FactoryId)
        <int>     : identifier (factory-specific)
      <str>       : calling DN (digit string)
      <str>       : called DN (digit string)
      oclg=[<str>] : original calling DN (digit string)
      ocld=[<str>] : original called DN (digit string)
      m=[0:65000] : media.rxFrom: Switch::PortId
    C             : CPG
      (0:5)       : progress: Progress::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
    R             : REL
      (0:20)      : cause: Cause::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
  )
  CT              : CIP Terminator (network side)
    (1:16)        : TestSessionId
  (               : signal abbreviation...
    C             : CPG
      (0:5)       : progress: Progress::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
    A             : ANM
      m=[0:65000] : media.rxFrom: Switch::PortId
    R             : REL
      (0:20)      : cause: Cause::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
  )
  PX              : Proxy Call (user side)
    (1:16)        : TestSessionId
```

```
  (               : signal abbreviation...
    I             : IAM
      r           : RouteResult
        (0:255)   : selector (FactoryId)
        <int>     : identifier (factory-specific)
      <str>       : calling DN (digit string)
      <str>       : called DN (digit string)
      oclg=[<str>] : original calling DN (digit string)
      ocld=[<str>] : original called DN (digit string)
      m=[0:65000] : media.rxFrom: Switch::PortId
    C             : CPG
      (0:5)       : progress: Progress::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
    A             : ANM
      m=[0:65000] : media.rxFrom: Switch::PortId
    R             : REL
      (0:20)      : cause: Cause::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
  )
  PS              : POTS Shelf
  (               : signal abbreviation...
    B             : offhook ('begin')
      (0:65000)   : header.port: Switch::PortId
      m=[0:65000] : media.rxFrom: Switch::PortId
    D             : digits
      (0:65000)   : header.port: Switch::PortId
      <str>       : digit string: (0..9|*|#)*
    A             : alerting
      (0:65000)   : header.port: Switch::PortId
    L             : flash ('link')
      (0:65000)   : header.port: Switch::PortId
    E             : onhook ('end')
      (0:65000)   : header.port: Switch::PortId
  )
  PC              : POTS Call (user side)
    (1:16)        : TestSessionId
  (               : signal abbreviation...
    F             : facility
      (0:65000)   : header.port: Switch::PortId
      f           : facility info
        (0:511)   : sid: ServiceId
        (0:255)   : ind: Facility::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
      c=[0:20]    : cause: Cause::Ind
      p=[0:5]     : progress: Progress::Ind
    S             : supervise
      (0:65000)   : header.port: Switch::PortId
      r=[t|f]     : ring on?
      s=[<str>]   : scan: (x|d|f|df)
```

```
      m=[0:65000] : media.rxFrom: Switch::PortId
      c=[0:20]    : cause: Cause::Ind
      [f]         : facility info
        (0:511)   : sid: ServiceId
        (0:255)   : ind: Facility::Ind
    L             : lockout
      (0:65000)   : header.port: Switch::PortId
    R             : release
      (0:65000)   : header.port: Switch::PortId
      (0:20)      : cause: Cause::Ind
  )
  PM              : POTS Multiplexer (network side)
    (1:16)        : TestSessionId
  (               : signal abbreviation...
    B             : offhook ('begin')
      (0:65000)   : header.port: Switch::PortId
      m=[0:65000] : media.rxFrom: Switch::PortId
    D             : digits
      (0:65000)   : header.port: Switch::PortId
      <str>       : digit string: (0..9|*|#)*
    A             : alerting
      (0:65000)   : header.port: Switch::PortId
    L             : flash ('link')
      (0:65000)   : header.port: Switch::PortId
    E             : onhook ('end')
      (0:65000)   : header.port: Switch::PortId
    F             : facility
      (0:65000)   : header.port: Switch::PortId
      f           : facility info
        (0:511)   : sid: ServiceId
        (0:255)   : ind: Facility::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
      c=[0:20]    : cause: Cause::Ind
      p=[0:5]     : progress: Progress::Ind
    P             : progress
      (0:65000)   : header.port: Switch::PortId
      (0:5)       : progress: Progress::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
    R             : release
      (0:65000)   : header.port: Switch::PortId
      (0:20)      : cause: Cause::Ind
  )
)

verify            : Checks a message RECEIVED by a factory or one of its PSMs.
(                 : factory abbreviation...
  CO              : CIP Originator (network side)
    [0:16]        : TestSessionId (default=0: next message)
  (               : signal abbreviation...
```

```
    C             : CPG
      (0:5)       : progress: Progress::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
    A             : ANM
      m=[0:65000] : media.rxFrom: Switch::PortId
    R             : REL
      (0:20)      : cause: Cause::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
  )
  CT              : CIP Terminator (network side)
    [0:16]        : TestSessionId (default=0: next message)
  (               : signal abbreviation...
    I             : IAM
      r           : RouteResult
        (0:255)   : selector (FactoryId)
        <int>     : identifier (factory-specific)
      <str>       : calling DN (digit string)
      <str>       : called DN (digit string)
      oclg=[<str>] : original calling DN (digit string)
      ocld=[<str>] : original called DN (digit string)
      m=[0:65000] : media.rxFrom: Switch::PortId
    C             : CPG
      (0:5)       : progress: Progress::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
    R             : REL
      (0:20)      : cause: Cause::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
  )
  PX              : Proxy Call (user side)
    [0:16]        : TestSessionId (default=0: next message)
  (               : signal abbreviation...
    I             : IAM
      r           : RouteResult
        (0:255)   : selector (FactoryId)
        <int>     : identifier (factory-specific)
      <str>       : calling DN (digit string)
      <str>       : called DN (digit string)
      oclg=[<str>] : original calling DN (digit string)
      ocld=[<str>] : original called DN (digit string)
      m=[0:65000] : media.rxFrom: Switch::PortId
    C             : CPG
      (0:5)       : progress: Progress::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
    A             : ANM
      m=[0:65000] : media.rxFrom: Switch::PortId
    R             : REL
      (0:20)      : cause: Cause::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
  )
```

```
  PS              : POTS Shelf
  (               : signal abbreviation...
    S             : supervise
      (0:65000)   : header.port: Switch::PortId
      r=[t|f]     : ring on?
      s=[<str>]   : scan: (x|d|f|df)
      m=[0:65000] : media.rxFrom: Switch::PortId
      c=[0:20]    : cause: Cause::Ind
      [f]         : facility info
        (0:511)   : sid: ServiceId
        (0:255)   : ind: Facility::Ind
    L             : lockout
      (0:65000)   : header.port: Switch::PortId
    R             : release
      (0:65000)   : header.port: Switch::PortId
      (0:20)      : cause: Cause::Ind
  )
  PC              : POTS Call (user side)
    [0:16]        : TestSessionId (default=0: next message)
  (               : signal abbreviation...
    B             : offhook ('begin')
      (0:65000)   : header.port: Switch::PortId
      m=[0:65000] : media.rxFrom: Switch::PortId
    D             : digits
      (0:65000)   : header.port: Switch::PortId
      <str>       : digit string: (0..9|*|#)*
    A             : alerting
      (0:65000)   : header.port: Switch::PortId
    L             : flash ('link')
      (0:65000)   : header.port: Switch::PortId
    E             : onhook ('end')
      (0:65000)   : header.port: Switch::PortId
    F             : facility
      (0:65000)   : header.port: Switch::PortId
      f           : facility info
        (0:511)   : sid: ServiceId
        (0:255)   : ind: Facility::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
      c=[0:20]    : cause: Cause::Ind
      p=[0:5]     : progress: Progress::Ind
    P             : progress
      (0:65000)   : header.port: Switch::PortId
      (0:5)       : progress: Progress::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
    R             : release
      (0:65000)   : header.port: Switch::PortId
      (0:20)      : cause: Cause::Ind
  )
  PM              : POTS Multiplexer (network side)
```

```
    [0:16]        : TestSessionId (default=0: next message)
  (               : signal abbreviation...
    F             : facility
      (0:65000)   : header.port: Switch::PortId
      f           : facility info
        (0:511)   : sid: ServiceId
        (0:255)   : ind: Facility::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
      c=[0:20]    : cause: Cause::Ind
      p=[0:5]     : progress: Progress::Ind
    P             : progress
      (0:65000)   : header.port: Switch::PortId
      (0:5)       : progress: Progress::Ind
      m=[0:65000] : media.rxFrom: Switch::PortId
    S             : supervise
      (0:65000)   : header.port: Switch::PortId
      r=[t|f]     : ring on?
      s=[<str>]   : scan: (x|d|f|df)
      m=[0:65000] : media.rxFrom: Switch::PortId
      c=[0:20]    : cause: Cause::Ind
      [f]         : facility info
        (0:511)   : sid: ServiceId
        (0:255)   : ind: Facility::Ind
    L             : lockout
      (0:65000)   : header.port: Switch::PortId
    R             : release
      (0:65000)   : header.port: Switch::PortId
      (0:20)      : cause: Cause::Ind
  )
)

  OK.
st>quit
nb>pots
pots>help full
tsports           : Displays the circuit(s) in a range of timeswitch ports.
  (0:65000)       : Switch::PortId
  [0:65000]       : Switch::PortId

tones             : Displays tones.
  [0:10]          : Tone::Id (default=all)
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

meps              : Counts or displays media endpoints.
  [0:255]         : FactoryId (default=all)
  [c|b|v]         : 'c'=count 'b'=brief 'v'=verbose (default='b')

codes             : Displays service codes.
```

```
dns               : Displays the profile(s) in a range of DNs.
  (20000:99999)   : DN
  [20000:99999]   : DN

features          : Displays features that can be assigned to a DN.
  [0:63]          : PotsFeature::Id
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

register          : Adds a new DN.
  (20000:99999)   : DN

deregister        : Deletes a DN.
  (20000:99999)   : DN

subscribe         : Assigns a feature to a DN.
  (20000:99999)   : DN
(                 : feature abbreviation...
  sus             : Suspended Service
  boc             : Barring of Outgoing Calls
  htl             : Hot Line
    (20000:99999) : DN
  wml             : Warm Line
    dn=[20000:99999] : DN (must be valid to activate feature)
    to=[3:9]      : timeout (default=5)
  bic             : Barring of Incoming Calls
  cfu             : Call Forwarding Unconditional
    [20000:99999] : DN
  cfb             : Call Forwarding Busy
    [20000:99999] : DN
  cfn             : Call Forwarding No Answer
    dn=[20000:99999] : DN (must be valid to activate feature)
    to=[12:42]    : timeout (default=30)
  cwt             : Call Waiting
  twc             : Three-Way Calling
  cxf             : Call Transfer
)

activate          : Activates a feature assigned to a DN.
  (20000:99999)   : DN
(                 : feature abbreviation...
  wml             : Warm Line
    dn=[20000:99999] : DN (must be valid to activate feature)
    to=[3:9]      : timeout (default=5)
  cfu             : Call Forwarding Unconditional
    [20000:99999] : DN
  cfb             : Call Forwarding Busy
    [20000:99999] : DN
  cfn             : Call Forwarding No Answer
    dn=[20000:99999] : DN (must be valid to activate feature)
```

```
    to=[12:42]    : timeout (default=30)
)

deactivate        : Deactivates a feature assigned to a DN.
  (20000:99999)   : DN
(                 : feature abbreviation...
  wml             : Warm Line
  cfu             : Call Forwarding Unconditional
  cfb             : Call Forwarding Busy
  cfn             : Call Forwarding No Answer
)

unsubscribe       : Removes a feature from a DN.
  (20000:99999)   : DN
(                 : feature abbreviation...
  sus             : Suspended Service
  boc             : Barring of Outgoing Calls
  htl             : Hot Line
  wml             : Warm Line
  bic             : Barring of Incoming Calls
  cfu             : Call Forwarding Unconditional
  cfb             : Call Forwarding Busy
  cfn             : Call Forwarding No Answer
  cwt             : Call Waiting
  twc             : Three-Way Calling
  cxf             : Call Transfer
)

reset             : Resets a DN to its initial state.
  (20000:99999)   : DN

sizes             : Displays class sizes.
  [t|f]           : display sizes in base classes? (default=f)

  OK.
pots>quit
nb>an
an>help full
traffic           : Generates POTS calls for load testing.
(                 : subcommand...
  profile         : displays circuit and call states
  rate            : sets call rate
    (0:48460)     : calls per minute
  query           : displays traffic statistics
)

  OK.
an>quit
nb>sn
```

```
sn>help full
treatments        : Displays treatments.
  [0:5]           : PotsTreatmentRegistry::QId (default=all)
  [b|v]           : 'b'=brief 'v'=verbose (default='b')

sizes             : Displays class sizes.
  [t|f]           : display sizes in base classes? (default=f)

  OK.
sn>quit
nb>ct
ct>help full
import            : Adds a directory to the code base.
  <str>           : directory name
  [<str>]         : path within SourcePath configuration parameter

show              : Displays library information.
(                 : what to show...
  dirs            : code directories
  failed          : code files that failed to parse
  items           : memory usage by item type
)

type              : Displays the items in a set, separated by commas.
  <str>           : a set of code files or directories

list              : Displays the items in a set, one per line.
  <str>           : a set of code files or directories

count             : Counts the items in a set.
  <str>           : a set of code files or directories

countlines        : Counts the number of lines of code.
  <str>           : a set of code files

scan              : Scans files for lines that contain a string.
  <str>           : a set of code files
  <str>           : string to look for (quoted; '$' = wildcard)

assign            : Assigns a set of files or directories to a variable.
  <str>           : variable name
  <str>           : a set of code files or directories

purge             : Deletes a variable.
  <str>           : variable name

sort              : Sorts files by build dependency order.
  <str>           : a set of code files
```

```
fileinfo          : Displays information about a code file.
  <str>           : file name (including extension)

fileid            : Displays information about a code file.
  (1:4095)        : file's identifier

parse             : Parses code files.
  <str>           : parser options (enter >help full for details)
  <str>           : a set of code files

check             : Checks if code follows C++ guidelines.
  <str>           : filename for output
  <str>           : a set of code files

trim              : Analyzes #include and using statements.
  <str>           : filename for output
  <str>           : a set of code files

apply             : Interactively applies changes in a >trim file
  <str>           : file name (assumed to end in ".trim.txt")

format            : Reformats code files.
  <str>           : a set of code files or directories

export            : Exports library information.
  <str>           : filename for output

shrink            : Shrinks the library's element containers.

exp               : Performs an experimental test.

Commands in this increment often take an expression as the last parameter.
An expression contains variables and operators.  Variables can be defined
using the >assign command.  The code library also provides the following
variables, which cannot be modified using the >assign command:
   $dirs : directories that have been added to the code base by >import
  $files : all code files (headers and implementations) found in $dirs
   $hdrs : headers in $files
   $cpps : implementations (.c*) in $files
   $exts : headers whose directories were not included by an >import
   $subs : headers that declare a subset of the items in $exts in order
           to make them available during a compile
   $vars : all variables (those above, or user-defined)

An expression is evaluated left to right, but parentheses can be used to
override this.  A variable is a set of either directories or files.  The
following notation is used below:
  <ds> : the name of a directory (from >import) or a set of directories
  <fs> : the name of a specific file or a set of files
```

```
   <s> : a <ds> or an <fs>

A list of current operators appears below.  What the operator returns is
on the left; it becomes the input to commands such as >assign and >type.
The type of parameter(s) that the operator expects are shown on the right:
   <s> = <s1> | <s2>    set union of <s1> and <s2> (the '|' is optional)
   <s> = <s1> & <s2>    set intersection of <s1> and <s2>
   <s> = <s1> - <s2>    set difference between <s1> and s2>
  <fs> = f <ds>         the files in <ds>
  <ds> = d <fs>         the directories in <fs>
  <fs> = <fs> fn <str>  files in <fs> with the file name <str>*
  <fs> = <fs> ft <str>  files in <fs> with the file type *.<str>
  <fs> = <fs> ms <str>  files in <fs> that contain <str>
  <fs> = <fs> in <ds>   files in <fs> whose directory is in <ds>
  <fs> = im <fs>        implements: for each item declared in <fs>,
                        add the file that defines it, and vice versa
  <fs> = us <fs>        users: files that #include any in <fs>
  <fs> = ub <fs>        used by: files that any in <fs> #include
  <fs> = as <fs>        affecters: ub <fs>, transitively
  <fs> = ab <fs>        affected by: us <fs>, transitively
  <fs> = ca <fs>        common affecters: (as f1) & (as f2) & ... (as fN)
  <fs> = ns <fs>        needers: files that also need <fs> in a build
                        (im ab <fs>, transitively)
  <fs> = nb <fs>        needed by: files that <fs> also needs in a build
                        (im as <fs>, transitively)

Here is how the last six operators act on the <fs> parameter:
  lower layers     <fs> parameter     higher layers
       |<----ub/as/nb----|----us/ab/ns---->|

The im operator links declarations with definitions, so it requires all
files that affect <fs>, or that are affected by <fs>, to have been parsed.
The ns and nb operators use the im operator, so they also require these
files to have been parsed.  The parsing occurs automatically if required.

Here are some examples of library commands:
  >import nbase "nb" : add NodeBase files to the code base
  >import sbase "sb" : add SessionBase files to the code base
  >type us Thread.h : show all files that #include Thread.h
  >assign h1 f sbase ft cpp : h1 = all SessionBase implementations
  >assign c1 ab Thread.h : c1 = files that could be affected by
    changing Thread.h
  >assign s1 h1 & c1 : s1 = SessionBase .cpps that could be affected
    by changing Thread.h

Parser options in the >parse command:
  - (none)
  p (save parse tree on failure)
  s (always save parse tree)
```

```
  x (generate "object code" during execution)
  i (immediate tracing: needed if 'x' causes trace buffer to overflow)
  f (enable FunctionTracer)
```