Cheyenne BitShare
Advanced Topics and Troubleshooting

Introduction

The purpose and intent of this training module is to define the purpose of BitShare and BitWare utilities and to develop troubleshooting techniques that will aide in solving problems that may be encountered during the installation or usage of the products.

Whenever using a communications package in DOS or Windows a good understanding of the way communications products and communications ports interact with each other. Having a good knowledge of this will allow you to better assist the customer as well as guided them in configuring their network and make smart decisions when purchasing products.

BitShare provides a means of sharing a modem across the network, and it only requires a connection at the NetBios level. The program can be installed on a machine configured as a dedicated or non-dedicated “Modem Host”. It is best to describe the “modem server”, which is the machine that physically has the modem attached to it, as a “modem host” so you don’t inadvertently have a customer think the modem need to be installed on a network Server.

The program provides a number of different interfaces:

- Interrupt 14
- Novell NASI interface
- Windows communications interface

Objectives

Upon completion of this module you will be able to efficiently:

- Define and describe how to use the following BitShare utilities:
  - Parameter Editor
  - Server Monitor
- Define and describe the configuration files used by BitShare.
- Define and describe potential problems that may occur during the installation of BitShare

BitShare Overview

The communications Server MSERVER.EXE is a DOS TSR program that manages the serial ports on any PC connected to a NetBIOS LAN. Its function is to allow the ports to be used by other stations on the LAN.

Since MSERVER is a TSR it does not have to be dedicated to the Server function; Other DOS programs can run concurrently with the communication Server. BitShare includes a communication monitor program that displays information about port usage when the Server is loaded.

As many as 16 ports can be managed by a BitShare communications Server. The ports can be any combination of the standard built-in ports and those added using non-intelligent multi-port adapter cards. The ports can be connected to communication lines, with or without modems, or can be directly attached to other serial devices or computers. As the line connection is really the focus of interest rather than the port.

The MSERVER.EXE program uses the SERVER.CON parameter file. The parameter file describes both the hardware facilities for the shared ports and the software attributes to be used during connections. Hardware facilities include the port addresses of each line, the interrupt level being used, and the type of equipment connected to the line. Software attributes include names for the lines, names for groups of lines, names for services, and default values to use for line parameters when not specified in a connection request.

A basic parameter file for both the Server and Client is built during the installation process. These can be used without making any changes if the line parameters are suitable, or they can serve as a model for a more extensive file.

In order for a station on the LAN to use a line on the Server, a connection must be establish between the station and the line. The communications Server provides two types of connections: Outbound connections which are initiated by a request from a Client to the Server, and Inbound connections which are initiated by a call from an off-LAN device such as a PC calling a dial-in telephone number.

The outbound connection is referred to as Outbound Service. The connection is established by the Server and the Client then has exclusive use of the line until it disconnects.

The inbound connection is referred to as Inbound Service. The connection is established and the off-LAN caller can conduct a remote session on the network. The LAN Client has exclusive use of the line until it disconnects.

Requests for outbound connections ask for a line by a name, or use a group name to ask any line from a group of lines. Requests for inbound connections ask for service, not a line, again by naming the service. The communication Server establishes service names when INHOST is loaded on a Client workstation.

Controlling the Com Ports

After the MSERVER.EXE program is loaded on the Modem Host, it controls all direct access to the ports being shared. This means that you must not run any program that attempts to directly access the same ports that you assigned to the Server. Accessing the shared ports must be done using the MCLIENT OR MLCLIENT programs. Attempting to directly access a port that is being controlled by the Server will likely hang the system. However, if you are using a COM1, COM2, COM3 or COM4 port with the Server, and you want to run a program that uses one of these ports, you can remove the port from BitShare service using the Monitor program. When a port is removed from service, the port can be used directly by a program. This procedure will not work for multi-port boards because multiple port addresses are sharing a single interrupt level.

Naming Servers, Lines and Groups

To properly configure your Servers[Modem Hosts], it is important to understand the consequences of naming Servers, Lines, and Groups.

Every line on a Server must have a name and, within a single Server, line names must be unique. Lines on different Servers can have the same name although this would remove the ability to ask for a specific line. In general, if a Client makes a connection request using a line name, any Server having a line with the requested name can supply the connection. This is assuming the line is not in use by some other Client. The default line name given during Setup is the same as the Machine name given to the machine that is running the MSERVER TSR.

Lines can belong to one or more groups. You will probably want to assign each line to at least one group although this is not a requirement of the communication Server. Forming groups of lines on a Server makes a more powerful and flexible configuration increasing the chances of a Client getting a connection because the request can be satisfied by using any of several lines in a group. As with line names, group names must be unique within a single Server but can be duplicated across numerous Servers.

If you want to have a certain Server allocate a port for a line request from a Client, you must not duplicate line names across Servers. Likewise, if you want to have a certain Server allocate a port for a group request from a Client, no two Servers should have duplicate group names.

In a NetBIOS network, a connection to the communication Server can only be made when the Server name matches the name specified in the Client parameter set. By specifying Server names at each Client you can separate Servers and limit the access certain Clients have to those Servers.

Let’s look at example #1:

Server#	**Server name**	**Groups**	**Lines**
1	SERVER1	GRP1 (contains all lines)	LINE1 LINE2
2	SERVER2	GRP1 (contains all lines)	LINE1 LINE3

A Client whose parameter set specified SERVER1 as the Server name would have access only to the lines on the first Server, even though both Servers duplicate group and line names.

Now let’s consider example #2:

Server#	**Server name**	**Groups**	**Lines**
1	SERVER1	GRP1 (contains all lines)	LINE1 LINE2
2	SERVER1	GRP1 (contains all lines)	LINE1 LINE3

Any Client can get to all lines on both Servers. In this configuration only the line names LINE2 and LINE3 are guaranteed to be specific to one Server since they are unique in the configuration.

Lastly, let's consider example #3:

Server#	Server name	Groups	Lines
1	SERVER1	GRP1 (contains all lines)	LINE1 LINE2
2	SERVER2	GRP2 (contains all lines)	LINE1 LINE3

With this configuration we have the same separation as shown in example #1, assuming that some Clients use a group name of GRP1 and others use GRP2 for their connection requests.

If you use INT 14, the BitShare Extended INT 14, or the Windows communications interface for communication programs running on Client workstations, the maximum length of names for Server name, Group name and Line name is 16 characters.

If you choose to use the NASI interface for communication programs running on Client workstations, the differences in the length of names is as follows:

Resource	BitShare	NASI
Server Name	16 characters	8 characters
Group Name	16 characters	8 characters
Line Name	16 characters	14 characters

Important! Since NASI cannot accept names longer than those shown in the table above, any Server or Group name longer than 8 characters, or line names longer than 14 characters will never match an inquiry made by a communication program using the NASI interface. Server, Group and Line names that exceed the maximum allowed by NASI will not be visible to communication programs.

When using NASI the following conditions must be true:

1. All Servers must have unique names.
2. A single Server cannot have names duplicated in any combination.

Deciphering SERVER.CON Parameters

The SERVER.CON file is used to:

1. Define the equipment attached to each port on the communication Server.
2. Assign a name to each port.
3. Specify the default line parameters[speed, parity, etc.]
4. Set default network values.

The parameters in SERVER.CON, can be modified to match your requirements. You can create and modify the Server parameter file either by using PAREDIT or by using any text editor that can save in an ASCII text format. If you use PAREDIT, it will fill in defaults, prompt you for required values and perform consistency checks on all entries.

The communication Server parameter file consists of parameter sets[up to 99 can be defined]. Each parameter set is a group of statements that defines one configuration of the communication Server. For example, parameter set 1 may define four modems and four direct connects, while parameter set 2 may define eight modems.

A particular parameter set can be specified when MSERVER is invoked by using the following syntax:

MSERVER PARFILE [=SETNO]

Where PARFILE is the name of the parameter file [SERVER.CON], and SETNO is the parameter set number that must be between 1 and 99.

The PARSET statement defines the start of a parameter set within a .CON file. There is only one PARSET statement per parameter set and it is always the first statement in the set. Remember there can be up to 99 PARSET statements in a .CON file.

Figure 1 is an example SERVER.CON parameter file. You’ll notice that the first executable line in the file is the PARSET statement. Note the additional parameters included on the same command line as the PARSET statement.

The **PARSET** statement is immediately followed by a one or two digit number which defines the parameter set. Valid numbers are from 1 to 99.

The buffer space is the number of bytes to be reserved for communication buffers. The default is 15,360 or 16k.

The next statement is the **SERVER** statement. The syntax for the SERVER statement includes NAME= which is a 16 character network name for the communication Server. Remember, if you use the NASI interface the name is limited to 8 characters or less. The default SERVER NAME is BITSHARE.

Another optional parameter is the TIMEOUT= which is the frequency in minutes that the Server will check to ensure that a Client holding a connection is still active. The recommended value is 1 minute.

The next statement is the TRANSMIT statement which defines a set of transmission attributes for a communication line. These sets of attributes are used only when they are referenced in a LINE statement or a GROUP statement.

Notice that there are 6 different TRANSMIT statements, one for each baud rate.

The syntax for the TRANSMIT statement is as follows:

- NAME= is the reference name of this transmission parameter set that must appear with the TRANSMIT statement. NAME can be up to 4 characters.
- ALPARS=(speed,parity,data bits,stop bits)
    - speed is an integer that specifies the transmission speed in bits per second. Valid values are 110,150,300,600,1200,1800,2400,3600,4800, 7200,9600,14440,19200,38400,57600, or 115200.
    - parity can be one of the literals ODD, EVEN, MARK, SPACE or NONE
    - data bits is an integer that specifies the number of data bits for transmission. Valid values are: 1, 1.5, and 2.

  NOTE: If an ALPARS statement is not defined in a TRANSMIT set, the default values are: 2400 baud, No parity, 8 data bits, and 1 stop bit.
- FLOW=(X,Y)
    - X - is a decimal or hexadecimal number whose value is the same as the data stream character to be used for XON.
    - Y - is a decimal or hexadecimal number whose value is the same as the data stream character to be used for XOFF. **To specify no flow control, enter FLOW=(0,0). Hardware flow control(RTS/CTS) is selected by using the value FLOW=(1,1).**

The next statement is the LINE statement which defines the attributes of each port on the communication Server. Each of the keyword parameters in the LINE statement must appear on a separate line. The keywords can be in any order.

The syntax for the LINE statement is as follows:

- LINE NAME=name - is the external reference name of the line(e.g., LINE1) that **must appear** with the LINE statement. This name can be up to 16 characters long. The default name is the same as the REDIR name.
- PORT=port - is a hexadecimal integer specifying the base port address of the line. **This value must be supplied.** For example, if you were using COM1, the port address would be specified as PORT=3F8H.
- INTERRUPT=level - is the interrupt level for the line. **This value must be supplied.** The interrupt level for COM1 is INTERRUPT=4.
- TRANSMIT=name - is the name of a set of transmission parameters defined by a valid TRANSMIT statement.
- MODEM=Y/N/AT...:
    - Y indicates that a modem is connected to the line/port. You must specify this value or supply an initialization string if you intend to use the line for inbound service.

  The default initialization string is MODEM=ATQ0E1X1V1S0=0
    - AT indicates that you are supplying the modem initialization string to be used. The string must begin with the characters AT and can be up to 32 characters long.
    - N indicates that no modem is attached to the port. The Server will not check the port’s operating status and will not send any init string.

- AUTOBAUD= is for inbound lines.
  - Y indicates that the Server should match the port speed to the line speed reported by the modem when a call is answered.
  - N indicates that the port speed should be fixed at the value specified in the TRANSMIT set being used by the line.

The next statement is the GROUP statement which defines a name that is used to access a line or set of lines. The syntax for the command is as follows:

- GROUP NAME=gname - is the external reference name of the group. This value must be supplied with the GROUP statement.
- LINES=lines - are the names of the lines to be included in the group. All lines must be previously defined in a LINE statement and must be enclosed in ().
- INBOUND= is an integer specifying the number of lines in the group for which incoming calls will be answered by the Server.

FIGURE 1. - Sample SERVER.CON Parameter File

```
;*********************************************
; STANDARD SET FOR COM1 WITH MODEM 9600
;*********************************************
PARSET1  16384           BUFFER SPACE 16K
;
SERVER NAME=BITSHARE
;
;*********************************************
; Transmit sets  (Used in Lines & Groups)
;*********************************************
;
TRANSMIT NAME=T120       THE 1200 BAUD SET
  ALPARS=(1200,NONE,8,1) SPEED,PARITY,FORMAT
  FLOW=(1,1)             HARDWARE FLOW CONTROL
;
TRANSMIT NAME=T240       THE 2400 BAUD SET
  ALPARS=(2400,NONE,8,1) SPEED,PARITY,FORMAT
  FLOW=(1,1)             HARDWARE FLOW CONTROL
;
TRANSMIT NAME=T960       THE 9600 BAUD SET
  ALPARS=(9600,NONE,8,1) SPEED,PARITY,FORMAT
  FLOW=(1,1)             HARDWARE FLOW CONTROL
;
TRANSMIT NAME=T192       THE 19200 BAUD SET
  ALPARS=(19200,NONE,8,1) SPEED,PARITY,FORMAT
  FLOW=(1,1)             HARDWARE FLOW CONTROL
;
TRANSMIT NAME=T384       THE 38400 BAUD SET
  ALPARS=(38400,NONE,8,1) SPEED,PARITY,FORMAT
  FLOW=(1,1)             HARDWARE FLOW CONTROL
;
TRANSMIT NAME=T576       THE 57600 BAUD SET
  ALPARS=(57600,NONE,8,1) SPEED,PARITY,FORMAT
  FLOW=(1,1)             HARDWARE FLOW CONTROL
;
;*********************************************
; The line (port) definitions
;*********************************************
;
LINE NAME=SERVER
 PORT=3F8H               COM1 ADDRESS
 INTERRUPT=4             IRQ4
 MODEM=ATQ0E1X1V1S0=0    MODEM ATTACHED
 TRANSMIT=T960           9600 BAUD DEFAULTS
 AUTOBAUD=N              LOCK PORT SPEED AT 9600
;
;*********************************************
; The group definitions
;*********************************************
;
GROUP NAME=GROUP1        NAME OF THE GROUP
  LINES=(SERVER)
  INBOUND=(1)            INBOUND SUPPORT IF # > 0
;
;*********************************************
; END OF PARAMETER FILE
;*********************************************
```

Sharing a Modem

To share a modem that is connected to a communication Server requires a communication Client. If the modem is to be shared from a Workstation that does not have a modem physically attached to it the MCLIENT is used to scan for communication Servers via a NetBIOS connection. If the modem is to be shared from the Server that has the modem physically attached then MLCLIENT is required.

Regardless to which location you're sharing the modem from, a CLIENT.CON parameter file is used by the communication Client to provide information required for inbound and outbound connection requests.

If BitShare is to be used with Windows communication programs, the BitShare Client DLL provides access to modems at communication Servers. The Client DLL is loaded automatically by the Redirector(MACOMM.DRV) if it determines that the MCLIENT.EXE program has not been already loaded.

NOTE: The Local Client function is not provided in the Client DLL. If you want to use the Local Client, you must use the MLCLIENT.EXE program.

During the installation of LANtastic Power Suite you are prompted to enter the port to be redirected to the communication Server. This sets up the Client DLL in Windows, therefore you will just need to set up the Windows communication program to use the redirected com port. The default redirected port is COM4.

The MACOMM.DRV driver allows the Client to intercept Windows communications calls for one or more COM ports and redirect the request to communication Servers

To properly establish a redirection in Windows Setup must make the following changes:

- Change comm.drv line in [boot] to: COMM.DRV=MACOMM.DRV.
- Add the DEVICE=C:\LANTASTI\BITSHARE\MCLIENT.EXE to the 386Enh section of the System.INI.

The Windows redirections are managed through the Port Status & Setup program.

The MCLIENT.INI file contains information used by BitShare to define the redirected ports, locate the communications driver to use for non-redirected ports, locate the Client parameter file, and for the Redirector to locate the Client DLL. Figure 2 shows an example MCLIENT.INI file.

FIGURE 2. - Sample MCLIENT.INI

```
[Redirector]
CommDriver=RHSICOMM.DRV
ClientDLL=D:\LANTASTI\BITSHARE\MCLIENT.DLL
RedirectPorts=4

[ClientParms]
FileName=D:\LANTASTI\BITSHARE\CLIENT.CON
```

The Redirector section contains the information necessary to redirect Windows COM ports and allows the Redirector to locate the Client DLL.

NOTE: If a Windows communications driver is installed after BitShare, the BitShare redirector will no longer function.

The Redirectports parameter specifies the COM ports to be intercepted and redirected to the communications Server.

The ClientParms section tells the Redirector where to find the Client parameter file.

IMPORTANT! Most Windows communications programs use the Windows communications interface and can be satisfied by BitShare by using the Client DLL. However, if you have a Windows communications program that requires the INT 14 or NASI interface, you will need to load the MCLIENT.EXE at the DOS prompt. Also note that if you want to run a DOS communications program from a Windows DOS box you will need to load MCLIENT.EXE prior to starting Windows.

MCLIENT Command Line Parameters

MCLIENT can be started with the following command line parameters:

- - is used to unload the Client or Local Client from memory. (MCLIENT -)
- MENU requests the Client or Local Client to display a menu groups/lines used to allocate a line for a communications program using the INT 14 interface.
- CONNECT requests a connection to either the default name specified in the DEST= parameter of the CLIENT.CON file or a specific group/line name that you supply. This parameter is used to allocate a line for a communications program using the INT 14 interface.
- RELEASE releases the current connection to the Server port and is used to release lines used by a communications program.

NOTE: The INT 14 interface, unlike the NASI and Windows interface, does not provide functions to acquire and release a port on the communications Server. Therefore, the CONNECT and RELEASE parameters can be used in conjunction with the MCLIENT or MLCLIENT commands.

Below is a sample batch file using the CONNECT and RELEASE parameters:

```
c:\lantasti\bitshare\mclient connect
if errorlevel 1 goto noline   (connects returns 1 if no modem)
c:\comprog\comprog
goto :end
:noline
echo No modem available
:end
c:\lantasti\bitshare\mclient release
```

Deciphering CLIENT.CON Parameters

The communication Client parameter file, CLIENT.CON, selects the line or group name for making a connection, specifies the desired line parameters(speed,parity, etc.), and sets appropriate operating values.

As with the communication Server parameter file SERVER.CON, the communication Client parameter file CLIENT.CON consists of parameter sets(up to 99 can be defined). Each parameter set is a group of statements that defines one configuration of the communication Client. For example parameter set 1 may connect to LINE1, and parameter set 2 may connect to GROUP1.

The following command lines are included in the CLIENT.CON parameter file:

- The PARSET command is immediately followed by a one or two digit set number which identifies the parameter set. Valid values are from 1 to 99.
  - Following the PARSET command is the buffer space which is the number of bytes to be reserved for communication buffers.(default=8192)
- The next statement in the file is the SERVER statement which defines the network name of the communication Server, the time to wait for completion of any request, and the NASI interface level to be provided. Where:
  - NAME=name - is a 16 character network name for the communication Server or group of communication Servers to which connection requests will be sent. The name is the same as the REDIR name.
  - TIMEOUT is an integer ranging from 1 to 65,535 with each unit representing 1 sec. A value of 0 specifies no limit on the waiting period.
- The next statement is the CONNECT statement which defines a connection request that can be made to the communication Server.
  - QUEUE - indicates whether or not the request is to be queued at the communication Server.
  - RETRY - indicates whether or not the communication Client will periodically retry your connection request if no Server responds.
- The next statement is the TRANSMIT statement which defines a set of transmission attributes for a line when a connection is established. The attributes relates to items such as speed parity, and flow control.

Sample CLIENT.CON Parameter File

```
;********************************************************************
;* Edited using the Parameter Set Editor on 05/18/95 at 06:45:23     *
;********************************************************************
PARSET1 8192,
SERVER NAME=BITSHARE
  TIMEOUT=6
  NASI=2
  NETWORKS=2
CONNECT DEST=GROUP1
  SERVICE=ANYSTATION
  QUEUE=N
  RETRY=Y
TRANSMIT FLOW=(1,1)
  DTPARS=(140,60,720)
  TRIGGERS=(0DH)
```

Exercise #1 - Loading the Server and Client Communication Programs

This exercise covers loading the MSERVER program and the MCLIENT/MLCLIENT program from the DOS prompt.

1. Go to the machine that is going to be the communication Server. That is the machine that has the modem physically attached to it.
2. Login in to the server \\JEEVES and redirect a drive to the \BETA resource or install LANtastic Power Suite from the floppies or CROM.
3. Run either the DOS INSTALL.EXE or go to Windows and run SETUP.EXE.
4. Choose to install LANtastic NOS and Fax and Communications Software.
5. Enter the appropriate information for the NOS installation.
6. Enter appropriate information for the BitShare/BitWare installation. Choose to share modem later by manually running the Sharemdm.BAT file.
7. Upon completion of the installation, go to the c:\lantasti directory and view the contents of the Sharemdm.BAT file.

   Shown below is an example of the SHAREMDM.BAT file.

SAMPLE SHAREMDM.BAT

```
@REM
=============================================
@REM LANtastic Power Suite -
D:\LANTASTI\SHAREMDM.BAT
@REM BitShare Modem Sharing Startup
@REM Created On: 5-18-1995
@REM
=============================================
@Echo off
D:
cd D:\LANTASTI\BITSHARE
MSERVER.EXE
MLCLIENT.EXE CLIENT.CON
cd \
```

Notice that MSERVER.EXE and MLCLIENT.EXE is being loaded in this file. MCLIENT.EXE is optional and only needed if the modem is going to be used at communication Server itself.

8. Go to the c:\lantasti\bitshare directory.
9. Test the modem by entering the following command: **ECHO ATDT123 > COMx**.

   Where COMx is the COM port where the modem is physically attached. This command will activate the modem and try to dial out. This will let you know that the modem is configured properly. You must enter **ECHO ATH > COMx** to hang up the modem.

10. Run the MSERVER.EXE command manually at the DOS prompt by typing: **MSERVER SERVER.CON**

11. Note the screen.

12. Attempt to perform step 9 again.

    Note that the modem does not respond. This is due to the fact that the MSERVER program now has control of the port. Therefore, to use the modem on this machine you will need to load the communication Client program(MLCLIENT).

13. Run the communication Client program by typing MLCLIENT CLIENT.CON at the DOS prompt.

14. Repeat step 9 again choosing the redirected COM port. Type ECHO ATDT12345 > COM4. Remember that by default the Setup program chooses COM4 as the redirected port.

    Notice that the modem now responds to the command.

15. Enter ECHO ATH > COM4 to hang up the modem.

16. Go to the c:\lantasti\bitshare directory and run the MONITOR program.

17. Press the PGDN key and then press 1 to choose line 1.

18. Press Enter to toggle the line out.

19. Escape out to the DOS prompt.

20. Repeat step 9 again choosing the COM port the modem is actually attached to. Note that the modem does not respond to the command. Since the port has been toggled OUT there is no longer a redirection and therefore the communication Client program cannot gain access to it.

21. Run the MONITOR program and toggle the line back to InBnd.

22. Go to a Workstation that does not have the modem physically attached to it.

23. Install LANtastic Power Suite and choose to install LANtastic NOS and the Fax and Communication software.

24. Enter the appropriate information for the NOS installation.

25. Enter the appropriate information for the BitShare/BitWare installation. Choose no modem on this computer now or no time soon. Note that the redirected COM port defaults to COM4. This is the COM port to use for communication.

26. When the installation is complete go to the c:\lantasti\bitshare directory.

27. View the contents of the CLIENT.CON parameter file. At the same time view the contents of the CLIENT.CON parameter file on the Server. Compare the two files.

28. Test modem communication by entering the following command: **ECHO ATDT1234 > COM4**. The modem should not respond to the command. The modem did not respond because the MCLIENT.EXE communication program is not loaded yet.

29. From the c:\lantasti\bitshare directory enter the following command: **MCLIENT CLIENT.CON**.

30. Repeat step 27. The modem should respond to the command because the communication Client program is now loaded.

31. Unload the communication Client program on both computers by entering the following command: MCLIENT - on the Workstation and MLCLIENT - on the Server.

32. Unload the communication Server program on the Server by entering the following command: MSERVER -

Editing the Parameter Files

During the installation process SERVER.CON and CLIENT.CON files were created based on you answers to the installation questions.

These parameter files can be maintained using any text editor or the PAREDIT parameter editor program.

There are two general types of parameter values used by BitShare:

1. Parameters associated with the management of network traffic between Client and Server, such as packet size.
2. Parameters associated with the management of a modem, such as baud rate.

Since network parameters such as packet size can exist in both the Server and Client parameter files, the values in the Client parameter files take precedence over the Server values. This allows the Client stations to use different settings.

Parameter values for setting line and modem related items such as line speed, and parity can come from three different sources: the communication s program, Client parameters, and Server parameters. The values are used in exactly the order listed.

For example, if the communications program selected a line speed for the connection, the line speed would be used in making the connection. If the communications program did not indicate the line speed desired, the line speed in the ALPARS parameter of the Client parameter file would be used. Finally, if neither the communications program nor the Client parameter file selected a line speed, the line speed specified in the Server parameter file for the line allocated would be used.

The Parameter Set Editor(PAREDIT) is a DOS application that provides a set of menus for creating and editing parameter files.

Exercise #2 - Running PAREDIT

1. Go to the machine that has the modem physically attached to it.
2. Go to the c:\lantasti\bitshare directory.
3. Type PAREDIT and press Enter to run the program.
4. Press Enter on Station Type and select Server.
5. Choose parameter set 1. If you wanted to create a new parameter set you would select New set at this point and set 2 would be created.
6. Select Flat Network and press Enter. If you enter Yes for this option it indicates that there are not bridges or routers on the network. During the initialization, the stations attempts to dynamically locate any bridges on the network.

7. Select Server name. Notice that the default Server name is BITSHARE.
8. Change the name to **SERVER1.**
9. Select Transmit Sets and press Enter. You will see a list of the pre-defined transmit sets and have the option to create a new transmit set.
10. Select T960 and press Enter. Observe the settings for the transmit set.
11. Select Done to return to main menu.
12. Select Lines and press Enter. You will be prompted with the choice of a pre-defined line or to create a new line.
13. Select the pre-defined line and press Enter. Notice the parameters for the line. It is important that if you are creating a new line, the port address and irq is set properly.
14. Escape to the main menu.
15. Select Groups and press Enter.
16. Select the default GROUP1 and press Enter.
17. Notice the parameters for the GROUPS. A new line can be added only if it has already been defined.
18. Escape to the main menu.
19. Select Permit Sets and press Enter.
20. Select a line and press Enter. This allows you to specify up to 16 stations to have access to this line. Only the stations that are listed will have access to the line. The station identifier is its 12 hexadecimal digit network address.
21. Select File and choose Save and press Enter.
22. Go to a workstation and run PAREDIT.
23. Select Station Type and choose Client.
24. Select parameter set 1 and press Enter.
25. Select Connect Parms and press Enter.
26. Change the Server name to SERVER1 and press Enter.
27. Enter No for automatic retries.
28. Enter No for queue requests at server.
29. elect Line Parms and press Enter.
30. Choose Flow Control and press Enter.
31. Select No Flow Control and press Enter.
32. Choose Line Speed and press Enter.
33. Select the appropriate line speed.
34. Select File, Save and press Enter to continue.
35. Escape to DOS prompt.

Exercise #3 - Using MONITOR to Manage the Server

The communication Server Monitor program is used to provide a display of the resource usage and activity state for Clients using the Server. The monitor can also be used to:

- Disconnect Clients from the Server.
- Remove communication lines from service.
- Place communication lines that have been removed from service back into service.
- Collect the Server activity logging records and save as a ASCII text file.

Since the Monitor is a foreground program and the Server is an interrupt-driven process, the Server always has priority. This means that the Monitor will not affect the performance of the Server, it also means that the Monitor may not be updated on a regular basis if the Server is constantly servicing connections.

In order to run the MONITOR program the MSERVER program must be started first.

If the MONITOR program is invoked on a machine that does not have MSERVER resident in memory the following message will occur:

Communication Server is not active

If the MONITOR program is invoked on a machine that has MCLIENT loaded the following message will appear: **This station is not a communication Server.**

There are two screens that can be displayed by the MONITOR program:

- Connection Display which shows the active connections being serviced. This is the default screen upon starting the program.
- Line Display which shows the state of all of the lines as well as the communication parameters being used for active lines.

The information shown on the Connection display screen relates to Client connections, data transfer activity, error conditions, and the current XOFF status.

1. Go to the communication server and ensure the MSERVER program is loaded.
2. Go to the c:\lantasti\bitshare directory and type MONITOR to start the program.
3. The Connection Display will appear. The following is a list of the fields shown:
    - Stn Num - shows the number assigned by the Monitor to each active connection.
    - Station name - shows the network address of the Client that is connected to the communication Server.
    - Lne num - shows the line number used by the connection and is provided so that you can correlate the information from both Monitor displays.
    - Bytes sent/Bytes Recvd - shows the number of bytes transmitted to and from the Server from the off-LAN system.
    - Data Errs - shows the total number of data errors for the connection. Data errors include overruns, parity errors and framing errors.
    - Data Ovrn - shows the count of errors that were caused by asynchronous line overruns.

    - Buf Ovrn - shows the count of data errors that were caused by either a total shortage of buffer space or by the Server refusing to assign additional buffer space to a Client because the station had exceeded its "fair-share" buffer limit.
    - Bk - shows the number of “breaks” received during connection.
    - Cr Ls - shows the number of “carrier lost” conditions sensed during the connection.
    - Net Bufs - shows the number of buffers that contain data sent by the off-LAN system and are awaiting transmission from the Server to the Client.
    - OL Bufs - shows the number of buffers that contain data sent by the Client to the off-LAN system and are awaiting transmission to the off-LAN system.
    - XOFF STATUS - set of flags that shows the current XOFF conditions for the connection.

4. Press the PGDN key to view the Line Display.

   The information shows the status of ports defined in the parameter file when the Server was started. In addition, this display provides information on the communication parameters being used for connections. The following is a list of fields shown:

    - Num - shows the reference number assigned by Monitor to each asynchronous port. This number is used for removing a line from service or placing a line back in service.
    - Line Name - shows the name of the line as defined in the LINE statement of the Server parameter file.
    - State - shows the current state of the line.
    - XO - shows whether or not the connection is using flow control between the Client and the off-LAN system.
    - XF - shows whether or not the connection is using flow control between the Client and the off-LAN system.
    - C Time - is the character timer being used by the Server in 20ms units. The character timer sets a maximum time between placing characters in the buffers.
    - P Time - is the packet timer being used by the Server in 20ms units. The packet timer sets a maximum time for filling a buffer with data and begins when the first character is placed in a buffer.
    - First six - shows one to six characters being used as “trigger” characters. Receipt of any of these characters causes the Server to transmit the buffer being filled.
    - Baud - shows the current baud rate for the connection.
    - DB - number of data bits being used for communication with the off-LAN system.
    - PT - the parity being used for the communication with the off-LAN system.
    - SB - the number of stop bits being used for communication with the off-LAN system.

- Pk Siz - the size of the data packets being sent to the Client.

5. Go to the workstation that is the BitShare Client.
6. Type MCLIENT CONNECT to establish a connection.
7. Note the changes to the MONITOR display on the communication Server. It should now show a connection.
8. At the Client station enter the following command at the DOS prompt:

   ECHO ATDT9,8849675

9. Press PGUP on the communication Server machine and note the changes in the display.
10. Go to the communication Client.
11. Type MCLIENT RELEASE at the DOS prompt and press Enter.
12. Type MENU at the DOS prompt and notice the State of the line. The state should be AVAIL because there is not a connection established to the modem host.
13. Select either the Group name or the Line name and press Enter.
14. Type MENU at the DOS prompt and notice the State of the line has been changed to ALLOC because there is now a connection established. You would get the same results be entering MCLIENT CONNECT at the DOS prompt.
15. Go to the Server and run the MONITOR program. Notice the status of the line.
16. Go to the Client and type MCLIENT RELEASE and press Enter.
17. Type MCLIENT CONNECT and press Enter.
18. Go to the Server and run the MONITOR program and notice the status.
19. Press PGDN, select the line number (1), and press Enter toggle to modem out of service. Repeat the step to toggle the modem back to AVAIL status. This is how you would normally disconnect a user from the line.
20. Escape from the MONITOR program to the DOS prompt.

Problem:

When I run MCLIENT CONNECT I get the message “All communication servers are busy or unavailable” even though no one is using the modem.

Solution:

Even though the message “initialization completed” appeared when the MSERVER program was run on the Server, this does not mean that the modem is available for use.

Run the Monitor program and press the PGDN key. Look at the state field and see if it displays the state OUT, which indicates it was manually removed from service or MFAIL, indicating that the modem was turned off or is not functioning. Check for the following:

- modem is properly connected and powered on.
- modem is attached to the port specified in the Server.con file.
- modem works properly directly without BitShare.
- Set the MODEM=N parameter in the LINE statement for the port and try to access the modem again.

When the modem is initialized properly by MSERVER, the Monitor will reflect the state of the modem as AVAIL or INBD and is ready for connection.

Problem:

When connecting to a BBS using a third party INT 14 communication program i immediately get the connection dropped “No Carrier”.

Solution:

You need to check the FLOW control that is set in your communications program and confirm that BitShare is set the same way. If any changes are to be made you must unload and reload MCLIENT.

Problem:

When I load MSERVER I immediately get a network error and I cannot go any further.

Solution:

The Network Interface Card IRQ is probably set to the same IRQ as the communications port you are using. Change the NIC’s IRQ to solve the problem.

Problem:

When looking at the MONITOR screen I see frequent Data Overruns.

Solution:

This is usually an indication that the NIC IRQ is set at a higher priority(lower value) than the communications port you are using. Change the NIC’s IRQ to a higher value.

Problem:

Loading MSERVER on an older PC causes the workstation to hang.

Solution:

Check the BIOS date of the PC, if it is prior to 1988 it probably does not support INT 14 and should not be a communications Server. Configure as a workstation.

Problem:

I have a high speed modem but the INT 14 communications package I am using only allows me to go up to 9600 baud.

Solution:

The communications program probably supports only the basic INT 14 specification which is a maximum baud rate of 9600. To exceed this baud rate you must use an INT 14 communications program that supports the extended INT 14 specification.

Problem:

Can a 286 computer handle 4 9600 BPS modems as a communications Server?

Solution:

If the station is dedicated and not being used as a workstation it can handle the task easily.

Problem:

Will there be a considerable loss in performance working in WIndows on a 386 computer that has two 9600 BPS modems on COM1 and COM2?

Solution:

The user of the 386 will not see a significant decrease in performance but the users of the modems will see a slight decrease in performance when using the modems at 9600.

Problem:

Can MSERVER and MCLIENT be loaded into upper memory?

Solution:

Yes, by using the loadhigh included with DOS 5.0 or above, or by using any memory manager.

Problem:

Can I write a batch file that will automatically load MCLIENT and start my communications program but also make sure there is a valid connection established.

Solution:

This can be done using the MCLIENT CONNECT command. The command will return a DOS error level of 1 if the connection failed. Trap this error level and create a loop in your batch file.

Multiple Devices Assigned to a Interrupt Level

When using standard PC communications ports, you are limited to two lines because each needs a separate interrupt level to operate properly. This is true even for the boards that are able to be addressed individually, in which case there are four addresses but still only two interrupt levels. The following is a list of the standard ports and their interrupt levels:

- COM1 = 4
- COM2 = 3
- COM3 = 4
- COM4 = 3

You have the option of installing a non-intelligent multi-port board which contains either four or eight external connections. A multi-port board uses only one interrupt and you can install two per Server and specify the interrupt level and addresses.

Duplicate Hardware Addresses

Every board in the system must have a different base address. Each line(port) being shared by a Server must have a hardware address that is unique and is not being used by any other device in the system. The following is a list of standard COM addresses:

- COM1 = 3F8
- COM2 = 2F8
- COM3 = 3E8
- COM4 = 2E8

Incorrect Parameter Set Values

You have to remember that the communication values specified by the Client will override those values specified by the Server when a connection is made. The following is the order for collection and use of the parameter set values:

- any value supplied by a communications program in use will override the client and server values.
- any value supplied by the Client will override the Server’s values.
- any value not supplied by the Client is obtain from the Server.
- any value not supplied in either parameter file is defaulted based on values internally kept in the Server.

Modem Set to Auto-Answer Mode When using Inbound

Since the communications Server controls the modem conditioning for receiving calls, your modem must have auto-answer disabled and DTR set to Normal modem to get the Inbound feature to work properly.

Not Enough Buffer Space

Buffers are used to hold information sent to and received from the Server. Increasing the amount of buffer space allows a more efficient system since information moves much faster across the network than the movement of data across communications lines.

Incorrect Communications Parameters

you must be care not to specify a parameter that is not support by your equipment. An example would be to set up your 2400 BPS modem for a faster rate.

Wrong Type of Flow Control

The type of flow control you select is dependent on the hardware capabilities of your modem and the type of flow control supported by(and selected in) your communications software. BitShare defaults to XON/XOFF flow control.

Incorrect Parity, Data Bit, and Stop Bit Combination

The choice of these is dictated by your communications software.

Multiple Network Cards in the Network/Communication Server

When the communication Server is also being used as a Network Server and there are multiple adapters installed in the computer the BitShare communication programs will not operate through all of the adapters, only through adapter 0.

Therefore the computer cannot be used as a bridge to allow multiple network segments access to the shared modems.

Cheyenne BitShare Advanced Topics and Troubleshooting

BIT_TBL
Revision 0
05/19/95

ARTISOFT CORPORATE MISSION

To be a leading developer and marketer of network solutions to workgroups, small businesses, and remote network users.

SALES AND SUPPORT MISSION

To establish channel franchises through sales execution, support, and technical services on a global basis.

SALES AND SERVICE DEPARTMENT MISSION

To provide such excellent service to our customers that they will recommend and repurchase our products.

SERVICE AND PRODUCT QUALITY DEPARTMENT MISSION

To provide Artisoft Technical Support and Sales associates with the best training and best information tools in the industry.

PLEASE NOTE: This is a training document. It is not intended for use as a troubleshooting reference. Product features and concerns can and often do change without notice. Please refer to source documents or Folios for the most current product information.

©1995 ARTISOFT, INC. All rights Reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means without the written permission of ARTISOFT, INC.