AppleTalk Remote Access: Sample V.32bis Script with Comments

Article Created: 15 June 1992
Article Reviewed/Updated: 10 August 1993

TOPIC -----

I have a new V.32bis modem that I want to use with AppleTalk Remote Access (ARA), but there's no script on the application disk that I can use with it.

DISCUSSION -----

The following AppleTalk Remote Access script for the Telebit T3000 modem is a good example of how to write a script for any given V.32bis modem.

Notes

- This discussion assumes that you have access to the documentation for AppleTalk Remote Access and that you are familiar with the command syntax and scripting basics.

- The command syntax is in the AppleTalk Remote Access Modem Scripting Language Guide. You can order the guide from APDA at 800-282-2732 (USA), 800-637-0029 (Canada), or 408-562-3910 (International).

```
------
! "Telebit T3000 Modem 11/11/91"
! 11/14/91     - Added a ton o' comments
! 11/23/91     - Cleaned up comments a little bit more,
!                and added S11=50.
! 12/2/91      - Took S11 back out.
! 12/9/91      - Added the pinouts for the RTS/CTS cable to the comments
!                Since you have to make a hardware handshaking cable.
! 1/16/92      - Add a bunch more comments about the cable.
! 3/30/92      - Added comments about the hang up sequence.
! 3/31/92      - Added sbreak in hangup sequence to speed things up.
! 3/31/92      - Added S38=3 to help ensure that the modem will be
!                able to transmit all the data in its buffer (including
!                the disconnect command for the remote modem!) before
!                it hangs up.
! 4/2/92       - Changed S38 to 255.  See notes below.
! 4/2/92       - Changed S51 to 254 in hang up sequence since 19200 is
!                our DTE speed.
```

```
!
!  Note the cable requirements when using a V.32bis modem.  Since a lot
!  of people do not have this cable, you should not use this approach when
!  scripting for V.32 or slower modems.
!
!  The cable requirement for using a V.32bis modem with AppleTalk Remote
!  Access is a hardware handshaking cable.  The Macintosh can be overrun
!  with data when the modem is in high speed mode with compression enabled.
!  If a standard cable is used, ARA senses it and doesn't allow the modem
!  to connect at the high speed. It automatically drops down to 9600 bps.
!
!  The correct cables can be purchased from many of the modem vendors.
!
! Cable needed to use AppleTalk Remote Access with V.32bis modems:
! Din-8     DB-25
! -----     -----
! 1 (DTR)   4,20 (RTS, DTR)
! 2 (CTS)   5 (CTS)*
! 3 (TxD-)  2 (TD)
! 4 (SG)    7 (SG)
! 5 (RxD-)  3 (RD)
! 6 (TxD+)  Not Connected
! 7 (GPi)   8 (DCD)
! 8 (RxD+)  7 (SG)
!
! *Normally 2 (CTS) <-!6 (DSR) on other Macintosh cables.
!
!- One consequence of using this cable is that DSR (or DCD) from the modem
!  is no longer connected to the Macintosh.  This does not let your
!  Macintosh communications software use the DSR (or DCD) signal to detect
!  carrier loss.  And, since the Macintosh Serial driver does not support
!  the GPi input...you are sort of stuck, unless your communications
!  software does use the GPi input, or Apple builds GPi support into the
!  serial driver.
!
!- Since DTR and RTS are connected together, the modem must be configured
!  to ignore DTR (usually the &D0 command) when using this cable with
!  other communications applications.  Otherwise, when RTS handshaking from
!  the Macintosh is used, the modem will drop connection the first time the
!  Macintosh de-asserts RTS.
!
!- If there is a need to use DTR to make the modem disconnect, RTS
!  handshaking cannot be used to control the flow of data from the modem to
!  the Macintosh.  CTS handshaking (from the modem to the Macintosh) is
!  available.  This is what ARA does so it can force the modem to hang up,
!  and at the same time the modem can signal the Macintosh to stop sending
!  data.  This assumes that the Macintosh will always be able to accept
!  data from the modem.  This will not be true if the Macintosh is talking
!  to the modem at 57.6KBps with V.32bis & V.42Bis.  There will be times
!  when the Macintosh will need to signal the modem to stop sending data.
!
! In summary, with this cable:
!
```

```
! If you want to use RTS hardware handshaking, you cannot use DTR to
! control the modem.  You will have to resort to other methods to coerce
! the modem to disconnect.
!
! If you want to control the modem with DTR, you cannot use RTS hardware
! handshaking so the Macintosh must be able to accept data from the modem
! at all times, or can recover if data is lost.
!
! In either case you can use CTS hardware handshaking so the modem can
! signal the Macintosh to discontinue sending data.
!
@ORIGINATE
@ANSWER
!
! Talk to the modem at 19,200 bps.  The T3000 should auto-baud this
! unless the user has locked the port to a particular speed.  If it
! is locked to a different speed, the user will need to change that.
!
serreset 19200, 0, 8, 1
!
!  The idea here is to get the modem into a known state, and then change
!  only the registers that are necessary to support the connection.  Most
!  of the time AT&F will be sufficient, but some modems allow the user to
!  change the F0 parameters.  There isn't much that can be done to prevent
!  this, but if the modem has any pre-configured configurations, and one of
!  them sets hardware handshaking, use it.
!
! Recall the factory configuration
! F9 is the built in pre-configured setting for CTS/RTS handshaking on the
! T3000.  Since it's possible for the user to modify F0 parameters, this is
! a little safer.
!
! AT&F9 sets:
! &C1 - DCD is on after connect
! &D2 - DTR on/off disconnects
! S58=2 - Use RTS/CTS flow control in full-duplex mode
! S61=1 - Go into command mode when receiving break from DTE (see
!         @HANGUP for why the script cares about this).
!
!  Every time the script needs to send commands to the modem, the strategy
!  is: Clear all matchstrings, look for specific responses, and loop around
!  a couple of times.  Later in the script, certain loops pause 50-70
!  seconds, such as when the script dials a number and is waiting for a
!  connection. Other times, the script pauses 3-5 seconds and loops around.
!  When the script is sending commands to the modem, it should expect to
!  see a response within a couple of seconds, so it's best to look quickly
!  and exit with an error in a reasonable amount of time so the users do
!  not wait a for a long time before they are notified that they may need
!  to power-cycle/reset the modem.  When the script is dialing out over a
!  telephone system or PBX, it needs enough time to make a connection.  In
!  short, if it's communicating to a modem, loop in 3-7 second increments.
!  If the script is waiting for something other than a modem response (like
!  a completed connection or terminal server) it may need 60-70 seconds.
```

```
!
!  If the defaults cannot be set, jump down to label 59, which exits and
!  asks the user to check out the modem.  If an  AT&F command will not be
!  accepted, the modem may be hung and needs to be manually reset.
!
settries 0
matchclr
@LABEL 1
matchstr 1 4 "OK\13\10"
write "AT&F9\13"
matchread 30
inctries
iftries 2 59
! Modem is not responding, reset and send a break
DTRClear
pause 5
DTRSet
SBreak
jump 1
!
!  The script was able to get the modem into a default factory state.  Now
!  set the basic hardware type configuration such as command echo, hardware
!  handshaking, and DTR control.  If the &F9 command had not set up
!  handshaking this is where it would be done.  It's not desirable to
!  create one long command string with everything on it because some modems
!  cannot handle a long command string, and long strings are harder to
!  debug.  It's easy to enter an incorrect S-register value.  For the most
!  part, the following commands are probably common across a lot of modems,
!  but always look up the commands in the modem manual.
!
! Next, Set up the configuration: drop connection after losing DTR
! Turn off auto answer and command echo.
!
! &D3 - DTR on/off resets modem
! S0=0 - Don't answer calls
! E0 - Turn command echo off
!
@LABEL 4
matchclr
pause 5
matchstr 1 5 "OK\13\10"
write "AT&D3S0=0E0\13"
matchread 30
jump 59
!
!  Now that the modem hardware & flow control parameters are set, make sure
!  any protocol negotiation is disabled, and issue any modem specific
!  features here.  Make sure that MNP4/V.42, and MNP5-10/V.42bis
!  negotiations are disabled.  By the way, some V.32/V.32bis modems have an
!  option to disable Trellis error control,  which is part of the physical
!  layer modulation. This is not the same as MNP/V.42, and you do not want
!  to disable it!
!
```

```
!  Make sure that the modem is configured so it does NOT require error
!  control to complete a link.  ARA 1.0 does all error correction/data
!  compression in software.  All ARA wants is the fastest raw data pipe it
!  can get.  If the script spends time trying to negotiate some error
!  control, the modems and/or Remote Access may time out.
!
!  Also note the S38 configuration.  It is noted later in the script that
!  it is desirable to ensure that the modem's buffer has transmitted all of
!  its data before it actually hangs the modem up.  This ability appears to
!  be implemented on a lot of modems.
!
!  This set of commands is going to be implemented differently on different
!  vendors V.32bis modems.  In this example, Telebit uses S registers.
!  Other modems may use S registers (but different registers), or \
!  commands, or % commands; you get the idea.  (Did I mention that you
!  really, really want to have your modem manual handy?)
!
!  It is important that the modem is configured so that it returns
!  the connected speed, NOT the DTE speed.  The script needs to know what
!  the real line speed is in order to set ARA's internal timers.  Some
!  modems don't have the option to display the line speed.  In that case
!  the performance of the connection may not be optimal.
!
! Next, disable MNP and error control
! Turn on internal buffering (for V.32bis), set the delay before
! disconnect, and extended result codes (CTS/RTS flow control was set when
! we issued &F9, so it is not necessary to do it again).
!
! S180=0 - Turn off all error detection/correction (ARA does MNP and
!          compression itself.  It needs these turned off in the modem).
! S181=1 - Turn *on* DTE <-> line buffering if there is no error control.
!                             Since the modem will be talking to the
!                             Macintosh at 19,200 bps no matter what speed
!                             it connects at, this needs to be on.
! S38=255  - Wait until the modem's buffer is clear OR the other modem
!            disconnects after an ATH is issued before dropping the line.
!            This is done to ensure that all any data in the modem's buffer
!            has been transmitted to the remote modem before it
!            disconnects.  If the remote connection does not receive the
!            disconnect packet (usually the last one sent) it could take up
!            to 45 seconds for the remote connection to timeout and
!            disconnect.
! X2 - Issue extended result codes.  This will display busy, connect XXX,
!      and so on.  X2 will say "CONNECT XXX"  Where XXX is the line speed
!      (as opposed to DTE speed).  This is so ARA can determine what speed
!      the modems are communicating at for timing.
!
@LABEL 5
pause 5
matchstr 1 6 "OK\13\10"
write "ATS180=0S181=1S38=255X2\13"
matchread 30
jump 59
```

```
!
!  The modem should now be properly configured.  Now check to see if the
!  user has turned off the modem speaker.  If they have, send an additional
!  command to turn it off.
!
! If speaker on flag is true, jump to label 8.  Otherwise turn off the
! speaker.
!
@LABEL 6
ifstr 2 8 "1"
pause 5
matchstr 1 8 "OK\13\10"
write "ATM0\13"
matchread 30
jump 59
!
! The modem is ready so enable answering, or originate a call.
!
@LABEL 8
pause 5
ifANSWER 30
note "Dialing ^1" 3
write "ATDT^1\13"
!
!  Be aware that different modems will have different format strings
!  to return connection results.  You need to understand the different
!  possible strings and set this area (and then answer area at label 31) to
!  the appropriate value.  Also, remember that the modem was configured to
!  return the connect speed if possible (The X2 command up at label 5).
!  It's also useful if the modem can return busy, no dialtone, etc. since
!  the script will be able to exit quicker and let the user know what is
!  going on.
!
!  Also note that the script waits at the bottom of label 9 for a 70
!  seconds, rather then looping around.  Why?  Well, if the script
!  re-issues the dial command too soon, that would cause the modem to hang
!  up.  At this point the script should wait a reasonable amount of time
!  for one of these strings to return from the modem and take the
!  appropriate action.
!
@LABEL 9
matchstr 1  11 "CONNECT 1200\13\10"
matchstr 2  12 "CONNECT 2400\13\10"
matchstr 3  13 "CONNECT 4800\13\10"
matchstr 4  19 "CONNECT 7200\13\10"
matchstr 5  14 "CONNECT 9600\13\10"
matchstr 6  20 "CONNECT 12000\13\10"
matchstr 7  18 "CONNECT 14400\13\10"
matchstr 8  50 "NO CARRIER\13\10"
matchstr 9  50 "ERROR\13\10"
matchstr 10 52 "NO DIALTONE\13\10"
matchstr 11 53 "BUSY\13\10"
matchstr 12 54 "NO ANSWER\13\10"
```

```
matchread 700
jump 59
!
!  All that is done for different connect speeds is to issue a
!  "CommunicatingAt" command.  Remember, the interface speed is locked
!  to 19,200 bps so the script doesn't want to reset the serial speed after
!  it connects.
!
!  CommunicatingAt tells ARA what the actual line speed is so that it
!  can set its timers appropriately.  I guess performance would be
!  sub-optimal if this is not set...
!
@LABEL 11
note "Communicating at 1200 bps." 2
CommunicatingAt 1200
jump 15
!
@LABEL 12
note "Communicating at 2400 bps." 2
CommunicatingAt 2400
jump 15
!
@LABEL 13
note "Communicating at 4800 bps." 2
CommunicatingAt 4800
jump 15
!
@LABEL 19
note "Communicating at 7200 bps." 2
CommunicatingAt 7200
jump 15
!
@LABEL 14
note "Communicating at 9600 bps." 2
CommunicatingAt 9600
jump 15
!
@LABEL 20
note "Communicating at 12000 bps." 2
CommunicatingAt 12000
jump 15
!
@LABEL 18
note "Communicating at 14400 bps." 2
CommunicatingAt 14400
jump 15
!
! Set CTS handshaking ON in the serial port (that's the 1 in the HSReset
! command below)
!
!  The modems have connected, so enable hardware handshaking on the serial
!  port.  If the script is answering a telephone call, just exit right away
!  and starting communicating.  If the script is dialing out, give the
```

```
!  other end some time (3 seconds in this example) to get ready to talk to
!  this modem.  Exit 0 tells Remote Access that the script was successful
!  in attempting a connection.
!
@LABEL 15
HSReset 0 1 0 0 0 0
ifANSWER 16
pause 30
@LABEL 16
exit 0
!
!  Notice that the @ANSWER label is actually a comment here, and that
!  @ORIGINATE and @ANSWER start at the same place.  What's the point of
!  having separate entry points if they are not used?  Well, in the case of
!  modems, when they dial out or wait for a call, the setup is usually the
!  same.  One reason for separate entry points is when the script is not
!  directly talking to a modem, but maybe to a PBX or terminal server.  It
!  may be necessary to have completely different configuration for
!  answering and originating connections.
!
! @ANSWER
! Set up the modem to answer the telephone.
!
@LABEL 30
write "ATS0=1\13"
matchstr 1 31 "OK\13\10"
matchread 30
jump 59
!
!  What is userhook 1 doing in label 32?  Here's the idea:  Either this
!  script controls a server that is waiting to answer the telephone, or
!  it's waiting for a callback to a connection that was initiated.
!  AppleTalk Remote Access does a "passive" listen on the serial port (via
!  the Serial Port Arbitrator) so that other communications applications
!  can use the serial port when ARA is not using it.  When a call comes in
!  for a server or callback, there will be about 5-14 seconds while the
!  modems negotiate the connection.  What would happen if a communications
!  application on this Macintosh wanted to use the serial port during that
!  time?  Both connections would fail.  The userhook 1 command tells ARA to
!  mark the serial port in use.  When that happens, applications that want
!  to use the serial port will be told it's busy, and the incoming
!  connection can complete.  With that in mind, the strategy below is:
!  When the modem receives a ring, jump to label 32, issue the userhook 1
!  command, then jump back up to label 31, wait for the connect result code
!  and continue processing the script.
!
@LABEL 31
matchstr 1  32 "RING\13\10"
matchstr 2  11 "CONNECT 1200\13\10"
matchstr 3  12 "CONNECT 2400\13\10"
matchstr 4  13 "CONNECT 4800\13\10"
matchstr 5  19 "CONNECT 7200\13\10"
matchstr 6  14 "CONNECT 9600\13\10"
```

```
matchstr 7  20 "CONNECT 12000\13\10"
matchstr 8  18 "CONNECT 14400\13\10"
matchstr 9  50 "NO CARRIER\13\10"
matchstr 10 50 "ERROR\13\10"
matchstr 11 52 "NO DIALTONE\13\10"
matchstr 12 53 "BUSY\13\10"
matchstr 13 54 "NO ANSWER\13\10"
matchread 700
jump 31
!
@LABEL 32
userhook 1
note "Answering phone..." 2
jump 31
!
!  These are some common error messages when the line is busy, no dialtone,
!  etc. They are documented in the Scripting Language Guide.  When the
!  script exits with a code other than zero, Remote Access knows that the
!  connection failed, and will inform the user with a dialog.
!
! 50: error messages
!
@LABEL 50
exit -6021
!
@LABEL 52
exit -6020
!
@LABEL 53
exit -6022
!
@LABEL 54
exit -6023
!
@LABEL 59
exit -6019
!
! Hang up the modem
! Note:  Why try to enter command mode and hang up the line with ATH, when
!        de-asserting DTR will always work, and it is used as a last resort
!        anyway?  If DTR is used immediately,  the modem will hang up
!        immediately.  This can have the ill effect of hanging up before
!        all the data in the modem's internal transmit buffer has been
!        sent.  It is very desirable to have the last byte of data sent
!        make it out of the modem and across the phone line.  Typically,
!        the last packet sent is the disconnect packet, and if the other
!        side misses this packet, it may have to wait up to 45 seconds to
!        hang up.
!
@HANGUP
@LABEL 60
settries 0
HSReset 0 0 0 0 0 0
```

```
@LABEL 61
!
!  Here's the basic logic for hanging up: If the modem can be configured
!  to enter command mode when it receives a short break, send a short
!  break.  Send an ATH to hang the line up (and if possible up in the
!  configuration, set the modem to attempt to send all the data in the
!  buffer before it disconnects).  If that fails, it must still be on
!  line, so send the escape sequence to try to drop into command mode.
!  Don't issue a short break again since it did not work the first time.
!  If that fails, de-assert DTR which should force the modem to hang up
!  (make sure the cable is wired properly for this option!).
!  If +++ worked, don't send a short break again; flush the serial port
!  buffer in case the ATH failed due to any stray data hanging around.
!
!  How was this sequence determined?  Trial and error.  Different vendors'
!  modems behave differently when disconnecting.  Some modems will not
!  enter command modem during a disconnect, and the only option is to
!  de-assert DTR to force them to reset.  That's why DTR resets the modem
!  instead of just disconnecting it!  Experiment with this sequence to make
!  it function, but it should work with the majority of the modems
!  available.
!
! Now, since the Telebit modems will drop into command mode when they
! receive a short break (S61=1), issue one here.  This will speed up the
! disconnect sequence by about 5-6 seconds.  Then continue on with normal
! AT disconnect processing.
!
Sbreak
!
! Wait a brief amount of time (1/2 second in this case) so the modem will
! be ready to accept the ATH command.  Pause 1 actually seems to work ok,
! but it's set to 5 just to be safe.
!
pause 5
write "ATH\13"
matchclr
matchstr 1 63 "NO CARRIER\13\10"
matchstr 2 63 "OK\13\10"
matchstr 3 63 "ERROR\13\10"
matchread 30
inctries
iftries 3 63
! no response, try escape sequence
write "+++"
matchclr
matchstr 1 62 "OK\13\10"
matchread 15
!
! No Response from modem, toggle DTR
!
DTRClear
pause 5
DTRSet
```

```
jump 61
!
@LABEL 62
! Pause 1 second to ensure we meet the escape time delay
pause 10
Flush
write "ATH\13"
matchstr 1 63 "OK\13\10"
matchstr 2 63 "NO CARRIER\13\10"
matchstr 3 63 "ERROR\13\10"
matchread 30
jump 61
!
!  Now that the modems have disconnected, and the script has possibly reset
!  the modem, restore the factory settings.  Remember, the script may have
!  hung up the modem in order to get ready for a callback, or it wants to
!  get ready to wait to answer a call again.
!
! recall the factory settings.  Use &F9 again (see note at top of script)
!
@LABEL 63
matchclr
matchstr 1 64 "OK\13\10"
pause 15
write "AT&F9\13"
matchread 30
!
!  Now turn off auto answer if it was turned on to answer a call.  If this
!  script controls a server, the @ANSWER sequence will be called by ARA.
!  One other thing to watch out for here is that some modems expect to
!  talk to the DTE at the last connected speed.  If this is a V.32bis
!  modem and it just finished a connection with a 2400 baud modem, it
!  doesn't necessarily want to talk at 2400 the next time!  Some modems
!  don't exhibit this behavior, so play with it and see what happens.
!  Finally, since it successfully hung up,  exit the script with a result
!  code of 0 to let Remote Access know everything worked.
!
! Turn off auto answer, set S51 so modem will check interface speed on next
! connection.  If this is not done, the modem will not try to autobaud,
! with the result being it exits the script with an error.
!
! S51=254 - Autobaud (19200 bps default)
! S0=0 - Don't try to answer the phone
!
@LABEL 64
pause 5
matchstr 1 65 "OK\13\10"
write "ATS51=255S0=0\13"
matchread 20
!
@LABEL 65
exit 0
```

As of August 1993, there are related AppleTalk Remote Access script files posted to the Software Sampler folder on AppleLink. Use this path:

```
Software Sampler
  Apple SW Updates
    Macintosh
      Networking & Communications
        AppleTalk Remote Access 1.0
          User Supported Modem Scripts
```

For more information about using V.32bis modems with AppleTalk Remote Access, use "AppleTalk Remote Access AND V.32bis" as a search string.

Article Change History:
10 August 1993 - Reviewed for technical accuracy and included additional cable information.

Keywords: <None>

==========

19960215 11:05:19.00
Tech Info Library Article Number: 10246