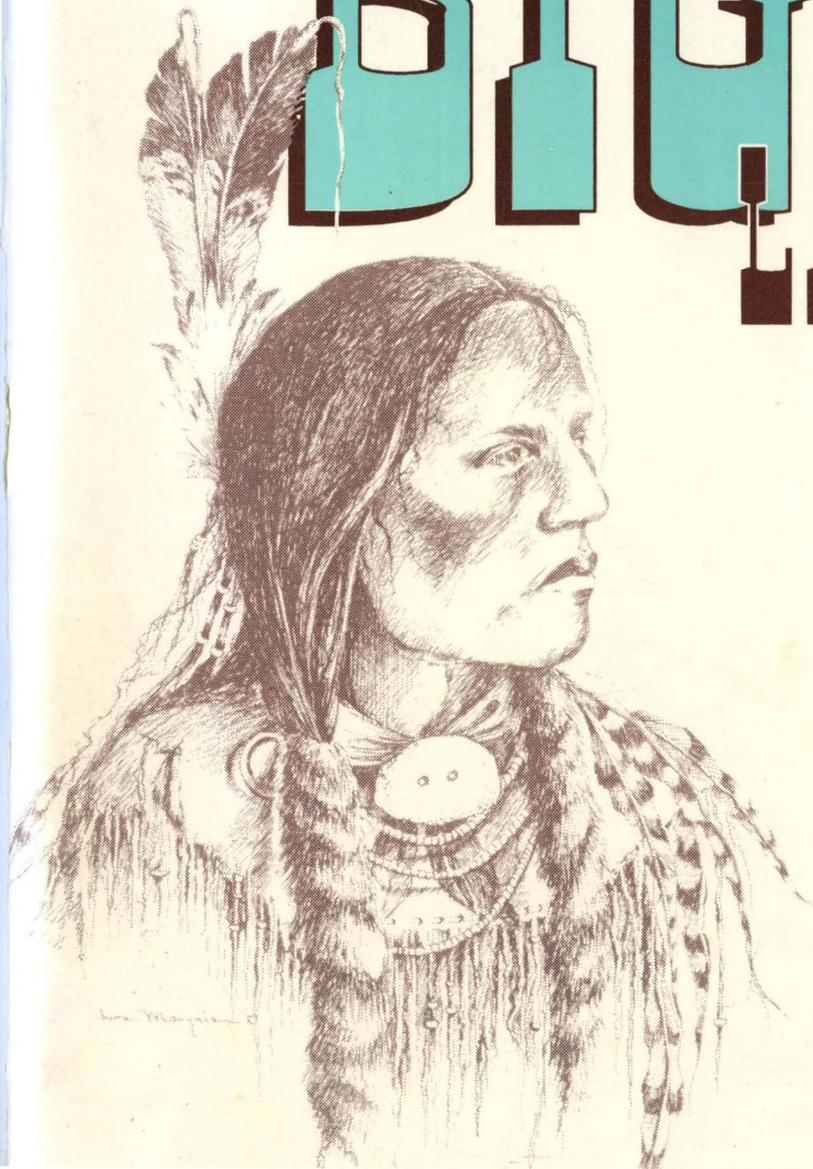


**BIG**

**LAN**



# Little Big LAN

Version 1.0m...

(c) Copyright 1990-1994  
Information Modes  
P.O. Drawer F.  
Denton, Texas 76202

Documentation manual for "Little Big LAN"  
Written by Don Jindra  
(with help from Alan Bretzin & Jim Bissett)  
Revision - January 1994

Cover Artwork by June Marquis

(C)Copyright 1994 by INFORMATION MODES.  
All Rights Reserved.

## FINE PRINT

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of INFORMATION MODES. With respect to this documentation, no warranties are either stated or implied. This especially holds true if the documentation is either not read or ignored.

Although every effort will be made to keep this information both valid and up to date, we assume no financial responsibility or liability for errors that may appear in this document.

From time to time, for the purpose of product improvement, changes may occur in filenames and files included on the distribution disks. Therefore, files or facilities as mentioned in this documentation may or may not exist on the current distribution disks or as part of the materials distributed. But other materials may be included which enhance or supercede those here mentioned.

This product is for single site use. We expect you to copy this to several machines on one integrated network. But any other copying or distribution of this product by any unauthorized person or business entity is expressly forbidden. This product is not shareware or public domain. It may not be shared freely. Any incorporation of this product, either in part or in whole, into another must receive prior written approval from Information Modes.

[		]
[	Little Big LAN V1.0q	]
[	Serial number: JM1160004862	]
[		]

# TABLE OF CONTENTS

Introduction.....	1
System Requirements.....	2
The Future.....	3
Decisions.....	4
The Physical Layer.....	4
Advantages & Disadvantages.....	5
Serial.....	5
Parallel.....	5
Arcnet.....	5
Ethernet.....	6
Modems.....	6
Cabling.....	7
Serial.....	7
Parallel.....	7
Arcnet.....	8
Star cards and hubs.....	8
Hubs come in two types.....	9
Bus cards.....	9
Bus - coax type.....	10
Bus - twisted pair.....	10
Expanding the bus.....	10
Ethernet.....	11
10base2 (coax).....	11
10baseT (twisted pair).....	11
10base5 (thicknet).....	12
Software Installation.....	13
Introduction.....	13
Theory.....	13
Preparation.....	14
INSTALL.EXE.....	15
List all Computers.....	15
Specify Connections.....	16
Special Case: Ethernet.....	16
Special Case: Modems.....	17
Entering node details.....	17
Specifying an INT.....	18
Specifying "Link to node".....	18
Specifying I/O.....	18
Specifying MEM.....	18
Specifying DMA.....	19
Checking your work.....	19
Saving your work.....	19
The First Test.....	19
Specifying Printers.....	20
Final Install.....	21
Manual (expert) mode.....	22
Configuration examples.....	23
Configuration Files.....	25
File Structure.....	26
NODELIST.LBL.....	26
CONNLIST.n.....	27
DRIVLIST.n.....	28

Diagnostics.....	29
ETHTEST - ethernet.....	29
PKTEST - ethernet.....	31
Loading the Packet Driver.....	32
ARCTEST - arcnet.....	33
PARTEST - parallel.....	35
COMTEST - serial.....	36
TMT/RCV mode.....	37
WATCH.....	38
File descriptions.....	39
ARC_LINK.COM.....	40
CDROM.COM.....	42
COMMUX.COM.....	43
COM_LINK.COM.....	44
ETHNAME.COM.....	47
ETH_LINK.COM.....	48
LBL.COM - screen.....	50
LBL.COM - batch.....	54
MDM_LINK.COM.....	55
N21_LINK.COM.....	57
NCON/RCON.....	58
NET00000.SYS.....	59
NET21.COM.....	61
NET8.COM.....	64
NETALIVE.COM.....	65
NETBIOS.COM.....	66
NETBSHAR.SYS.....	67
NETCLOCK.COM.....	70
NETCOM.COM.....	71
NETMSHARE.SYS.....	72
NETSHARE.SYS.....	73
NETSPOOL.SYS.....	74
NETUNITS.SYS.....	76
NODE.COM.....	77
PARMUX.COM.....	78
PAR_LINK.COM.....	79
PKT_LINK.COM.....	81
PORT.....	82
STIO.COM.....	83
Cables - serial.....	84
Non-standard cables.....	85
Ready made cables.....	85
Cables - parallel.....	86
Cables - 10baseT.....	87
Technical Notes.....	88
File Sharing.....	88
SUBST.....	88
Trouble shooting.....	89
Glossary.....	90

# Introduction

If you have two or more computers you know from experience that it's always the other computer that seems to have something you need. That something may be a file, a printer, or a program. As you acquire more and more computer "stuff" the problem becomes more and more acute. If this sounds familiar, this software was written for you.

Little Big LAN is a Local Area Network program. Sometimes such software is referred to as a Network Operating System (NOS). This software is really a collection of programs which attach themselves to DOS and become an extension to DOS. Little Big LAN is designed to look and feel like DOS so what you know about DOS should apply to this product.

When Little Big LAN is properly installed on your computers you should be able to use almost any hardware or software located on any of your computers from any other computer. All of this hardware and software looks to you like it is attached to your machine - the network is a master of disguises. As a user, all you need to know is that you have a lot more stuff available to you on your machine than you ever had before. After proper installation, the network handles all the details of where that extra stuff is and how to get it to you.

Things you should be able to share among computers include disk drives, printers, software, databases, plotters, and CD-ROMS. In some circumstances it may be possible to share remote modems and faxes. If you have another manufacturer's network then Little Big LAN may be able to use those network drives too.

If there was one word we had to choose to describe the overall design goal when developing Little Big LAN, that word would be "FLEXIBLE." Therefore, Little Big LAN is a peer to peer network. Every computer you wish to put on the network has full access to any other computer on the network. There are no "server" or "client" machines as with some other networks.

The Little Big LAN is flexible in that every drive letter is available to you to refer to any drive on any computer on the network. Drive G: for instance could be 'pointed' to your local hard drive, but it could also be redirected to access your laptop's B: drive, or your secretary's C: drive. Even your A: and B: drives can be redirected. Also printer names such as LPT1 can be redirected to any printer you have on any machine.

Little Big LAN offers you flexible growth. You can start with a two computer network linked through serial ports. This will cost a fraction of most other networks for software and even less in cabling costs. So for less than you might spend on an Ethernet adapter you can have your first LAN. Then if next month you buy a new 386, just link it into your other machines through another serial port and cable. There is no additional

software cost. We don't care how many machines you link together in one network at one site. We would be happy for you to get maximum benefit from your one licensed copy.

If you later decide serial port connections are too slow you can switch to using parallel ports instead. Parallel port connections are about 2.5 to 3 times faster than serial port connections if you are using AT class computers. If you are really hungry for speed, you can upgrade to Arcnet or Ethernet cards which offer about five to twenty times performance over serial ports.

Arcnet cards are typically less expensive than Ethernet cards and still give good performance. Ethernet cards have dropped in price dramatically over the past few years so for best performance per dollar you might want to use Ethernet.

Again, no matter which interconnection scheme you choose, Little Big LAN supports it at the same software cost. You don't have to come to us to buy a new driver, or a higher performance version of our software. We support all for the same low cost. **ONE SIZE FITS ALL!**

Not only can you choose different methods to link machines on the network, you can even mix those methods on the same network. For instance, some of your machines could be linked via Ethernet cards, and others could be linked into the network via serial ports and/or parallel ports. This kind of flexibility is offered by few other LAN products, and if offered is usually a high priced option. It's standard on ours.

You should be aware that there are a few things you may not be able to share. MODEMS present a problem - not because it is theoretically impossible to share modems, but in practice most commonly available communication software such as Procomm, Qmodem, and Telix read and write to the hardware directly. It is therefore not possible for any network to trap those hardware requests and redirect them. But because of the proliferation of networks more and more communication products are supporting redirection.

Sharing MICE or digitizers would pose the same sort of problem. FAX boards are also similar but with some it is possible to do certain things. If you have copy protected software it will probably force you to buy a legitimate copy for any computer running that software. This is especially true if the copy protection is hardware based - such as via some gizmo that attaches to your printer port. It is not possible for a network to defeat this method of copy protection.

## **System Requirements**

- o An IBM PC,XT,AT,386,486,Pentium, IBM PS/2, or 100% compatible,
- o DOS Version 2.0 or above, but 3.1 or above preferred, or DR-DOS 3.41 and above,
- o At least 1 serial port, parallel port or LAN card per computer,

- o Appropriate cables,
- o 35k-45k of available memory,
- o At least one disk drive per computer (no diskless workstations).

## **The Future**

Software is never finished, its just "further along." Therefore more utilities and features will be added as time permits. If you have any requests please pass them along. We have a list of upcoming features:

### **\*\*\* NEAR FUTURE: \*\*\***

- o Password protection for subdirectories
- o Remote booting, ie, diskless workstations
- o Built in EMAIL
- o OS/2 support
- o NDIS driver support
- o Token Ring support
- o Minimal cost upgrades to for at least the next year.

Please do not try to force us to commit to any dates on these features. We will implement them as soon as we can and probably in order of requests and ease of programming. It could be next week or next year.

Some of these things will appear prior to changing this manual so for the current status of any of these additions or others please look at the README.DOC file on the distribution diskette.

# Decisions

Little Big LAN is very FLEXIBLE. With that flexibility comes a price - you are free to choose among a wide variety of options. What's wrong with that you might ask. Well, the problem is that you must make decisions which nobody but you can make. We cannot offer you much help with many of those choices. We will attempt to give you enough information to make an informed decision but that's about as good as we can do.

The first and most basic question you need to ask yourself is: "How am I going to link my computers." What follow are some options.

## **The Physical layer**

- or -

"How are these computers going to talk to one another, anyway?"

One of the most powerful features of Little Big LAN is the freedom you have to choose the method by which you will connect your computers. In network jargon a computer on the network is called a node. There are presently five different methods of interconnecting nodes.

- o Serial ports
- o Parallel ports
- o ARCnet cards
- o Ethernet cards
- o Modems

Not only can you choose any of these methods to link all of your computers, you can mix any method on the same network. Several computers could be linked via Ethernet cards, and any of these nodes could also be linked to other computers via Arcnet or serial or parallel ports or modems. Why would you want to do that? Because all of these methods have both advantages and disadvantages. Little Big LAN allows you to tailor your network to fit your present needs. Later you can change interconnection methods as your needs change.

## **Advantages and Disadvantages**

### **SERIAL:**

Linking your computers through serial ports (also called communication ports, COM ports, or RS232 ports) is the least expensive, and usually the simplest method. Serial ports are cheap and abundantly available. Most computers have at least one free serial port so setting up Little Big LAN would require only the purchase of inexpensive null modem cabling. These cables can be successfully used to lengths of about 80 feet. This is long enough for most office environments. With some laptop computers your only linkage option will be serial ports.

The biggest disadvantage you have with serial ports is lack of speed. The pure baud rate of the cards is 115,200 bits per second. Between two XT's the true rate of data transfer is about 6000 to 6500 bytes per second. Between two AT's the rate is about 8000-8500 bytes per second. This means loading a 250k program across the network would take about 30 seconds. This would be unacceptable if you were having to load large programs repeatedly throughout the day. But if you were mostly using remote data such as database files, spread sheets, or letters and documents then this kind of speed might be perfectly acceptable.

### **PARALLEL:**

Parallel ports are also cheap. Many computers have only one printer but two printer ports. But proper cables are hard to find, and hard to make yourself. Speed is fairly good. Between two AT's the parallel ports are between 2 1/2 to 3 times faster than serial ports. This puts the true data rate in the range of 25000 bytes per second. But since parallel port transfer speed is directly related to CPU speed, the transfer rate between two 4.77 MHZ XT's is actually slower than serial port transfers. The transfer rate between two 10 MHZ XT's can be almost double serial port transfers on the same machines. But the transfer rate between two 50 mhz 486 computers might exceed 50k bytes per second. One big disadvantage in parallel port cables is that they should be kept to 15 feet or less. This means that machines have to be very close together to use parallel port links.

### **ARCNET:**

If speed is high on your list of priorities then you may want to use Arcnet cards. These cards operate at a baud rate of 2.5 megabits per second. The actual transfer rate you will see is about 40000 to 80000 bytes per second which is approximately five to ten times faster than serial ports. Unless you are very impatient this may be fast enough. Just as important is the fact that Arcnet cards are designed to link many machines together in one network, rather than two machines as in serial and parallel links. The previously described methods are wired point-to-point, or Daisy-chained whereas Arcnet is BUS oriented. The difference is that ANY two computers linked on Arcnet talk directly to each other. With point-to-point schemes some computers may

have to relay your request to some other node - this causes an unavoidable delay because an intermediate computer has to act as an active hub (repeater) which requires processor time.

Arcnet will cost more money. Current market price for these cards is quite low. Cabling cost is not too expensive but tends to be higher than serial cabling. Also, you may not be able to buy an add-on Arcnet device for your laptops.

The disadvantage of course is that you have to purchase an Arcnet board for each Arcnet node. Then you have to take the cover off your computer and install them. Another disadvantage is that the marketplace is losing interest in Arcnet in favor of Ethernet, so the current pricing advantage Arcnet has over Ethernet may disappear shortly.

## ETHERNET:

Ethernet offers you the best speed. It is approximately twice as fast as Arcnet in a moderately loaded network. The cards operate at a baud rate of 10 megabits per second. The actual transfer rate you will see is about 100k to 200k bytes per second. Of course, faster machines will perform faster. But even on slow machines Ethernet speed is quite impressive. As with Arcnet, it is a BUS oriented wiring scheme, so any two machines "talk" directly to each other - there are no "relaying" nodes as with serial and parallel.

The disadvantages with Ethernet are cost and complexity of installation. Current market cost is somewhere between \$50-\$150 per card. There are many types of cards to choose from with some claiming better speed than others. This may or may not be true, but do not expect a card costing three or four times some other card to give three or four times the performance. Currently we directly support (ie, no other software or drivers needed) all 100% compatible NE2000, NE1000, and NE2100 cards. These are standards supported by many manufacturers. We also have an interface to Packet Drivers which are supplied by most manufacturers on a "drivers" disk shipped with the cards. The Packet Drivers are sometimes called TCP/IP or PC/TCP packet drivers, Clarkson packet drivers, FTP drivers, or Crynwr drivers. Using the packet driver interface we have gotten the following cards to work: 3com 3C501, 3C503, 3C507, 3C509, Intel EtherExpress 16, SMC Elite16, WD8003, and DEPCA, and some Pocket adapters. Check the README file on disk for the latest information.

At this writing we do not have an interface to IPX or NDIS but both of these are in our plans.

Don't let all this information scare you. The terms and rules may sound complicated but in most cases installing Ethernet cards is not that difficult.

## MODEMS:

Sometimes you have to link to a computer located at a remote site. Maybe you have a computer at home and want to access your "office" network to get a report, or lookup a telephone number. Or maybe you have two business locations

and you want to link them together on the same network. Or maybe you have a friend and want to copy a file to his computer.

You can buy a remote access software package to do this, but with Little Big LAN you can also access your "office" machine just as you access any other node on the network, without having to dedicate the machine for this use.

Of course speed is not very good over modems but sometimes there is no better way of doing it. And, since the remote site is accessed automatically - from placing the call to disconnecting - using the network to link to such sites is very simple from the user's point of view.

This capability makes Little Big LAN a Wide Area Network (WAN). With the cost of high speed modems dropping and their performance increasing we believe this WAN capability will make linking remote sites very practical and convenient in the near future.

## Cabling

### SERIAL:

Linking computers via serial ports requires a type of NULL modem cable. The wiring diagrams are in the CABLES section. The serial port connectors on your computer may be 25 pin XT style. Technically they are known as DB25p ("D" subminiature 25 pin male, i.e., "p" for pin). Instead your serial ports may be 9 pin AT style. Technically they are known as DB9p ("D" subminiature 9 pin). Actually either XTs or ATs can have either connector or both.

At least 99.9% of all serial ports are male. Some computers do have 25 pin female connectors, but these are usually parallel printer ports. So if you think you are one of the few who has a 25 pin female serial port, double and triple check before plugging a NULL cable into it. If you are wrong the computer may not enjoy the experience.

Since your computer's connectors are male, your cable must be female in most cases. So, to match your computer, your cable will probably be DB25s to DB25s, DB25s to DB9s, or DB9s to DB9s ('s' is for socket). We have found the best cable to be 6 wire phone cable, or screened data cable. Actually only 5 wires are needed (see cable diagrams). The cable can be quite long but if it is over 80 feet then the transmission rate may have to be reduced.

Linking several computers via serial ports can become a bit tricky. To link three machines you need two free serial ports on one of your computers. These two ports are connected to the two other computers in a "V" arrangement. You could also link all three directly to each other in a triangle arrangement but this requires 2 free serial ports on all three computers. The benefit is this avoids any intermediate node relaying delay.

### PARALLEL:

Most PC's parallel ports are DB25s (female) so the cable used to connect to

them would have a DB25p (male) on either end. The cable itself is not a standard item you can easily purchase. See the CABLES section for a wiring diagram. You can make a cable yourself although we do not recommend this if you are not an experienced cable maker. Ribbon cable works well for this purpose, although any multi-conductor cable with at least 11 conductors would do just fine. If using ribbon cable, some of the labor can be reduced by mass terminating one end using an IDC (insulation displacement contacts) type connector then all translation can occur from wires broken out at the other end using a solder type connector, or even another IDC connector if you are careful and patient, pressing the wires one-at-a-time into their various positions. You could also use a standard 25 wire serial cable with a jumper box properly wired on one end. If you get a Solderless DB25s-DB25p wiring adapter you will need no special tools.

This cable should not be more than about 15 feet long for reliable operation. However, performance may be considerably faster than a serial port connection (two to three times as fast). So this method can work well for neighboring nodes in an office or on a desk.

Linking several computers via parallel ports is accomplished just as it is with serial ports described above. But it is even more difficult with parallel ports due to the fact that generally fewer parallel ports are available on the average computer, and the computers must be in close proximity.

#### ARCNET:

Arcnet cards come in several "flavors." You will see references to Star, Bus, or Twisted Pair cards. The wiring between computers is a bit different on each type of card, either in the physical wire itself, or the logical method of wiring - usually called topology.

From Little Big LAN's point of view, all of these cards are identical. Their differences relate to wiring cost, reliability, and convenience. The software does not care which you use and, in fact, cannot detect a difference.

#### Star cards and hubs:

The original Arcnet topology was a star concept so Star cards have been on the market the longest. The Coax cable used to connect the cards is 93 Ohm RG62/U with BNC connectors on each end. These cards can be directly connected together through a cable of up to 2000 feet. But if you have 3 or more nodes each must be connected to a hub instead of directly to each other. The hub is in the middle and up to 8 coax cables radiate out of the hub to each of the computers. This is the reason this topology is called "star."

If more than 4 nodes are needed then one active hub may be used to expand up to 24 nodes.

The hub is usually a separate box, i.e., it probably is not a computer. Its sole purpose is to provide an electronic junction through which the computers can reach each other.

Hubs come in two basic types: Active and Passive.

A typical passive hub is a little rectangular box not much bigger than a matchbox and has 4 BNC connectors. Each connector must be either connected to a cable which goes directly to a computer's Arcnet card, or active hub, or it must be terminated by a 93 ohm terminating resistor. The terminator looks like a small twist-on cap. When using a passive hub, the maximum cable distance between any node and the hub is 100 feet. Remember that two cards connected directly to each other can be separated by 2000 feet. The maximum distance between passive hub and Arcnet card is much less because the passive hub does not amplify the network signals but only provides impedance matching and signal distribution for each node attached to it. Consequently, passive hubs may not be connected to passive hubs.

A typical active hub contains eight i/o ports, each of which is electrically isolated and amplified to provide digital signal regeneration and impedance matching thereby maintaining data integrity over long distances. Unlike the passive hub, unused ports on the active hub do not have to be terminated, and the maximum working distance between any node and the active hub is typically 2000 feet, or roughly the equivalent of two star nodes directly connected.

There are Arcnet cards available which also serve as active hubs. These cards usually have four BNC connectors on the back of the card.

Active and passive hubs may be used in combination for flexibility and cost effectiveness in expanding the star network arrangement.

Bus cards:

The bus card is an attractive version of the Arcnet adapter. BUS cards are available in Coax or twisted pair versions. Multiple BUS nodes are connected as points on the same wire. Logically, there is little difference between COAX bus cards and twisted-pair bus cards. The primary difference is what kind of cable and connectors are used to attach the nodes. Any jack or connector on a BUS topology is electronically the same jack - similar to a power strip where one wall outlet is transformed into multiple outlets and each new socket is electrically identical to the first. Think of a BUS as a wire strip on which there are several sockets, any of which can be used to attach a computer.

## BUS - Coax type

These cards have one BNC (coax) connector on the back (like the star cards) on which you need to attach a "T" connector. The "T" connector may be supplied with the card. This "T" connector forms a junction permitting two coax cables to be attached to the card which go to two other computers. Up to eight nodes may be connected directly in a straight line using this method. Coax cable is 93 Ohm RG62/U terminated with BNC plugs. These plug into the "T" connectors. The nodes on each end of the bus need a 93 Ohm terminating resistor to be attached to the unused "T" branch. Each end needs the terminating resistor for electronic stability. The cable, connectors, and terminator are the same as is used with star cards.

If more than 8 nodes are needed then one active hub may be used to expand up to 56 nodes.

## BUS - Twisted-pair type

In recent years there has been a cry to use existing telephone wiring to connect computers. Partly in response to this, twisted pair cards were developed and are becoming very popular. These cards have two RJ-11 connectors on the back. RJ-11 connectors are the same connectors used by ordinary USA telephones. Up to ten nodes may be connected using twisted pair cable. Some of these cards have only RJ-11 (twisted pair) connectors but others have BNC connectors too so that the same card supports both twisted pair and coax bus wiring. In either case there should be a terminator at each far end of the network bus.

If more than 10 nodes are needed then one active hub may be used to expand up to 72 nodes.

## Expanding the BUS with a HUB

Twisted pair cards can be connected in a row of ten nodes without any additional hardware other than the wiring and two terminators, one on each end. Similarly, coax bus cards can be connected in a row of eight nodes without any additional hardware other than the wiring and two terminators. In both cases if you need more nodes you must use an active hub to expand. The active hub you use for coax bus cards is exactly the same as you use with star cards. To connect a row of coax bus cards to an active hub you must take off the terminator on one end of the row of computers and run a cable from that computer to the active hub. Remember that the active hub counts as 1 node therefore only a maximum of seven or nine computers for coax or twisted pair respectively can be attached to an active hub connector. But since there are 8 connectors on most active hubs, up to 56 or 72 computers can be supported by just 1 hub.

## **ETHERNET:**

Ethernet cards come in two main "flavors." You will see references to 10base2 (coax) and 10baseT (twisted pair). The wiring for these cards is a bit different on each.

### **10base2:**

The 10base2 cards use RG58A/U coax cable using BNC type connectors. This cabling is sometimes described as Thin Ethernet or Cheapernet. Each card has a BNC twist connector on the back and a "T" connector should be attached to this connector. The cards should be shipped with a "T" connector since it is considered an integral and necessary part of the wiring. The "T" connector has two open ends to which two coax cables can be attached going to two other "T" connectors and therefore two other cards. Note that the middle of the "T" must be attached directly to the card and not indirectly through a third cable to the card!

Up to 30 nodes can be connected in this manner without using Repeaters or Concentrators. The open ends of the two "T" connectors on either end need to have a 50 ohm terminator. This terminator is absolutely necessary. Without proper termination Ethernet will not work at all. This means that while you are adding or subtracting nodes on Ethernet the fact you disconnect any part of the wiring will temporarily bring down the whole network so make sure nobody is doing anything on the network while you are modifying the wiring.

If you are linking over 30 nodes you must split the wiring into at least two segments with each segment having no more than 29 nodes. These segments are then attached to a repeater or concentrator. Concentrators typically have 4 to 10 connectors, so up to 232 nodes can be supported with one 8 port concentrator.

The minimum length of cable between nodes is 1 meter (3.1 feet). The maximum length of an ethernet cable segment is 185 meters (607 feet). A segment is a string of computers wired with no repeater.

### **10baseT:**

The 10baseT cards use Unshielded Twisted Pair (UTP), or simply twisted pair. This is similar to regular flat satin telephone wire but has 8 conductors, of which only 4 are used.

RJ45 telephone-like clips are used to connect to the cards. When using this type of card you usually have to purchase a concentrator because the cards are not designed to daisy-chain as with Arcnet twisted pair wiring. To connect even two computers requires a concentrator. It is possible to link two cards directly through a specially wired cable but when linking more than two computers you must use a concentrator. The wiring diagram for connecting only two 10baseT cards is in the Cables section of this manual. This concentrator-less cable is not a standard off-the-shelf item you can purchase, so you will probably have to wire it yourself.

Concentrators usually support 8 to 12 nodes. So if you link more nodes than one concentrator will support you must cascade concentrators.

Since you must purchase a concentrator to use 10baseT cards it is usually more expensive to use than 10base2 coax. Concentrators range in price from just over \$100 to nearly \$1000. But the wire itself is less bulky than coax and may be pre-wired in your building. There is no doubt that twisted pair wiring can be very convenient. But it may be the more expensive route so keep this in mind.

You can mix 10base2 with 10baseT wiring on the same network. But to do this you must buy a concentrator that supports both connectors. For example, you might find a twisted pair concentrator having 8 twisted pair connectors and one BNC connector for a coax segment.

### 10base5:

Both 10base2 and 10baseT cards may have an additional 15 pin connector for 10base5 "Thick Net" cabling (AUI). Thick Net is rarely used in the small LANs in which Little Big LAN will likely be used. It is usually used in large networks, like in a university where buildings are linked together. It is an expensive and cumbersome way of linking ethernet, but can traverse long distances, and support many nodes.

# Software Installation

\*\*\*\* The latest information is in README.DOC on disk! \*\*\*\*

## Introduction

Your computer provides you with a collection of resources such as a keyboard, a monitor, printers, disk drives, programs, databases, modems, etc. Most software you will ever buy is designed to let you use some of those resources on your machine. Little Big LAN is designed to let you use resources located on someone else's machine. It is therefore quite different from other software you may have purchased.

When you install or setup most software you probably have to tell the software something about your machine - for instance, whether you have a VGA or monochrome monitor. With Little Big LAN you must also describe something about the other machines to which you will be connected. So if you plan on using Bert's hard drive you must tell Little Big LAN how to find Bert's machine and which drive on Bert's machine you will be using. If you are using Ernie's hard drive you must do the same for Ernie's machine. So if you are linking five computers you must tell Little Big LAN something about the other four machines from each computer's point of view. That's right, you would have to install the network five times! You can see that installing Little Big LAN is not going to be a simple process if you are linking many machines. Unfortunately, this is the nature of networks. So if you have heretofore flown through software installations by the seat of your pants, you may have to rethink that policy.

## Theory

All networks consist of three major functions. There must be a REDIRECTOR which looks at every resource request you make and sorts out remote requests from local requests. There must be a COMMUNICATOR which does the actual "talking" between computers. And there must be a SERVER which sits on a remote computer, accepts your request and does its best to honor and answer it.

In Little Big LAN, the REDIRECTOR and SERVER both are combined into the NET21.COM file. It was named NET21 because its major function is to redirect and service DOS's INT 21H function calls. In some networks the REDIRECTOR is associated with the client machine and the SERVER is associated with a server machine. In Little Big LAN both functions are inseparable so are always present in all machines on the network.

The REDIRECTOR and SERVER do not know how to communicate at all. Little Big LAN's COMMUNICATOR is found in the combined link module file(s) named xxx\_LINK.COM. The link programs are very simple. They know how to communicate point to point only. This means that COM\_LINK, for instance, knows only how to send raw data to one other COM port, or

receive raw data from one other COM port. ARC\_LINK and ETH\_LINK are a little more advanced in that they can talk to any other arcnet or ethernet node, but they cannot communicate any farther than that. The link modules knows nothing about what is being sent or ultimately to whom. But the link module does know that for data to reach some particular node it and it alone must first send the data.

Little Big LAN needs one more module to coordinate the many link modules and physical layer interfaces it can use. That module is called the ROUTER and it also is contained in NET21. The traffic ROUTER knows whether or not a target node is directly connected to itself via one link, or else connected to a relaying node which in turn will forward the data communication request. Its first function is to accept an incoming request as its own thereby passing it to the SERVER or REDIRECTOR. Its second function is to detect an incoming request is not to itself so it must relay the request to some other node. It essentially performs a bridging function, that is, it bridges between similar or dissimilar hardware.

The REDIRECTOR must be told what resources exist outside the computer. The COMMUNICATOR/ROUTER must be told how to locate other computers on which those resources are to be found. The SERVER must be told which of its own resources it is permitting someone else to use. Many of these things can be specified in multiple ways. In simple setups everything can be specified in CONFIG.SYS and AUTOEXEC.BAT. In most setups it is necessary to put some of this information into special configuration files that Little Big LAN knows about. All of these things will be explained in detail later on. But fortunately you don't have to know most of it since our INSTALL program and utilities take care of most everything.

## Preparation

Before proceeding you should write protect the LBL distribution disk, make an exact copy, then store the original in a safe place. The best way of doing this is to run DOS's DISKCOPY. If you place the LBL distribution disk into drive A:, type:

```
C:>DISKCOPY A: A:
```

When DISKCOPY prompts you for the source disk, insert our original disk into drive A:, when DISKCOPY prompts you for the destination disk, insert a blank disk into drive A:. When finished, safely store the original.

Next you will need a bootable floppy disk. To make this type:

```
C:>FORMAT A: /S
```

When prompted, place a blank diskette into drive A:. After FORMAT finishes you need to copy all of the files from subdirectory \LBL on our Distribution Disk to a \LBL subdirectory on the System disk. We have included a utility to do this. Place the copy of LBL's Distribution Disk into drive A:, type:

```
C:>A:  
A:>MAKE A:
```

MAKE will read all files from the Distribution disk, then ask you to place the bootable System disk into drive A:. It will then copy all the files necessary for installing LBL onto the System Disk. If you had placed the System disk into drive B:, change the A: to B: in the previous example.

There is a reason for going through this step. It is usually easier to go through the installation process on a floppy disk which can later be taken to each computer for both testing and final installation. All setup files are created once for all computers. This keeps your work to a minimum, and also minimizes the opportunities for mistakes. Though this method is not absolutely necessary for proper installation, we highly recommend it. If you have 5.25 drives as A: on some machines, and 3.5 inch drives as A: on others, we recommend you make two System Disks and transfer all files in subdirectory \LBL from one medium to the other.

## INSTALL.EXE

This program will take away most of the complication of installing the network. Insert the System floppy diskette (described previously) into drive A: and type:

```
A:\>CD \LBL
A:\LBL>INSTALL
```

## List all computers

INSTALL will show you a menu of four choices:

```
Setup Network Connections
Setup Printers, etc...
Install to a Node
Install Spooler
Exit
```

Select the first choice, "Setup Network Connections," by hitting <enter>. You will see (or fill out) a screen which looks something like this:

Press F2 for  
node details

Creating/Editing Node List

###	Node Name:	COMs				LPTs				ARC	ETH
		1	2	3	4	1	2	3	4		
1	baby_286		y								y
2	tower_386			y							
7	joes_desk										y

Enter a list of all your computers. Assign them a number & name

Then put a 'y' in the appropriate columns.

Example: if using COM2 to link to a computer, put 'y' in the COM2 column for that node's row

Edit    \*\*\*\*    \*\*\*\*    Load    Save    Exit

You should create the whole list of all computers before proceeding. Assign each computer a unique node name and a unique node number. A node number is like a telephone number, it's used by Little Big LAN to contact other nodes just like the telephone company uses your number to contact you.

Node numbers must be in the range of 1 to 250. If you are using arcnet in a computer, then that node number must be the same as your arcnet card's node address as set by the card's switches. Serial port, parallel port, and ethernet nodes can be assigned any unique number.

Node names can be most anything, but try to limit them to 15 characters with no spaces. Use underscores or dashes for spaces. Little Big LAN will accept space characters but it's best not to use them.

## Specify connections

Next, for each node you must tell INSTALL what hardware you are using to link to other nodes. You should still be in the first screen of the "Setup Connections" module. You will see columns onscreen for each of the hardware types. These are your options for linking to another computer. There are columns for COM1,2,3,4, LPT1,2,3, ARCnet, and ETHERnet.

Put a "y" in the appropriate column(s).

For example, if you are using only ethernet, you should have a "y" in the ETH column for each node. In the example on the previous page there is a "y" in the ETH column for both node "baby\_286" and node "joes\_desk" indicating both of these nodes are using ethernet. Notice "baby\_286" also has a "y" in the COM1 column. This is perfectly ok because Little Big LAN can support mixing connection types. Node "baby\_286" will be using ethernet to communicate with some nodes, and using COM1 to communicate with another.

## Special case: ETHERNET

There are a wide variety of ethernet cards on the market, and INSTALL must be told which card you are using. INSTALL assumes NE2000 compatible cards unless told otherwise. Entering a "y" in the ETH column will select NE2000 cards. Notice that the ETH column allows 3 characters. This is to allow entering 3 character IDs for particular ethernet cards. Here is a list of current IDs:

y	=	NE2000	NE1	=	NE1000
ETH	=	NE2000	N21	=	NE2100
NE2	=	NE2000	PKT	=	Packet driver

If you have an ethernet card for which we do not yet have a direct driver, use the "PKT" ID. This is a general software interface we support. You must locate a packet driver for your card. Usually one will be shipped with the card, but if you have trouble finding one, contact us. You should copy the packet driver into the LBL subdirectory. This may help simplify the installation process.

We will be directly supporting other ethernet cards in the future, probably before this manual is re-printed. Therefore, this is a list of other possible ethernet IDs:

SMC = SMC Elite16	ISO = Isolan
WD8 = WD8003	DPC = Depca
ITL = Intel EtherExpress	EVX = Everex SpeedLink
501 = 3COM 3C501	NDI = NDIS
503 = 3COM 3C503	IPX = IPX
505 = 3COM 3C505	ODI = ODI
509 = 3COM EtherLink III	

For a complete list run ETHTEST. Select your card if it shows up on the menu. It may give some setup information, including the ID to use.

### Special case: MODEMS

When you are using a modem to link to a remote computer, you still must tell INSTALL which COM port the modem is using. But instead of entering a "y" to select COM1,2,3, or 4, use "m". When INSTALL sees the "m" it knows to use the MDM\_LINK driver (documented in the FILES section) rather than the normal COM\_LINK driver (also documented in the FILES section). But modem usage is not as straight forward, so you will have to edit AUTOEXEC.BAT to modify the MDM\_LINK line to add a default telephone number. See MDM\_LINK.

### Entering node details

For each node you should go to the 2nd screen by pressing F2. Position the cursor highlight anywhere on each node line and press F2. This will show you that node's details for every hardware type you have selected.

This page gives you a way of entering certain hardware specific information on each hardware type. You do not have to fill in everything for each type.

Here is a sample of the Second Screen:

#### Creating/Editing Node List

```
Node #1
Node name = "baby_286"
```

LinkType	INT	###	Name:	I/O:	MEM:	DMA
COM1	12	2	tower_386			
ETHERNET	11					

Show to which other nodes this computer is connected.

You will see LinkType names corresponding to the "y" selections you made on the first screen. You cannot change LinkType here. If you must change it, press F2 again to return first screen and position the "y" marks where they should be.

## Specifying an INT:

For all hardware types you probably need to specify an interrupt. Default values are sometimes ok, but it is best if you run our diagnostics, or your card's diagnostics to see how the hardware is configured. The INT setting is related to the IRQ but it is not the IRQ. It is the actual CPU interrupt used by the IRQ. The INT setting is reported by our diagnostics.

INT is a decimal number. It is the IRQ number plus 8 for IRQ 0..7, or the IRQ number plus 104 for IRQ 8..15. For further information on INT see the COM\_LINK, PAR\_LINK, ARC\_LINK, ETH\_LINK or Diagnostics sections in this manual.

Here are the translations from IRQ to INT:

IRQ2=	INT10	IRQ10=	INT114
IRQ3=	INT11	IRQ11=	INT115
IRQ4=	INT12	IRQ12=	INT116
IRQ5=	INT13	IRQ13=	INT117
IRQ7=	INT15	IRQ14=	INT118
IRQ9=	INT113	IRQ15=	INT119

For packet drivers this is the software interrupt the packet driver is using. The range is 60H to 80H which translates to INT96 to INT128. The default is INT96.

## Specifying "Link to node"

Only COM port and LPT port connections need this to be entered. This specifies to which node the hardware is connected. COM ports and LPT ports must know which computer is attached to the other end of the cable. Arcnet and ethernet already know they are connected to all arcnet or ethernet nodes.

You can enter the node number or the node name. INSTALL will lookup its counterpart.

## Specifying I/O

Also on this screen you can specify the hardware address of the card, for instance, COM3 could be 3E8 (use hex values). If the I/O column is left blank then default values are used. This is probably okay. This number is assumed to be a hexadecimal value. So if your manual states that the hardware port address is 300H, just enter "300".

## Specifying MEM

MEM will specify the hexadecimal memory segment of your arcnet card. Some ethernet cards (but not NE2000) may need this parameter. Again, default values are usually ok. But if you change your card's setting, show that change here; otherwise, you may leave it blank.

## Specifying DMA

Some ethernet cards use a dma channel. NE2100 cards do, but NE2000 and NE1000 cards do not. Arcnet does not use dma. This number should reflect your card's jumper settings. Valid settings are 1 to 7.

## Checking your work

After you have finished filling the 2nd screen for each node, you should let INSTALL check your work. Go back to the 1st screen and select the "Check" option from the bottom selector bar. If the cursor is in the edit box area, hit the ESC key to get down to the selector bar. Then use the right and left arrows to position the highlight cursor over "Check", and hit the <enter> key. INSTALL will examine all of the connection choices you have specified and make sure every node knows how to reach every other node with your setup. If it finds a problem it will attempt to tell you what is wrong.

## Saving your work

Once INSTALL checks your setup and everything appears ok, you should save everything. Again, get to the bottom selector bar and position the cursor over the "Save" option. Hit the <enter> key. Several files will be created and stored to the disk, so it may take a while. There will also be one batch file created for each node to help in the testing process.

## The First Test

If you have run install from a System floppy disk as we suggest, that floppy has everything Little Big LAN needs for the setup on every node. It also has simple CONFIG.SYS and AUTOEXEC.BAT files for every node. It is a simple matter to now establish and test a network connection on all the nodes.

You should take this one System Floppy disk to each machine. Change to the A:\LBL subdirectory, then type the node number of the computer at which you are sitting. New skeleton CONFIG.SYS and AUTOEXEC.BAT files will be created in the root directory on drive A:. Then reboot from that System floppy disk. If you do this on all of your machines, the network should be active.

Run the LBL.COM utility to setup some network drives, then test to see if everything works. Try redirecting drive H: to another node's drive and do a DIR. You may want to read about LBL.COM in the FILES section.

If you have a mixture of drive A: media types - some 5.25 and some 3.5 inch drives - then, before testing, copy all of the files in subdirectory \LBL to a second System disk of the second medium type.

Remember when you boot from a floppy not all of your hard drive partitions may be active - depending on the DOS versions. Also things like CDROMS will not work. This is normal. The purpose here is to verify the network can talk between nodes, not that the network can use all of your resources as yet.

Also, booting from the "clean" setup the System floppy contains will minimize the chances of a software or hardware conflict which may exist if you boot up with your system fully loaded. Having this conflict first would tend to confuse and over-complicate things, making the solution harder to find.

## Specifying printers

From the main menu of the INSTALL program select the second option, "Setup Printers, etc...". The first thing you will be asked is which node's setup you will be modifying.

This module of the INSTALL program is designed to redirect printer (character device) requests to another node where the device really is. So if Jack's computer is needing to use Mary's printer then it is Jack's node you will have to modify, not Mary's. Hopefully Mary's computer already knows how to use its own printer.

When answering the node question, you can enter the node number or node name or partial node name. Whatever you enter must be valid. That means you must enter nodes which you have already told INSTALL exist in the "Setup Connections" section, and you must have saved that setup.

After verifying the node you specified does exist, a current list of all redirectable devices on that node will be shown. You can now add to the list, modify a member, or delete a member. You will be asked if you want to add or modify anything first. If so, you will be asked for a device name. The name you enter is the local name of the device from this node's point of view. This is the name your software thinks it will be printing to. The next question you will be asked is the target device. You might think you just answered that, and this seems to be one of the biggest sources of confusion to many customers. So we will try to describe the question as best we can: This is the name by which the real device is known to the node on which the device is physically connected. In different words, it's the real device name known to the node which really has it, or, better yet, the name you've always called it before you had a network. The problem with describing the answers to "name" questions is that devices such as printers can have many names, or "aliases" on the network. You are telling the network how to translate those aliases to a physical device. To complete the target description, you will be asked the target node. This is the node the target device is really on. These "target" parameters are the defaults Little Big LAN uses when your computer boots. They can be changed later very easily with the LBL.COM utility.

The last question you will be asked is if you want redirection to occur automatically each day when you boot (turn on) your computer. Again, this does not imply permanence, only convenience. Redirection might very well be best turned off for some devices. For example, you might permit redirection of LPT1 here even though you already have a dot matrix LPT1 on your machine. But this ability might prove useful when you occasionally need to send your print job to, say, a laser printer on another node.

If you have no more devices you wish to add or modify, you will be asked if you wish to delete any of them. If so, you will be asked for the device name.

## **Final Install**

After you have setup and tested each node, it's time to install the network to your hard drives. You might ask why you should setup everything for all nodes on one common floppy diskette first before getting to this step. There are three reasons. First, it is easier to test and modify temporary setups if everything is kept in one place. If everything were spread among several disks it would become confusing, and more mistake prone. The second reason is that it dramatically reduces duplication of work. Describing the network individually to each machine might become tedious. The third reason is that future changes and modifications are much simpler if all information is kept together. If next month you add another computer to the network you'll be glad everything is safely modified in one place. So don't lose the system diskette. In fact, make a copy or two.

The final install is a simple matter of taking the system floppy diskette to each computer and running INSTALL. Select the third option, "Install a Node." You will be asked a few simple questions.

First, you'll be asked where the source drive is. INSTALL needs to know which drive has the system diskette, and what path contains the system files. This should normally be A:\LBL or B:\LBL.

Next you'll be asked for the destination path. This is where you want INSTALL to put the network program files and configuration files. Normally you would want these files placed on your hard drive in subdirectory \LBL, so the default is C:\LBL.

INSTALL will then copy all program files from the source path to the destination path. If those files are already there then you will be asked if you want to skip this step.

Next, INSTALL needs to modify CONFIG.SYS and AUTOEXEC.BAT, so you will be asked the boot drive. The answer should be A: or C:. Then you will be asked the node number of this computer. If the network has been previously installed the node number is known, but you will be prompted to verify it is still the same. With these two questions answered INSTALL copies several configuration files to the destination path. Finally you will be asked if it is okay to modify your CONFIG.SYS and AUTOEXEC.BAT files. If you don't intend on doing this manually yourself, answer 'Y' in both cases. Don't worry, your original files are copied to ".OLD" files in \LBL and nothing is removed from the originals, except, possibly, old LBL stuff.

If you have a printer on the node, you may want to "Install Spooler" before exiting INSTALL. This is not mandatory and can be done later.

INSTALL is now finished, so exit back to DOS. After rebooting, Little Big LAN should be working.

## Manual (expert) mode

This section is provided for informational purposes. There is really no need to manually install the network. Unless you are just curious, there is no need to read the next 7 pages.

Put the Little Big LAN diskette into drive A. Create a subdirectory on your hard drive named \LBL and copy the distribution files into it.

```
C:\>CD A:\LBL
C:\>MD C:\LBL
C:\>CD C:\LBL
C:\LBL>XCOPY A:*. * C: /S
```

Using a text editor or word processor in non-document (ASCII) mode, edit your CONFIG.SYS file by adding the following lines to it:

```
DEVICE=LBL\NETUNITS.SYS
DEVICE=LBL\NET00000.SYS #<node number> "<node name>"
```

Where: <node number> = 1 thru 250  
<node name> = ASCII string in quotes

The NETUNITS.SYS device is the preferred replacement for the DOS LASTDRIVE function. By default it will add enough drive letters to give you drive Z. The NET00000.SYS device acts as the network device chain terminator and is used to assign the global (absolute) node number to the local system. You must assign a unique node number to both machines. You may also optionally specify an ASCII alias for the local system using the <node name> parameter.

Then edit your AUTOEXEC.BAT file by adding the following lines to it:

```
NET8                               = timer tick service
COM LINK COM1 INT12 LINKTO:7      = com1 linked to node 7
NET21                              = redirector + server
```

The above example uses the serial port driver COM\_LINK.COM using COM1 and hardware IRQ4 (which is CPU interrupt 12). If you are using COM2 then probably you are using INT11. If you are linking machines via parallel ports or ARCNET cards substitute the parallel port driver PAR\_LINK, or the ARCNET driver ARC\_LINK instead of COM\_LINK. Ethernet would use ETH\_LINK provided you are using NE2000 cards, otherwise PKT\_LINK. Typically you would use:

```
PAR_LINK LPT2 INT15 LINKTO:7
- OR -
ARC_LINK INT10 LINKTO:7
- OR -
ETH_LINK INT11 LINKTO:7
```

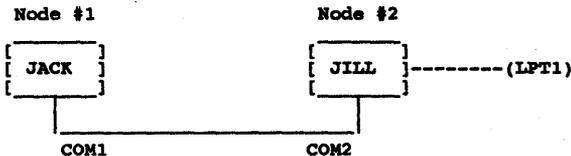
Then reboot your computers, run LBL.COM to setup drive redirection. If you want to share printers add NETBSHAR lines for each one (see description in FILES section).

## Configuration Examples

Perhaps the best way to explain what setup modifications to CONFIG.SYS and AUTOEXEC.BAT you must make is to give a few examples.

Here we will describe a typical two node setup where the machines are linked together through COM ports. We will arbitrarily call the machines Jack and Jill. Jack we will assign the node number 1, Jill we will assign the node number 2. Jill will have a printer attached to LPT1 and Jack will be using that printer. So assume:

Jack = node #1  
Jill = node #2  
Jill has a printer on LPT1  
Jack is using COM1 to talk to Jill  
Jill is using COM2 to talk to Jack



These would be the additions you would have to make to the DOS system files:

Jack's CONFIG.SYS,

```
DEVICE=LBL\NETUNITS.SYS
DEVICE=LBL\NETBSHAR.SYS LPT1 LPT1 ON:2
DEVICE=LBL\NET00000.SYS #1 "Jack"
```

Jack's AUTOEXEC.BAT,

```
LBL\NET8
LBL\COM LINK COM1 INT12 LINKTO:2
LBL\NET21
```

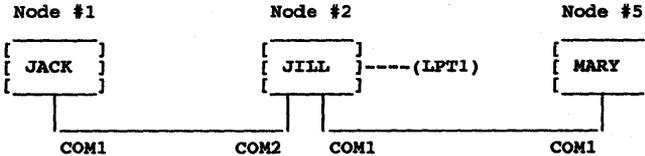
Jill's CONFIG.SYS,

```
DEVICE=LBL\NETUNITS.SYS
DEVICE=LBL\NET00000.SYS #2 "Jill"
```

Jill's AUTOEXEC.BAT,

```
LBL\NET8
LBL\COM LINK COM2 INT11 LINKTO:1
LBL\NET21
```

Now suppose we add a new computer, called Mary, to the network. We will assign Mary the node number 5. We could have assigned Mary any node number except 1 or 2 which are already being used by Jack and Jill, but 5 is what we chose for no good reason. Assume that Mary is using COM port 1 to link to Jill's COM port 1. Also assume Mary has no printer and wishes to use Jill's.



These are the modified files for all three machines:

Jack's CONFIG.SYS,

```

DEVICE=LBL\NETUNITS.SYS
DEVICE=LBL\NETBSHAR.SYS LPT1 LPT1 ON:2
DEVICE=LBL\NET00000.SYS #1 "Jack"
  
```

Jack's AUTOEXEC.BAT,

```

LBL\NET8
LBL\COM LINK COM1 INT12 LINKTO:2
LBL\NET21
  
```

Jill's CONFIG.SYS:

```

DEVICE=LBL\NETUNITS.SYS
DEVICE=LBL\NET00000.SYS #2 "Jill"
  
```

Jill's AUTOEXEC.BAT:

```

LBL\NET8
LBL\COM LINK COM2 INT11 LINKTO:1
LBL\COM LINK COM1 INT12 LINKTO:5
LBL\NET21
  
```

Mary's CONFIG.SYS,

```

DEVICE=LBL\NETUNITS.SYS
DEVICE=LBL\NETBSHAR.SYS LPT1 LPT1 ON:2
DEVICE=LBL\NET00000.SYS #5 "Mary"
  
```

Mary's AUTOEXEC.BAT,

```

LBL\NET8
LBL\COM LINK COM1 INT12 LINKTO:2
LBL\NET21
  
```

Unfortunately there is no way JACK can talk to MARY or vice-versa with this simplified setup. At this point the CONNLIST files must be used (described next) in order to tell the network an indirect link exists between JACK and MARY.

## CONFIGURATION FILES

Little Big LAN needs to know a lot about your system and how you plan on using it. When the network is operating it keeps everything it needs to know in internal memory. So there is never a need to retrieve anything from a disk drive. Nevertheless, all of that information has to come from somewhere. The configuration files provide a means of putting some of it there when the system boots.

There are three configuration files listed below:

NODELIST.LBL  
DRIVLIST.n  
CONNLIST.n

All of these files will be described in detail later. First you should know what they have in common. All should be in subdirectory \LBL, or other "home" directory you specify via NET00000. CONNLIST and DRIVLIST must have as a file extension the node number they belong to. For example, if you are on node 6 then the CONNLIST file should be named CONNLIST.6, and if you are on node 99 then the CONNLIST file should be named CONNLIST.99, etc. By using the node number as the extension it excludes that file from being read by any other node. When NET21 loads at boot time it attempts to read both CONNLIST and DRIVLIST. NET21 tries to find both in the same way. We will describe CONNLIST here, but DRIVLIST is found via the same method.

NET21 first looks for the CONNLIST file in the current directory. Next it looks in LBL's "home" directory which is normally \LBL on the boot disk. So NET21 makes two attempts to find the CONNLIST file. If we are speaking about node 99 then the search sequence is:

First try: CONNLIST.99  
then try: \LBL\CONNLIST.99

The utility program named LBL also reads CONNLIST and DRIVLIST using the opposite search order. The reason the order is different is because the network is not active until after NET21 loads and there may be times - especially on a floppy based workstation - that all your configuration files will be kept on a remote drive after booting but must be on the local drive while booting. But after the network loads the preferred search order should be: look in the "home" directory first, because the current directory may be someone else's "home" directory.

LBL also reads NODELIST. The search order for node list is identical. But because of the nature of the files, NODELIST should be using the common extension ".LBL", while CONNLIST and DRIVLIST should be used in their node number specific extensions.

## File structure

Although all of the files are standard ASCII text files, there is normally no reason for you to modify them directly. Either the installation programs or utilities such as LBL.COM will do that for you. But if you are interested we will describe their structure in detail.

The structure of all the configuration files is fairly flexible. They all contain several lines of columnar data. Each line of the file ends in a normal carriage return, and line feed. Each line composes the columns. The columns must be in a specific order but the exact spacing is not important. When the files are first opened, one line at a time is read and discarded until a line is read which starts with a dash. That line should be composed only of dashes and spaces. A sequence of dashes indicates the position of a data column field, while a sequence of spaces indicates the end of a column field. Every line thereafter will be assumed to fit this model and data will be taken from the line at the defined column field positions. Short lines will be padded with blanks. Each file needs specific information in each column and this will be explained next.

## NODELIST.tbl

The Node list is not used by Little Big LAN itself, but it is used by various utilities, chiefly the installation program and LBL.COM. It provides a list of all the computers on the network. Each computer should have a node name and a node number. The node name can be 24 characters long and can contain spaces. But because DOS expects NETBIOS names to be no more than 15 characters you may want to limit their length. Also, spaces might cause a problem with future NETBIOS or other utilities so you might want to use dashes or underlines instead. The node number can be any integer from 1 to 250. Each node name and node number must be unique, i.e., no other computer can have the same name or number.

The file can be created by any standard wordprocessor as long as it can create a standard ASCII file. The format of the file is fairly flexible. It must have two columns. The first column is a node number, the second column is the node name. The columns can have header information. A line of dashes and spaces starts the columns. That line's grouping of runs of all dashes and runs of all spaces delineates the spacing of the columns. All lines preceding this line are ignored and considered a header. All lines following this line are considered columnar data.

### EXAMPLE:

Node#	Node name
1	Baby 286
2	Tower 386
5	desktop 286
10	Toshiba 1000
64	Color xt

## CONNLIST.n

The Connection list is read by NET21.COM when it first loads and provides a list of information on how to reach any node from the current node. Essentially it is a road map the network uses to get from one node to another. LBL.COM also creates/modifies/uses this list. CONNLIST is extremely important to the network. It provides all the information Little Big LAN needs to communicate. A separate CONNLIST file should exist on each machine on the network.

Little Big LAN's ability to connect computers through various means is more flexible than most networks. So Little Big LAN must be told what hardware to use to get to any particular node. CONNLIST is the only method of specifying all of this information.

To communicate with any node the network needs to know to which computers it has a direct link, and which nodes can only be indirectly reached, i.e., a computer which can be reached only by going through another computer. A separate CONNLIST file is kept on each computer on the network. Since each computer will see connections to the other nodes from a different point of view, all CONNLIST files may be different from each other. The only case where they could all be identical is if all were connected via ARCNET cards.

Since CONNLIST is a list defining how one node can reach all other nodes, the file has one line for each computer on the network. So CONNLIST.99 describes how node 99 can reach all other nodes. The first two columns in each line specify a computer by node name and node number. The second and third columns tell the network which link module to use to reach that computer from this one. The link module number is used for internal purposes and should normally be left to the system to generate. If you are interested, the link module numbers are defined as:

- 21 = COM1
- 22 = COM2
- 23 = COM3
- 24 = COM4
- 31 = LPT1
- 32 = LPT2
- 33 = LPT3
- 40 = ARCNET
- 50 = ETHERNET

The link module name is the normal name of the link hardware, for instance, COM2 or LPT1, or ARCNET. This link is the first hardware used to reach the target node.

The last two columns are used if the target node is not directly linked. This is the routing information. If the columns are blank it indicates the link module provides a direct link to that node. Otherwise the routing information indicates all communication to the target node should be routed through the node specified by these last two columns. Because of its own CONNLIST file, that routing node, in turn, will know how to relay that request.

The format of the file is fairly flexible. It can be created or modified by any standard wordprocessor as long as it can use a standard ASCII file. It must have six columns. The first column is a node number, the second is the node name. The next four columns tell the network how to get to that node from here. The columns can have header information. A row of dashes starts the columns and spaces delineate the columns.

**EXAMPLE:**

To get to Node: ### Name (of Node)	Use Link Module: ### Name (of LINK)	Route Thru node: ### Name (of Routing Node)
1 baby 286	21 COM1	2 tower 386
2 tower 386	21 COM1	
5 desktop 286		
10 toshiba 1000	22 COM2	
64 color xt	21 COM1	2 tower 386

**DRIVLIST.n**

One of the principal uses of a network is to allow sharing of drives. Some drive letters already refer to local drives. You can use any of those or the remaining drive letters to reference some other computer's drives. For instance, you may want letter "W" to refer to Jack's C drive. If so, then you may want the network to boot up knowing this, otherwise you would have to setup the drive references manually every day. The Drive list is used by Little Big LAN to specify the default network drive letters so that you don't have to do it more than once.

When NET21.COM loads at boot time it reads the DRIVLIST file to learn which drive letters you are using to reference which remote drives. DRIVLIST does not make these network drive references permanent. Any drive can be reassigned later on by using the LBL utility program. In fact, there is no need for this file to exist at all if you do not have a standard setup of network drive references which you want loaded every day. This file just makes it more convenient.

The DRIVLIST file should be created and modified by the LBL utility program. It can also be created by any standard wordprocessor as long as it can create a standard ASCII file. The format of the file is fairly flexible. It must have four columns. The first column is a drive letter you will be using to reference a remote network drive, the second and third columns are the node number and node name of the computer on which the target drive resides, the fourth column is the target drive and a default path. The columns can have header information. A row of dashes starts the columns and spaces delineate the columns.

**EXAMPLE:**

Drive	Node#	Nodename	Drive/path
F:	10	baby xt	C:
G:	2	tower 386	C:
J:	10	baby xt	A:
V:	10	baby xt	C:

# Diagnostics

Wouldn't it be nice if everything worked the first time, every time! Unfortunately, many of you are going to have problems getting Little Big LAN to work on your first try. We have therefore included some programs which will help in diagnosing the trouble.

Another important reason to run these tests is to determine various parameter settings prior to installing the network. Some questions you might not know how to answer during the installation process may become clear if you run these tests first.

These programs are hardware tests and do not require the network to be installed. In fact, it is best to run them prior to installing the network. To require an operational network before using hardware diagnostic routines would be something like putting the cart before the horse. Also, it is likely that running these diagnostic programs will bring down a perfectly working network. So, to be safe, reboot the network after using any of the diagnostic routines.

You will not have to run all these tests. Each is designed for a particular hardware type. So, if you are using COM ports to link computers, you'll need to run the serial port test, if using ethernet cards run the ethernet test, etc...

The tests include:

- 1) ETHTEST, for Ethernet cards
- 2) ARCTEST, for Arcnet cards
- 3) PARTEST, for parallel ports
- 4) COMTEST, for serial ports
- 5) PKTEST, for ethernet packet drivers
- 6) WATCH, for all drivers

Also of value is the NODE utility which is documented in the FILES section. With this you can determine if the network is installed at all, or if it is installed as you expect.

## **ETHTEST.COM - testing your Ethernet cards**

The program ETHTEST is a diagnostics utility which will test your ethernet cards to assure they are functioning properly for Little Big LAN. As of the printing of this manual only NE1000 or NE2000 cards are supported by this test. These cards are an industry standard and are widely available at a low cost. Therefore we have chosen this type of card to be the primary card supported by Little Big LAN. We do support other cards and for a complete list read the ETHERNET.DOC file on the distribution disk. Or you can run this program since it will supply a list of cards and provide some information on many.

To run ETHTEST type:

```
A>ETHTEST [INTx] [IRQx] [ETH:hhh] [8BIT] [TYPE:x]
          [NE2000] [NE1000] [NE2100] [NODE:x] [?]
```

You probably should not specify any of the parameters. ETHTEST will try to sense your card and does a good job of figuring out everything on its own.

Simultaneously run ETHTEST on all of your machines which are using ethernet cards. Be sure all cabling is hooked up properly. If using coax cards, be sure a terminator resistor is connected to the two end nodes.

Upon loading, if not told via the command line which type of card you are using, a list of cards will be displayed. Select your card type and the test will continue if that card is supported by Little Big LAN drivers. If it is not directly supported, then ETHTEST will direct you to an alternate test and setup procedure, and may provide further information.

Ethernet cards must be given an arbitrary node number. This is so the cards can address each other during the test. You will be prompted for that number as soon as ETHTEST continues (provided you do not enter it on the command line via the NODE:x parameter). So just give each computer a unique node number. Any number between 1 to 254 is OK.

Next ETHTEST will try to determine the current jumper settings of the card. You will see it go through a list of tests until it finally prints a line showing how it thinks you should enter parameters on the ETH\_LINK (or alternate driver) line in AUTOEXEC.BAT. Maybe you should write down this information for use with INSTALL.

After ETHTEST has recognized your card it waits for you to press any key, then prints a screen that looks something like this:

```
Little Big LAN  ETHERNET Test Program

#####
### Active Nodes
### 0123456789 0123456789
###
### 000 .X..... ..S..
### 020 .....
### 040 .....X...
### 060 .....
### 080 .....
### 100 .....
### 120 .....
### 140 .....
### 160 .....
### 180 .....
### 200 .....
### 220 .....
### 240 .....
#####

TESTING ETHERNET Node# 17 (11 Hex)
ARCNET CARD Address = 0300H
CPU Interrupt = 113 (71H)      IRQ = 9

Press 'Q' to Quit
    'T' to Transmit
    'R' to Restart
    'S' to toggle sound, now ON

Status:
Broadcast #9

CPU speed = 760
```

The "Active Nodes" box has a status position for all possible 254 nodes on ethernet. There should be a "S" at the position of this node's address as

entered by you earlier. In the above example, this node's number is 17, so the 17th dot position is changed to an "S" indicating SELF. Any other nodes on the ethernet network also running this test will have their bit positions changed to an "X" indicating they are alive. Every node running ETHTEST should know about the existence of every other node. If this is not the case then there may be a cabling problem from that node, or there may be a problem with the card.

One test you may now perform is a transmit test. Hitting "T" will cause ETHTEST to prompt you for a node address to which you wish to send. Then 100 blocks of data will be sent to that node. You should see the progress of the test on both computers, and no others. Each block is verified for true data on the receiving end so if there is a data integrity problem this test should catch it. You can send a different number of blocks by entering the desired count. For example, if you want to send 1000 blocks to node 7 then when prompted for the target node enter:

Target node: 7,1000

If you wish to send an infinite number of blocks enter \*7 or 7,0. This will send blocks forever or until you strike any key.

If you hit the "R" key it forces this node to restart the test from the second screen.

To return to DOS strike "q".

## **PKTEST.COM - testing Ethernet packet drivers**

Little Big LAN must use the packet driver (or other software) interface for many ethernet cards. Manufacturers usually supply these drivers with the cards or they will make them available on their support BBS. These drivers are sometimes called the Clarkson Packet drivers, the TCP/IP packet drivers, FTP drivers, or the Crynwr drivers.

PKTEST is designed to allow you to test the ethernet through the packet driver interface. It is compatible with ETHTEST. You can mix different manufacturers' cards on the network and with ETHTEST and PKTEST you can test this mix, transmitting test data among the cards.

To run PKTEST type:

```
A>PKTEST [INT:x] [NODE:x] [?]
```

It is probably best not to supply any of the parameters. If using INT, note that this refers to the software interrupt the packet driver is using, and does not refer to the IRQ or CPU interrupt of the card. The default is INT96 which is equivalent to 60H used by most packet drivers.

When PKTEST loads it behaves very much like ETHTEST. Refer to the preceding section on ETHTEST for a detailed description.

## Loading the Packet Driver:

Before running PKTEST you must load your packet driver. This is because, unlike ETHTEST, PKTEST does not "talk" to the card directly. It talks to the packet driver only. Therefore the packet driver must be loaded first. You can either put the packet driver in AUTOEXEC.BAT or you can run it from the DOS prompt. Follow the directions that come with the packet driver. However, most packet drivers use a similar syntax. The general syntax is:

```
pkt_drv_name pd_int [irq] [ioaddress] [memseg] [dma]
```

The "pd\_int" parameter is the software interrupt you wish the driver to use to communicate to upper level programs such as LBL and PKTEST. This interrupt number should be 60H. The options allow using any interrupt from 60H to 80H. But it is probably best to use 60H.

The "irq" parameter is the interrupt request setting of the card. Some cards permit the driver to read this setting from EEPROM on the card.

The "ioaddress" parameter is the hardware I/O address of the card and is probably jumper selectable to four or more options. Again, some drivers will not need this parameter, especially if you use the default setting of the card.

The "memseg" parameter tells the driver where to find the card's onboard memory, and may be required. But many cards do not use onboard memory, or use I/O addressing of that memory so the parameter is irrelevant.

The "dma" parameter tells the driver which DMA channel to use. This will be needed by few drivers, but if needed, should reflect the card jumper settings.

If parameters are entered in hexadecimal, you have to use the "C" language form, ie, precede the number with "0x", as in "0x300".

Most packet drivers are based on common code which has a bug preventing LBL from changing the local station address. We supply a program called ETHNAME which corrects that bug. Please read the section on ETHNAME in the UTILITIES section of this manual.

Here are a few setup examples. All examples assume the computer is to be node 100, I/O address 300H, IRQ 3, and memory address D000H.

For NE2000:

```
NE2000 0x60 3 0x300  
ETHNAME NE2000 NODE:100 ETH:300
```

For WD8003:

```
WD8003E 0x60 3 0x300 0xD000  
ETHNAME WD8003 NODE:100 ETH:300
```

For Intel EtherExpress 16:

```
EXP16 0x60 0x300
```

For SMC Elite16:

```
8003PKDR /r:D000 /b:300 /i:3  
ETHNAME SMC ETH:300 node:100
```

For 3COM EtherLink III:

```
3C5X9PD 0x60
```

Some of the packet drivers shown above are not shipped with the cards but are available on the manufacturer's Bulletin Board Systems (BBS). Also, some cards, such as the Intel card, may require you to run a setup program to install the cards for your system. For example, the Intel packet driver does not auto-sense the connector type so it will fail until Intel's SOFTSET is run to switch the card from auto-sense to BNC or Twisted pair. Note also that the NE2000 packet driver is often called by another name, as the others may be.

## **ARCTEST.COM - testing your Arcnet cards**

The program ARCTEST is a diagnostics program which will test your Arcnet cards to assure they are functioning properly for Little Big LAN.

To run ARCTEST type:

```
A>ARCTEST [INTx] [ARC:hhh] [SEG:hxxx] [IRQx] [?]
```

You probably should not specify any of the parameters. ARCTEST will try to sense your card and does a good job of figuring out everything on its own. You will see it go through a list of tests until finally it prints a line showing how it thinks you should enter parameters on the ARC LINK line in AUTOEXEC.BAT. Maybe you should write down this information for use with INSTALL.

Run ARCTEST on all of your machines using ARCnet simultaneously. Be sure all cabling is hooked up properly.

After ARCTEST has recognized your card it prints a screen that looks something like this:

```

Little Big LAN  ARCNET Test Program

#####
|          Active Nodes          |
|  ###  0123456789 0123456789  |
|-----|-----|
| 000   .X.....   .....S..   |
| 020   .....   .....   |
| 040   .....   .....X..   |
| 060   .....   .....   |
| 080   .....   .....   |
| 100   .....   .....   |
| 120   .....   .....   |
| 140   .....   .....   |
| 160   .....   .....   |
| 180   .....   .....   |
| 200   .....   .....   |
| 220   .....   .....   |
| 240   .....   .....   |
|-----|-----|
CPU speed = 760

TESTING ARCNET Node# 17   (11 Hex)
ARCNET CARD Address = 02E0H
CPU Interrupt = 113 (71H)  IRQ = 9
Memory Buffer SEGMENT = D000H

Press 'Q' to Quit
      'T' to Transmit
      'R' to Restart
      'S' to toggle sound, now ON

Status:
Broadcast #9
#####

```

The "Active Nodes" box has a status position for all possible 255 nodes on ARCnet. There should be a "S" at the position of this node's ARCnet node address as set by your DIP switches on the card. In the above example, the card's node number is 17, so the 17th dot position is changed to an "S" indicating SELF. Any other nodes on the ARCnet network also running this test will have their bit positions changed to an "X" indicating they are alive. Every node running ARCTEST should know about the existence of every other node. If this is not the case then there may be a cabling problem from that node, or there may be a problem with the card.

One test you may now perform is a transmit test. Hitting "T" will cause ARCTEST to prompt you for a node address to which you wish to send. Then 100 blocks of data will be sent to that node. You should see the progress of the test on both computers, and no others.

If you hit the "R" key it forces this node to do an ARCnet reconfigure command. All other cards should then do the same.

To return to DOS strike "q".

## PARTEST.COM - testing the parallel ports and cable

The program PARTEST is a diagnostics program which will test your parallel ports and your cable to assure they are functioning properly for Little Big LAN.

To run PARTEST type:

```
A>PARTEST [LPTx] [INTx] [IRQx] [?]
```

Run PARTEST on both of your machines simultaneously. Be sure the cable is hooked up to the parallel (printer) ports you wish to use. If you know which port you are using you can specify it, otherwise PARTEST will go into a scan mode looking at all the ports until it senses activity on one of them. If you don't specify the interrupt, PARTEST will test the most likely possibilities.

Remember, you are testing two ports, one on either end of the cable!

Once the port number is sensed (or specified) you will see a screen which looks something like this:

```
Little Big LAN Parallel Port Test program

Testing LPT3 at hardware address 278H

Strike "Q" to Quit
Strike "T" to Transmit
Strike "R" to Restart
Strike "S" to toggle Sound, now ON

Current data on port: 08          Interrupt: 13  Irq: 5
Values detected so far: Y.....Y.....
Acknowledge status: .....

Lines states detected: .1...  Individual states: .1...
                   00000                      .....
```

If you started PARTEST in the scan mode it shows to which LPT number the cable is attached. If an interrupt has been detected you will see "Interrupt" filled in with the proper value. Jot down these two numbers for use in INSTALL.

The "Current data" report should alternate between "00" and "08".

Now strike "t" from one of the machines to initiate a transmit test. All 32 possible values are sent to the other machine sequentially. On the receiving computer you should see several signs of life. The "Values detected so far" line will slowly be filled in with "Y"s indicating the values it has seen. The "Current data on port" value will count in hex from 0 to 1Fh. There are two reports which might help diagnosing cable or port problems if they exist. The "Line states detected" and "Individual states"

reports reflect the bit positions of the data on the parallel input port. The "Line states detected" will show all lines that have been sensed "0"s or "1"s. All 5 lines should eventually show both. If not, there is a problem with that particular bit, either in the input of this machine, the output of the other machine, or the cable. The "Individual states" shows when each input has been sensed at one state while all other lines were at the other. This helps diagnose electrical shorts.

On the transmitting machine you should see the "Acknowledge status" line filled in with all "G"s indicating the other machine has acknowledged those transmitted values.

You can then start the transmit test from the other machine. This will perform the test going the other direction. If these tests pass then Little Big LAN should run. If the tests fail then there is something wrong with your cable or one of your printer ports.

You can restart the test by striking "r". You should do this on the receiving end every time prior to a test.

To return to DOS strike "q".

Before exiting, PARTEST will print a line showing how it thinks you should enter parameters on the PAR\_LINK line in AUTOEXEC.BAT. Maybe you should write this information down for use with INSTALL.

## COMTEST.COM - testing the serial ports and cable

The program COMTEST is a diagnostics program which will test your serial ports and your cable to assure they are functioning properly for Little Big LAN.

To run COMTEST type:

```
A>COMTEST [COMx] [INTx] [IRQx] [?]
           [RCV] [TMT] [DSR:n] [DTR:n] [/n]
```

Run COMTEST on both of your machines simultaneously. Be sure the cable is hooked up to the serial ports you wish to use. If you know which port you are using you can specify it, otherwise COMTEST will go into a scan mode looking at all the ports until it senses activity on one of them.

It is probably best to let COMTEST detect the serial port you are using and the interrupt. So first run COMTEST without specifying any parameters.

With the optional parameters you can force COMTEST to use a specific serial port or a specific interrupt. The INT and IRQ parameters are exactly as described on COM\_LINK in the FILES section. The DTR and DSR parameters are meant for special cabling situations. Their use is described in the CABLES section and the COM\_LINK description. DTR:0 will turn off hardware handshaking.

Once the port number is sensed (or specified) you will see a screen which looks something like this:

```
Little Big LAN Serial Port Test Program
Testing COM2 at hardware address 02F8H

Strike 'Q' to Quit
Strike 'T' to Transmit
Strike 'R' to Restart
Strike 'S' to toggle Sound, now on

Current data on port: 30   DSR:1   Interrupt: 11   Irq: 03
                        RTS:0           Activity: 77
                        RLSD:0
                        RING:0

Status:
```

If you started COMTEST in the scan mode it shows to which COM number the cable is attached. If an interrupt has been detected you will see "Interrupt" filled in with the proper value. Jot down these two numbers for use in INSTALL.

The "Current data" report should alternate between "30" and "31". The DSR report should alternate between "0" and "1". Once every second COMTEST alternately transmits a "30" or "31" and toggles its DTR line. This data should show up on the "Current data" and "DSR" report on the opposite computer.

Now strike "t" from one of the machines to initiate a transmit test. Ten blocks of data will be transmitted, verified, and acknowledged. This test verifies the reliability of the link at 9600 baud. It is not an absolute guarantee your serial ports will transmit data at 115k baud. But at least 99% of the time this will be the case.

To return to DOS strike "q".

Before exiting, COMTEST will print a line showing how it thinks you should enter parameters on the COM LINK line in AUTOEXEC.BAT. Maybe you should write this information down for use with INSTALL.

## TMT / RCV - mode

(Additional RS232 TESTS)

Additional options can be used to do more extensive tests of your COM ports. Normally it is not required to run these tests if COMTEST ran OK. To run the tests:

- 1) Run COMTEST RCV on one machine.
- 2) Run COMTEST TMT on the 2nd machine WHILE RCV IS RUNNING ON THE 1ST MACHINE!

RCV runs continuously. It prints the character "z" while the 2nd machine is dormant. TMT initiates the test. TMT runs once and then exits to DOS. The programs assume you are using COM1 to communicate. If not, then enter the COM number on the command line when you run the program:

```
A>COMTEST TMT COM2
- OR -
A>COMTEST RCV COM2
```

The TMT computer initiates the test. It first tests to make sure the COM number you specify is really installed. Then a list of each serial port installed is printed out.

The 2nd test is of the data channel (pins 2-3). If this test fails then the cable may be bad, or it may not be a null modem cable, or it may be hooked up to the wrong port. Make sure you do not have two cards installed at the same port address. Unfortunately a program cannot test for this possibility. But with the proliferation of Multi I/O cards it is a common mistake to put two different cards in a computer with the same RS232 port address. If your computer has more RS232 connectors than the number of ports RCV shows you have installed then it is possible you have 2 ports at one address.

After the data line test is the DSR-DTR line test. If this test fails then the cable you have is probably missing these two lines or they are not crossed between the machines (pins 6-20 on a 25 pin connector or pins 4-6 on a 9 pin connector).

The last test is a string transfer from RCV to TMT. The test string will be printed to the TMT screen. If this message appears then the test is good and "CHECK OK" will be printed to both screens. This program may fail the first time even on good hardware so be sure to run it several times to make sure.

## **WATCH.COM - testing the drivers**

This is an additional debugging test for the LINK drivers. It is designed to show all activity the hardware driver sees, so can help diagnose a non-working LAN. To use it, the network must be installed, and you must add the word DEBUG to the xxx\_LINK line in AUTOEXEC.BAT on both of your machines.

To run WATCH type:

```
C>WATCH [LPTx] [COMx] [ETH] [ARC] [IRQx]
```

Boot LBL as normal, then run WATCH on one of the computers. You will not have to supply any of the parameters.

You should see activity on the IRQ report when the other computer is trying to access you - if not then you have an IRQ problem. Various other things will be reported.

# Files

The following pages describe all of the main Little Big LAN files. Some must be placed into CONFIG.SYS, others must be placed into AUTOEXEC.BAT. Still others are stand alone programs you run from the DOS prompt.

Most of these programs have parameters you may or may not need to enter. If a parameter is in brackets it means it is not necessary to enter that parameter. If it is not surrounded by brackets it is a necessary parameter. The ordering of optional parameters is almost never important. The case size of letters is never important because upper or lower case is always considered the same.

If a parameter includes a numeric subparameter that number in some cases may either be decimal or hexadecimal. A hex number ends with a "H" as in 200H. But some numbers can only be hex. One example is port addresses or memory segments which are almost never represented as decimal numbers.

In the syntax examples an "nn" expects a decimal number, or a hex number if ending in "H". A "x" expects a decimal number. An "hh" expects a hex number so the ending "H" is not required.

If a parameter needs a list of subparameters that list must be separated by commas, and no spaces.

If a path parameter is required you must specify the path fully, including the drive specification.

## NOTE:

There may be other parameters added since the printing of this manual so be sure to check README.DOC for the latest information.

# ARC\_LINK.COM

**Purpose:** Provides ability to link computers via Arcnet cards.

**Syntax:** ARC\_LINK [INTx] [ARC:hh] [SEG:hh]  
[LINKTO:nn,nn,...] [DEBUG] [MODx]

Every computer with an Arcnet card needs to install this driver. It provides a standard interface to the card so that Little Big LAN can use it. Place it in AUTOEXEC.BAT prior to NET21.

## INTx or IRQx

INTx specifies the CPU interrupt being used. Optionally, IRQx specifies the hardware interrupt being used. These parameters are interchangeable. Only use one of them.

Arcnet cards are interrupt driven so you must configure your card to use one of the IRQ lines your card permits. Most cards allow selection of IRQs 2,3,4,5, and 7. Most manufacturers configure the card for IRQ2 at the factory so if you cannot use IRQ2 then you must jumper the card for another selection. IRQ2 is usually a good choice on an XT. On an AT or above you may find IRQ5 to be safer since on most motherboards IRQ2 is considered special. On most ATs IRQ2 is really IRQ9. Whatever you choose, you must jumper the hardware. Software is not capable of changing your board's jumpers.

The ARC\_LINK must be told what IRQ line you are using. You do this with the INT or IRQ parameters. INT specifies the CPU interrupt which is the IRQ interrupt plus 8 for IRQ 0 thru 7, and is the IRQ number plus 104 for IRQs 8 thru 15. IRQ2 is equivalent to INT10, IRQ7 is equivalent to INT15. The IRQ parameter specifies the true interrupt line. These are interchangeable parameters. You need only use one of them. Why do we offer two parameters for the same purpose? Mainly because if you think in terms of software internals, as we do, you may prefer to use the CPU INT#, but if you prefer to think in terms of hardware you may prefer to use IRQ#.

## ARC:hh

ARC specifies the hardware address of the card in hex. Arcnet cards have a base I/O address which is usually jumper selectable. Most cards will be factory set at 2E0H. Other common addresses are 2F0H and 300H. The ARC parameter allows you to specify which I/O address your card is using.

## **SEG:hh**

SEG:hh specifies the arcnet card's RAM segment in hex. Arcnet cards have at least 2k of memory onboard which is used as a temporary holding area in packet transmission. ARC\_LINK must be told this address which is probably D000H as set by the factory but is also a jumper selectable option. SEG permits you to tell ARC\_LINK where that memory segment is located.

## **MODx**

MODx specifies the driver number (if using multiple drivers). MODx is a parameter you should stay away from using unless there is an internal conflict which should only happen if there are two arcnet cards in your computer. It specifies the Link Module Number used by Little Big LAN utilities to address the ARC\_LINK module directly. By default the arcnet driver module number is 40.

## **LINKTO:list**

LINKTO:list specifies other nodes reached directly via this driver. In other words, all nodes having arcnet cards are directly reached via ARC\_LINK. This parameter is one method you can use to tell LBL how to find a particular node. The list is either "ALL", or a list of node numbers. When this module detects a request to any of the node numbers in this list it will know it is supposed to send the request, and does so. For example:

LINKTO:1,2,4

This indicates a direct connection exists to nodes 1,2,and 4. This is an optional method of supplying a connection list. It has an inherent limitation of not being able to specify indirect links. The preferred method is via the CONNLIST file described elsewhere. A special case can be used under some circumstances:

LINKTO:All

This tells LBL that all nodes are reached via this LINK driver. It is especially useful if all your computers are linked via arcnet.

## **DEBUG**

This option turns on debugging capabilities for use with WATCH. It is helpful if there are problems with communicating over the network.

Defaults = INT10 ARC:2E0 SEG:D000 MOD40

# CDROM.COM

**Purpose:** Co-existing with CDROMS

**Syntax:** CDROM d:  
"d" is the drive letter you wish to make your CDROM

By default NETUNITS will allocate all drives through Z to the network, so no drive letter is available to the CDROM driver (MSCDEX.EXE) when it attempts to load. The CDROM.COM utility is used to circumvent this problem. This utility tells DOS and LBL which drive letter should be made available for the CDROM.

If you want your CDROM to appear as drive S: then put into your AUTOEXEC.BAT file prior to the CDROM extensions:

**CDROM S:**

This will make drive S: available to the CDROM driver when it loads. If you do not do this then you will get an error message from the CDROM driver saying there are not any drive letters remaining, therefore it will not load. Here is an example:

**CONFIG.SYS:**

```
DEVICE=\CDROM\MTMCDS.SYS /D:MSCD001 /P:300 /A:0  
DEVICE=\LBL\NETBSHAR.SYS LPT1 LPT1 ON:0 OFF  
DEVICE=\LBL\NETUNITS.SYS  
DEVICE=\LBL\NET00000.SYS #45 "CDROM 286"
```

**AUTOEXEC.BAT:**

```
\LBL\UTILS\CDROM S:  
\CDROM\MSCDEX.EXE /D:MSCD001  
\LBL\NET8  
\LBL\ARC LINK INT13  
\LBL\NET21
```

A second method of installing a CDROM requires limiting the number of units allocated by NETUNITS. You must use the @n parameter on the NETUNITS line - "n" should be a number at least 1 less than the number of free drives before LBL is installed. This number can be found by running the NODE utility. Look for the number of added LBL drives indicated by "LBL added drive count = n" report. Subtract 1 from this number and put this on the NETUNITS line and tell DOS to reserve drive Z. Then you must run LBL.COM and redirect drive Z to be on the local node's drive Z, ie, to itself.

# COMMUX.COM

**Purpose:** Allows several COM ports to share the same IRQ line.

**Syntax:** COMMUX [INTx] [IRQx] [COMa] [COMb] [COMc] [COMd]

This utility allows Little Big LAN to share the same IRQ lines between two separate COM ports. This makes it easier to set up multiple node configurations using COM ports. Normally you have to specify a separate interrupt line for each COM port you are using with COM\_LINK. This means the network might need both IRQ4 and IRQ3, which are the normal COM port interrupt lines. If you are using a modem or mouse then these also need an IRQ line. Interrupt lines are a scarce resource on a PC so expansion to 3 nodes was hard enough. Expansion to four nodes was usually out of the question. The multiplexer simplifies this problem.

COMMUX.COM shares one IRQ line with up to four networked COM ports. To do this you must set your COM port's hardware jumpers to the appropriate interrupt, then specify the same interrupt number on each COM\_LINK command line. Lastly you must enter a new parameter on the COM\_LINK line which tells each COM port to place it's interrupt line in an inactive state. Then run COMMUX.COM once, specifying the same interrupt and all of the COM ports sharing that interrupt. For example, to use COM1 and COM2 for the network sharing IRQ4, enter the following in the AUTOEXEC.BAT file:

```
COM_LINK COM1 INT12 OUT12:0
COM_LINK COM2 INT12 OUT12:0
COMMUX INT12 COM1 COM2
```

Be sure that COMMUX is run after all the COM\_LINK lines. Also, COMMUX should not be used if fewer than two COM ports are sharing an interrupt. This multiplex program is designed to run with the Little Big LAN, not alien programs or device drivers. While this may be possible, it may require a small modification to the alien program, or a small custom program to trick COMMUX into thinking it is a network program.

The INSTALL program will automatically install COMMUX if two or more network linking COM ports are sharing the same interrupt.

## INTx or IRQx

INTx specifies the CPU interrupt being used. Optionally, IRQx specifies the hardware interrupt being used. These parameters are interchangeable. Only use one of them. The INSTALL program uses INTx, not IRQx.

Default: INT12

# COM\_LINK.COM

**Purpose:** Provides ability to link computers via COM ports (serial ports).

**Syntax:** COM\_LINK [COMx] [INTx] [DEBUG] [/nn]  
[LINKTO:nn,nn,...] [MODx]  
[OUT12:nn] [DSR:nn] [DTR:nn]

Every computer that uses COM ports to link to another needs to install this driver. It provides a standard interface to the card so that Little Big LAN can use the COM port for communication. Place it in AUTOEXEC.BAT prior to NET21.

## COMx or COMx=hh or COM:hh

You can specify the COM port you are using in one of three ways. COMx specifies via COM port number, COM:hh specifies via COM port hardware address, and COMx=hh specifies via COM port while setting the card's hardware address.

There can be up to eight possible COM ports in your computer, so the COM\_LINK driver needs to know which one you are using to link to another computer. If you are using several then you must have several COM\_LINK lines in AUTOEXEC.BAT.

You can specify a COM port hardware address instead of a COM port number. The address is the hex hardware address of your card. If you had a card using base address 1F8 then you can specify COM:1F8. This is not the preferred way of specification so it should be avoided unless you fully understand it. It does force you to enter the MODx parameter.

You can also enter the COM number and the card's address together. For example, suppose you want to use COM3 to link to another machine but your BIOS does not recognize COM3. Just enter:

```
COM_LINK COM3=3E8
```

This forces the COM3 port address to be 3E8 hex which is the usual COM3 address. The usual COM4 address is 2E8.

## INTx or IRQx

INTx specifies the CPU interrupt being used. Optionally, IRQx specifies the hardware interrupt being used. These parameters are interchangeable. Only use one of them. COM ports are interrupt driven so you must configure your card to use one of the IRQ lines your card permits. Most cards allow selection of IRQs 3 or 4. Some also permit IRQs 2,5, and 7. COM1 and COM3

are usually set by the manufacturer to IRQ4. COM2 and COM4 are usually set to IRQ3. COM LINK can use any of these interrupts but you must know how your hardware is jumpered. Software is not capable of changing your board's jumpers. There are utilities in the DIAGNOSE directory which help you determine your card's current configuration. Specifically, look at QDT and COMPAT.

Once you have determined your COM ports interrupt setting, COM LINK must be told which IRQ line you are using. You do this with the INT or IRQ parameters. INT specifies the CPU interrupt which is the IRQ interrupt plus 8 for IRQ 0 thru 7, and is the IRQ number plus 104 for IRQs 8 thru 15. IRQ3 is equivalent to INT11, IRQ4 is equivalent to INT12. The IRQ parameter specifies the true interrupt line. Again, these are interchangeable parameters. You need only use one of them. Why do we offer two parameters for the same purpose? Mainly because if you think in terms of software internals, as we do, you may prefer to use the CPU INT#, but if you prefer to think in terms of hardware you may prefer to use IRQ#.

## **MODx**

MODx specifies the driver number. MOD is a parameter you should stay away from using. The system generates an internal module number and its best just to leave it be. It specifies the Link Module Number used by Little Big LAN utilities to address the COM LINK module directly. By default the COM port driver module number is the COM port number plus 20. So COM1 is module 21, COM2 is module 22, etc.

## **LINKTO:list**

LINKTO:list specifies other nodes reached directly via this driver. Since a COM port is directly linked to only one computer the list should be only one node number. This parameter is one method you can use to tell LBL how to find a particular node. The list is either "ALL", or a list of node numbers. When this module detects a request to any of the node numbers in this list it will know it is supposed to send the request and does so. For example:

```
LINKTO:2
```

This indicates a direct connection exists to node 2. This is an optional method of supplying a connection list. The preferred method is via the CONNLIST file described elsewhere.

## **OUT12:nn**

OUT12:n forces the OUT1 and OUT2 lines to a known state. Its chief use is to allow COM LINK to share interrupts. One problem PCs have is a shortage of hardware interrupts. Through the use of this parameter and the COMMUX utility one interrupt can be truly shared among several COM\_LINK drivers. Read the description of COMMUX.

## DEBUG

This option turns on debugging capabilities for use with WATCH. It is helpful if there are problems with communicating over the network.

### /nn

Use this to alter the speed of the link. This will specify the baud rate divisor. The "nn" is any decimal number from 1 to 9. The default is "/1" which is 115k baud. Both computers must use the same baud rate, so if this parameter is entered on one machine, the same value must also be entered on the other. Some selected values are:

/1	115,200 baud	<-- default
/2	57,600 baud	
/3	38,400 baud	
/6	19,200 baud	

## PARITY

Parity is normally disabled since COM\_LINK does error checking on its own. But if you want to enable it you can enter this parameter on the command line. Parity will slow down transfers by about 10%.

## DSR:n and DTR:n

For use with non-standard cabling. These specify the bits used for handshaking. Do not use these parameters unless you are absolutely sure you know what you are doing! It's rarely necessary.

There are two possibilities for DTR output:

DTR:1	for DTR=DTR	<--default
DTR:2	for DTR=RTS	

There are four possibilities for DSR input:

DSR:1	for DSR=CTS	
DSR:2	for DSR=DSR	<--default
DSR:4	for DSR=RING	
DSR:8	for DSR=RLSD	

For more information see CABLES section.

Defaults = COM1 INT12 /1 MOD21 DTR:1 DSR:2

# ETHNAME.COM

**Purpose:** Fixes name for some packet drivers

**Syntax:** ETHNAME [NODE:x] [ETH:hhh] [ethertype]

This utility is sometimes needed for the packet driver interface due to a bug in some packet drivers. You should read the packet driver section in the DIAGNOSTICS section of this manual.

## **NODE:x**

This tells ETHNAME the LBL node number of this machine. It is not needed if the network is loaded.

## **ETH:hhh**

You must tell ETHNAME the hardware address of your ethernet card. This is a hex address. The default is 300H.

## **[ethertype]**

You must tell ETHNAME what kind of card you have. Valid types are NE2000, SMC, WD8003, and DEPCA. Others may be added later, as needed.

# ETH\_LINK.COM

**Purpose:** Provides ability to link computers via NE2000/1000 Ethernet cards.

**Syntax:** ETH\_LINK [INTx] [IRQx] [ETH:hhh] [NE1000] [8BIT]  
[LINKTO:a,b,...] [DEBUG] [MODx] [TYPE:n]

Every computer with an NE2000 or NE1000 ethernet card needs to install this driver. It provides a standard interface to the card so that Little Big LAN can use it. Place it in AUTOEXEC.BAT prior to NET21.

## INTx or IRQx

These parameters are interchangeable. Only use one of them.

Ethernet cards are interrupt driven so you must configure your card to use one of the IRQ lines your card permits. Most cards allow selection of IRQs 3,4,5,9,10,11,12, and 15. Most manufacturers configure the card for IRQ3 at the factory so if you cannot use IRQ3 then you must jumper the card for another selection. IRQ3 is used by COM2 or COM4 so there may be a conflict forcing you to change the ethernet card.

INT specifies the CPU interrupt which is the IRQ interrupt plus 8 for IRQ 0 thru 7, and is the IRQ number plus 104 for IRQs 8 thru 15. IRQ2 is equivalent to INT10, IRQ7 is equivalent to INT15. The IRQ parameter specifies the true interrupt request line.

## ETH:hhh

ETH specifies the hardware address of the card in hex. Ethernet cards have a base I/O address which is usually jumper selectable. Most cards will be factory set at 300H. Other common addresses are 320H and 340H. Also, common is 360H but this may conflict with LPT1, and if so, any print request will reset the ethernet card causing the LAN to go down. The ETH parameter allows you to specify which I/O address your card is using.

## 8BIT

This tells ETH\_LINK you have placed an NE2000 card into an 8 bit slot.

## NE1000

ETH\_LINK assumes it is running on an NE2000 card in a 16 bit slot. This parameter forces ETH\_LINK to run in the NE1000 mode.

## **TYPE:x**

Most NE2000/1000 cards function identically. But there are some variations. We have found some cards using a UMC chip reset in a different manner. This card requires the TYPE:1 parameter in order to work. Other types may be added if needed.

## **LINKTO:list**

LINKTO:list specifies other nodes reached directly via this driver. In other words, all nodes having ethernet cards are directly reached via ETH\_LINK. This parameter is one method you can use to tell LBL how to find a particular node. The list is either "ALL", or a list of node numbers. When this module detects a request to any of the node numbers in this list it will know it is supposed to send the request, and does so. For example:

LINKTO:1,2,4

This indicates a direct connection exists to nodes 1,2,and 4. This is an optional method of supplying a connection list. It has an inherent limitation of not being able to specify indirect links. The preferred method is via the CONNLIST file created by INSTALL and described elsewhere. A special case can be used under some circumstances:

LINKTO:All

This tells LBL that all nodes are reached via this LINK driver. It is especially useful if all your computers are linked via ethernet.

## **DEBUG**

This option turns on debugging capabilities for use with WATCH. It is helpful if there are problems with communicating over the network.

## **MODx**

MODx specifies the driver number (if using multiple drivers). MODx is a parameter you should stay away from using unless there is an internal conflict which should only happen if there are two ethernet cards in your computer. It specifies the Link Module Number used by Little Big LAN utilities to address the ETH\_LINK module directly. By default the ethernet driver module number is 50.

Defaults = INT11 ETH:300 MOD50 NE2000

# LBL.COM - screen

**Purpose:** This is a utility which provides resource redirection management. Both drives and printers can be redirected with this program.

**Syntax:** LBL [MONO] [ACCESS]

This program allows users to configure the network for purposes of installation and setup or to change redirection "on the fly" without having to reboot. It can be run in a full screen editing mode which is described here. For details on the command line mode see the "LBL - batch" section.

## MONO

MONO forces the program to run in the monochrome mode. It is useful on laptops which emulate color cards but have no colors.

## ACCESS

This parameter changes the way LBL runs. Instead of its normal operation it enters a mode of operation which allows you to modify the ACCESS RIGHTS and SECURITY for the LAN. At the printing of this manual none of this has been implemented, but may be soon.

## Capabilities

The program provides three main capabilities. First, the primary use of this utility is to select where drive letters A-Z are to be attached, i.e., the node and drive they are to reference. The second use is to redirect print requests to any printer on the network. The third use is to tell the network how to talk to any node on the network.

Initial settings are stored in various files with filename extensions of ".LBL" or a node number.

NODELIST.LBL keeps node names  
CONNLIST.n keeps connection list  
DRIVLIST.n keeps drive redirection info

For descriptions of these files see the CONFIGURATION FILES section.

Initial printer settings are entered into CONFIG.SYS by using the NETBSHARE.SYS file. Read that section for further information.

## SCREENS - Drive redirector

When you run LBL you will see a screen which looks something like this:

Little Big LAN  
[Drive Redirector]

LOCAL Drive	R-E-M-O-T-E ###	Node Name	R-E-M-O-T-E Target Drive
A:			
B:			
C:			
D:	12	Bert's Machine	C:\
E:	15	Jack's 386	C:\
F:	15	Jack's 386	E:\
G:			
H:			
I:	16	Invoice Desk	C:\
J:			
K:			
L:	18	My Laptop	C:\
M:			

Any drive letter may be redirected to any other computer. Use this screen to select the remote node and drive to which a local drive is to refer.

Edit    Connections    LPTs/etc    Load Defaults    Save Defaults    Exit

The bottom line on the screen is used to make selections on what you want to do. Use the left/right arrow keys to make your selection, then hit <ENTER>. You can exit LBL by moving to [Exit] and hitting <ENTER>, or by hitting the ESC key twice.

If you select "Edit" you will be able to scroll through and modify data in all but the first column. Notice that the first column is a list of drive letters A thru Z. Any one of these letters can be associated with any drive on one of your networked computers. If you are redirecting a drive letter the rightmost three columns show the current network node and drive you are accessing. You can modify the target drive or node any time. The change will take effect when you exit the LBL utility. If you modify the node number then LBL will lookup the node name and fill it in for you. If you modify the node name then LBL will lookup the node number for you. If you enter a partial name LBL will find the first match and fill in the remainder of the name. If you enter an unknown name then both columns are blanked.

Remember, when LBL first loads it reads the current status from the network itself. If you want to restore all operating parameters to the initial bootup state then select "Load Defaults."

If you want to save new initial operating parameters then make your changes and then select "Save Defaults". The next time your computer boots these new settings will be in effect. But any changes you do make are always immediately effective, regardless of whether they are saved or not.

## SCREENS - Connections

If you select the "Connections" option you will see a screen something like this:

Little Big LAN  
[Node Connection List]

To get to Node: ### Name (of Node)		Use Link Module: ### Name (of LINK)		Route Thru node ### Name (of Node)	
0					
1	Jack				
2	Jill	21	COM1		
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					

Use this screen to specify which route to take to get to any particular node.  
THIS IS NODE 1 NAMED: "JACK"

Edit .... .... Load Defaults Save Defaults Main

The node connection list is specific to each individual node. In other words, each node will have different data displayed on this screen. It is the node's picture of the rest of the world so each node has its own perspective. The purpose of this screen is to tell a node how to talk to all other nodes. You will probably not need to see or modify this screen. Let the INSTALL program do it.

You can scroll through all nodes. All valid network node numbers will have data in their rows. Presence of this data tells the network which link module to use to get to that node from here.

Conceptually there are two possible ways the network can communicate with another node. One way is a direct link. This would be a computer directly connected to your computer. A COM port provides a direct link to one computer - the computer directly on the other end of the cable. Arcnet and ethernet cards provide a direct link with all other nodes having those cards. The other way to link nodes is via indirect links. An indirect link would be the case if node Jack was linked via COM ports to Jill and Jill was in turn linked via COM ports to Mary. There would be an indirect link between Jack and Mary because Jill would serve as a go-between in all communication.

The example screen above shows the network world from node number 1's point of view. That node is named Jack. The screen shows a single node accessible to Jack. That node is Jill, node number 2. It also shows that to reach Jill, COM1 must be used. It also shows this is a direct link because no

routing node is specified. If there is an indirect link then a routing node must be specified in the last two columns. This routing node is the computer through which traffic to that target node must be sent.

For a few example setups see the XPERT file on disk.

## SCREEN - LPTs/etc..

If you select the "LPTs/etc" option you will see a screen something like this:

Little Big LAN  
[Device Redirector]

LOCAL Device	R-E-M-O-T-E ###	Node Name	TARGET Device	On?
LPT1	1	Jack	LPT1	y
LPT2	3	Bert	LPT1	
LPT3	5	Bill	LPT2	y

Listed LOCAL devices can be redirected to any REMOTE device. Use this screen to select the remote node and device.

Edit .... Load Defaults Save Defaults Main

You can edit all columns but the first. The list of devices comes from your CONFIG.SYS file. Each NETSHARE.SYS or NETBSHARE.SYS file you included in CONFIG.SYS adds a device which will show up here. The default values from CONFIG.SYS command lines can be modified now. So if you have currently redirected LPT1 to be Jack's LPT1, you can change it to Jack's LPT2, Bill's LPT1, Bill's LPT2, Bert's CON, or just about anything. If you do not have a "y" in the last column then redirection is temporarily disabled. It can be re-enabled anytime by putting a "y" there.

You can restore the default values or save new ones. When saving, your CONFIG.SYS file is modified.

# LBL.COM - batch

**Purpose:** Provide redirection control via batch files

**Syntax:** LBL [UN]MAP [DRIVE] d: [TO] [NODE] #nn path  
LBL [UN]MAP device1 [TO] [NODE] #nn device2  
LBL [UN]PROTECT d:  
LBL MODEM ... (see MDM\_LINK)

LBL's second mode of operation is command line driven. You can tell LBL what redirection action you want it to perform if you include the "map", "protect", or "modem" parameter on the command line. This prevents LBL from entering the full screen mode, therefore it allows batch files to control redirection for particular applications without operator intervention.

Let's say you want to keep all of your word processing files on node 3's drive C:\WP\DOCS. If everytime you run the editor you want to be sure drive W: referred to node 3's C:\WP\DOCS, then include the following line at the top of a batch you use to run your wordprocessor:

```
LBL MAP DRIVE W: TO NODE #3 C:\WP\DOCS
```

If you want the same example to redirect LPT1 to node 4's printer you might include another line in your batch file something like this:

```
LBL MAP LPT1 TO NODE #4 LPT1
```

You can also "unmap" drives and printers:

```
LBL UNMAP LPT1
```

This would restore LPT1 to be your local printer.

Here are some EXAMPLES of valid command line use:

```
LBL map drive h: to node #99 c:\path
LBL map drive h to node #24 c:\path
LBL map h: to #4 c:\path
LBL map lpt1 to node #4 con
LBL map lpt1 to node 4 con
LBL unmap h:
LBL unmap lpt1
LBL protect w: (drive w: is write protected)
```

In the above examples you might notice the following syntax rules:

TO, NODE, and DRIVE are ignored  
If DRIVE is present there is no need for ":

# MDM\_LINK.COM

**Purpose:** Provides ability to link computers via modems

**Syntax:** MDM\_LINK [COMx] [INTx] [IRQx] [TELE:string] [/n]  
[HANGUP:x] [DEBUG] [MODE:n]  
[LINKTO:a,b,...] [MODx]

This driver permits you to use a modem over standard telephone lines to link to computers off-site. Modem linking is transparent to you. It looks just like any other hardware connection, except for speed. MDM\_LINK dials the telephone number, establishes a connection, and hangs up automatically.

INSTALL will place MDM\_LINK in AUTOEXEC.BAT if you select a COM port link with an "m" instead of a "y". MDM\_LINK must load after NET8 but before NET21.

## COMx

This specifies the COM port the modem is using. For detailed explanation refer to COM\_LINK.

## INTx or IRQx

This specifies the IRQ the modem port is using. For detailed explanation refer to COM\_LINK.

## TELE:string

This specifies the default telephone number MDM\_LINK will dial when it must establish a connection. You must give MDM\_LINK a default number unless you have a direct connection between the nodes. In this case MDM\_LINK assumes no modem is present. This number can be changed by running LBL in the batch mode. Use dashes, not spaces, to separate numbers.

## /n

This is the default computer to modem baud rate. This rate may be changed when a connection is established. This is a divisor. The baud rate is 115200 divided by this number, so:

/48 = 2400 baud  
/12 = 9600 baud  
/8 = 14400 baud

## HANGUP:x

MDM\_LINK will automatically hangup after a certain time period of non-use. This time is normally 5 minutes. You can change the hangup time with this parameter. Specify the number of seconds. To wait 2 minutes:

```
HANGUP:120
```

## DEBUG

When MDM\_LINK is in debug mode it will echo all modem responses to the screen. This may be helpful if there is a problem establishing a connection.

## MODE:

Some internal operating modes can be controlled with this parameter  
Currently only these are defined:

```
..... .100 = activity "wheel" shown
..... .000 = no activity "wheel" shown
..... .001 = baud set to connect baud
```

## Changing parameters with LBL.COM:

Use LBL.COM in batch mode to change the telephone number to something other than the default. Type:

```
LBL MODEM [COMx] [NODE:x] TELE:xxx-xxxx
```

You can also manually hangup:

```
LBL MODEM [COMx] [NODE:x] HANGUP
```

## Using DOS ECHO:

If you want to send Hayes AT commands to the modem when it is off line you can use ECHO. For instance, assuming the modem is on COM2,

To turn off sound:

```
ECHO ATM0 >COM2
```

To make the modem auto answer on ring 3:

```
ECHO ATS0=3 >COM2
```

To turn off auto answering:

```
ECHO ATS0=0 >COM2
```

# N21\_LINK.COM

**Purpose:** Provides ability to link computers via NE2100 Ethernet cards.

**Syntax:** N21\_LINK [INTx] [IRQx] [IO:hh]  
[LINKTO:a,b,...] [DEBUG] [MODx] [DMA:x]

Every computer with an NE2100 ethernet card needs to install this driver. It provides a standard interface to the card so that Little Big LAN can use it. Place it in AUTOEXEC.BAT prior to NET21.

## INTx or IRQx

These parameters are interchangeable. Only use one of them.

Ethernet cards are interrupt driven so you must configure your card to use one of the IRQ lines your card permits. Most cards allow selection of IRQs 3,4,5,9,10,11,12, and 15. Most manufacturers configure the card for IRQ3 at the factory so if you cannot use IRQ3 then you must jumper the card for another selection. IRQ3 is used by COM2 or COM4 so there may be a conflict forcing you to change the ethernet card.

INT specifies the CPU interrupt which is the IRQ interrupt plus 8 for IRQ 0 thru 7, and is the IRQ number plus 104 for IRQs 8 thru 15. IRQ2 is equivalent to INT10, IRQ7 is equivalent to INT15. The IRQ parameter specifies the true interrupt request line.

## IO:hh

IO specifies the hardware address of the card in hex. Ethernet cards have a base I/O address which is usually jumper selectable. Most cards will be factory set at 300H. Other common addresses are 320H and 340H. Also, common is 360H but this may conflict with LPT1, and if so, any print request will reset the ethernet card causing the LAN to go down. The ETH parameter allows you to specify which I/O address your card is using.

## DMA:x

NE2100 cards have a jumper selectable DMA channel. This parameter tells N21\_LINK how the cards are jumpered. Most cards will support 3,5,6, and 7. The default is DMA:5.

OTHER PARAMETERS are as described on ETH\_LINK

# NCON & RCON

**Purpose:** Provide remote control capability

**Syntax for NCON:** DEVICE=NCON.SYS

**Syntax for RCON:** RCON [DISPLAY] [ON:n]

RCON uses the remote NCON device driver to allow you to type keys to the remote computer. Just run RCON. Any key you type will be sent to the remote computer. The remote computer will use these keys as if they were entered on that machine from its keyboard. You can optionally see the screen of the other computer.

## SETUP

Put NCON in CONFIG.SYS on the machine you want to remotely control. There are no parameters. Put into CONFIG.SYS on your to-be-controlled machines:

```
DEVICE=\LBL\UTILS\NCON.SYS
DEVICE=\LBL\NETSHARE.SYS CPU2 CONX ON:nn
DEVICE=\LBL\NET00000.SYS ... (always last!)
```

If you want to remotely control just 1 computer then the ON:nn parameter should specify the node number of that machine. This is all you have to do. If you want to control another computer run LBL.COM and get into the <LPTs,etc> screen to edit the target node#.

To take control of a remote machine, just type: RCON  
This will send keystrokes to the remote machine.  
No screen data will be returned to you.

Optionally type: RCON DISPLAY  
This will send keystrokes and bring back the screen. Text only is supported right now, so graphics oriented programs will not bring back the screen to you.

Optionally type: RCON ON:nn  
This will bypass current setup and send to the specified node.

TO EXIT RCON HIT Alt-E.

REMEMBER, Mixing monitor types may cause problems. For instance, attributes or colors may not be reported properly. Also remember "Hotkeys" may not be recognized.

# NET00000.SYS

**Purpose:** Provides global control information to the network

**Syntax:** DEVICE=NET00000.SYS #n ["name"] [d:\path]  
[SYSREQ:int,id,c1,c2,c3,c4,c5]

This module is used by Little Big LAN to keep certain important control information concerning internal communication. It is also used to detect certain control errors. As such it is of little use to you other than to know it should be installed at or near the bottom of CONFIG.SYS. IT MUST BE INSTALLED BELOW ALL OTHER LITTLE BIG LAN FILES.

## SYSREQ

One thing you have noticed is that Little Big LAN is a collection of programs and files, most of which become resident in memory and stay there. These modules must be able to talk to one another and to do so they all use the same System Request Interrupt. By default this interrupt is the CPU's Interrupt 14H which is the same interrupt used by the COM port's BIOS routines. There will almost never be a conflict between this interrupt and other TSR software, but if there is, Little Big LAN is capable of using any software interrupt as the SYSTEM REQUEST interrupt. The SYSREQ parameter allows you to do this. SYSREQ is followed by 7 numbers which have the following meaning:

- |                          |                 |
|--------------------------|-----------------|
| 1) software interrupt #, | default = 14h   |
| 2) ID number,            | default = AB00H |
| 3) call 0 command,       | default = C0h   |
| 4) call 1 command,       | default = C1h   |
| 5) call 2 command,       | default = C2h   |
| 6) call 3 command,       | default = C3h   |
| 7) call 4 command,       | default = C4h   |

These parameters must be entered in order but the whole list need not be specified. It is acceptable to use a different interrupt, but leave all other parameters as they are. For those interested, in brief, register dx is loaded with the system id number, ah is loaded with the appropriate command, then the software interrupt is issued.

### EXAMPLE:

```
DEVICE=NET00000.SYS sysreq:25h,1234h,D0h,D1h,D2h,D3h,D4h
```

This sets System Request INT to 25H, ID=1234h, commands= D0..D4

If using DOS MODE for a serial printer, you will have to use SYSREQ. Try SYSREQ:25h, or SYSREQ:61h.

**#n**  
**"name"**

Because LBL can link so many machines together in many ways, it uses node numbers to distinguish one computer from another. So every computer must have a unique node number assigned to it. Node number values can be anything from 1 to 250. Node 0 is invalid, and we reserve node numbers 251,252,253,254,255. You must assign the node number using NET00000.SYS. Also, you should assign the node a name. The name must be no more than 24 characters but it is advisable to keep it to no more than 15 characters. Put the name in quotes after the node number. The node name is for your benefit and is not used by LBL at all, however, some future utilities will use it so it's best to get in the habit of using it now.

**EXAMPLE:**

```
DEVICE=NET00000.SYS #12 "Bert"
```

In this example, the node number is 12, the nodename is "Bert."

If you are using ARCNET to link machines then the node number should match your ARCNET card's node number set by the DIP switches.

**d:\path**

At times Little Big LAN needs to know where its "home" directory is. This is normally in subdirectory \LBL on the boot drive. If you installed the network on some other drive or some other directory then you must indicate the new path here. You must specify the whole path, including the drive specification.

**EXAMPLE:**

```
DEVICE=NET00000.SYS #12 "Bert" D:\NETWORK
```

# NET21.COM

**Purpose:** This is the heart of the network, containing the redirector and server

**Syntax:** NET21 [HANDLES:nn] [BUFSIZE:nn [,nn] [nn]] [SHARE:list]  
[MODE:nn] [DEBUG:nn] [HANDLES:nn]  
[ABS] [VER] [QUIT] [NOWIN] [NODESQ]

All requests to remote devices go through this program so it must be installed in AUTOEXEC.BAT before the network will work at all. It should be placed near the end of AUTOEXEC.BAT, after all xxx\_LINK files.

## HANDLES:nn

NET21 reserves space for opening 30 remote files. This is usually ok but if you run some programs you may require more. These programs will be accounting (like Peachtree), or data base programs which use alot of indexes. Sometimes Windows needs more files - especially if you open several windows and multitask. With this parameter you can force NET21 to allocate up to 255 file handles. Only rarely will you need more that 100.

## BUFSIZE

Little Big LAN allocates two sets of internal memory buffers for various uses. The size of some of these buffers has a direct effect on speed and memory use.

Two primary buffers are set to 1024 bytes, but you can change this with the first number. The minimum size is 330 bytes. The maximum size is 8192. Using 330 byte buffers will keep memory overhead low but will slow remote accesses somewhat, depending on link methods and device types. Remote floppy disks will slow down considerably with small buffers. In some cases using 4096 byte buffers will increase overall speed dramatically but also will increase memory usage by 8k. These are trade-offs you will have to choose according to your needs. We suggest trying a 1024 byte buffer. This is the default and appears to be a good all around choice.

The second number, if entered, sets the relay buffer size. Relay buffers are used, for instance, when a node has both a COM port and an ethernet link and must relay requests between the two. The default relay buffer is 400 bytes. The maximum is 4096.

The third number, if entered, specifies how many relay buffers to allocate. NET21 will try to calculate a good number - usually 5 - but sometimes you may need more in a heavy traffic situation.

If you are permitting absolute sector redirection then your buffer size must be at least as large as the largest sector on any of those drives. This is normally 512 bytes but could be as large as 8192 with huge drives under lower DOS versions.

**NOTE:**

Buffer size is a system wide parameter and must be the same on all computers.

## **ABS**

Most networks do not allow absolute sector redirection. Little Big LAN does. This means it is possible to run programs like PC-TOOLS, NORTON UTILITIES, and DEBUG and look at or even alter remote drive sectors. You may not want to allow this to occur so this option is off unless you enter the ABS parameter.

## **MODE**

Little Big LAN was designed to be as invisible as possible. It was designed to coexist with almost any other software, including other networks. MODE is used to tell Little Big LAN how it is to answer certain questions an application may ask of it. For instance, if you want LBL to always report it is not a network and should think of all remote devices just like local devices then use MODE:0. Normally there is no reason for you to use this parameter as the default values are fine.

Bits defined by MODE option:

1...	....	....	....	report remote drives
.1..	....	....	....	report illegal drives
..1.	....	....	....	report lan drives as fixed
...1	....	....	....	report remote handles
....	....	..1.	....	use stringent fcbtraptests
....	....	...1	....	use dos server calls
....	....	....	...1	report as MSNET

MODE:4000h = no network

MODE:4001h = show as MS network  
(but no net drives  
and no net handles)

MODE: D001h = remote drives & MSNET

Other options will be documented in future releases.

LBL.COM can modify these bits in batch mode: LBL MODE:nn

## **QUIT**

If you must remove NET21 from memory, type: NET21 QUIT

## SHARE

Another important parameter is `SHARE:list`. This specifies a list of drives you are permitting other people to use. The default is all drives will be shared, ie, `SHARE:ABCDEFGHIJKLMNQRSTUUVWXYZ`. This is a global protection mechanism. If you do not share a drive then nobody but you can use that drive.

SHARE Example:      `SHARE:CDW`

This would let other nodes gain access to drives C, D, and W only.

## DEBUG

The last parameter is `DEBUG`. This is used to trace through network activity. There are several options, each option generating different messages.

```
DEBUG      = same as DEBUG:1 below
DEBUG:on   = no debug now, but load code
DEBUG:0    = no debug
DEBUG:1    = normal errors & boot echo
DEBUG:2    = background activity
DEBUG:4    = foreground activity
DEBUG:8    = routing traffic flow
DEBUG:16   = INT 21 reports
DEBUG:32   = INT 21 reports, wait
```

More than one of these options can be combined. For instance, if you want to see all normal error messages (#1), all background tracing (#2), and all traffic (#8), then enter `DEBUG:11` because  $1 + 2 + 8 = 11$ .

## VER

If you need to get the current version and the serial number, type: `NET21 VER`

## NOWIN

`NET21` is Windows aware. If you wish to disable Windows mode you can enter this parameter. Normally you should not do this.

## NODESQ

`NET21` is Desqview aware. If you wish to disable Desqview mode you can enter this parameter. Normally you should not do this.

Defaults = `NET21 BUFSIZE:1024,400,5 MODE:D001H`

# NET8.COM

**Purpose:** This coordinates all critical timing on the network.

**Syntax:** NET8 [ INTnn ] [ RES:nn ]

Many times when Little Big LAN is doing things it needs to know how long something takes or when to retry something. NET8 provides this service. It will almost never need any parameters.

NET8 must be installed for Little Big LAN to work. Place it in your AUTOEXEC.BAT file before any other network program. That is, it should be placed prior to COM\_LINK, PAR\_LINK, ARC\_LINK, ETH\_LINK, and NET21, or any other network program which loads from AUTOEXEC.BAT.

Programmers may be interested in some of the services this program provides. If interested, inquire direct.

[ INTnn ]	CPU interrupt # to use
[ RES:nn ]	approx ticks per second

Defaults: NET8 INT8 RES:18

# NETALIVE.COM

**Purpose:** Testing for remote node readiness

**Syntax:** NETALIVE NODE:x WAIT:y  
x = node# to test  
y = wait time in 1/18 ths of a second

Sometimes it is necessary to check that a remote node is booted up and ready for LAN requests. This utility is designed to check this.

NODE parameter specifies which node to test.  
This is a number, 1..250.

The WAIT parameter is rarely needed. It specifies how long to wait for the remote node to respond. The default is 9 ticks which is about a 1/2 second.

NETALIVE can be run from the DOS prompt or you can run it from a batch file.

For those writing batch files, NETALIVE will report exit codes depending on its operational status. The exit codes are:

0 = remote node is ready  
1 = link did not respond  
2 = network not installed

You can use a batch IF ERRORLEVEL statement to test the exit status. One use of this utility would be to turn off redirection prior to loading a program such as PCTOOLS which may try to scan all drives including those powered off:

```
echo off
rem drive H: is on node 100
netalive node:100
if errorlevel 2 goto err2
if errorlevel 1 goto err1
echo NODE IS ALIVE
goto end

:err1
echo LINK IS NOT ALIVE
LBL UNMAP DRIVE H:
goto end

:err2
echo NETWORK IS NOT INSTALLED
LBL UNMAP DRIVE H:
goto end

:end
PCTOOLS
```

# NETBIOS.COM

**Purpose:** Provides NETBIOS capability to Little Big LAN

**Syntax:** NETBIOS [LINK:a,b,c...] [MODE:x]

There is a lot of confusion as to what NETBIOS is and what it does for you. Many people, including many programmers who should know better, think NETBIOS and record locking are the same thing. They actually have nothing to do with one another. It is possible to use NETBIOS to provide record locking but this is an overly complex solution. Record locking is built into DOS via the SHARE command. There is no reason to look anywhere else for the capability.

There are a few programs which will require NETBIOS. But these are few and tend to be communication oriented packages such as Email or diagnostic utilities. Almost all multuser software uses DOS SHARE. Even many packages which list a NETBIOS network as a requirement never, in fact, use NETBIOS. So Little Big LAN does not automatically load NETBIOS. There is no reason to waste memory on something never used.

But if you need NETBIOS you must load this program.

Put it in AUTOEXEC.BAT after NET21.

## **LINK:a,b,c,...**

The "a,b,c,..." is a list of all nodes running NETBIOS. This is not needed if all nodes are running netbios. If not all nodes are running it, you should enter this parameter for performance reasons.

## **MODE:x**

Some internal operating modes can be controlled with this parameter. Currently only these are defined:

```
.... .... .... ...1 = ALIVE checking = on
.... .... .... ...0 = ALIVE checking = off
```

Default: MODE:1

# NETBSHAR.SYS

**Purpose:** Provides access to networked printers, plotters, etc.

**Syntax:** DEVICE=NETBSHAR.SYS addname targname [ON:nn]  
[BUFFER:nn] [OFF] [BIOS]

One of the most used features of a network is its ability to share remote printers. NETBSHAR allows you to tell Little Big LAN that a remote printer does exist and you want to be able to use it. If you have a printer hooked up locally to your computer then your computer already knows how to talk to it, therefore NETBSHAR is not needed to be able to print to local printers. NETBSHAR is useful only when you need to use a remote printer, i.e., one not attached to your computer.

Parameters "addname" and "targname" must always be entered. The following additional parameters may be entered AFTER the two device names. These are the default values you wish your computer to have after booting. Most can be changed on-the-fly by LBL.COM or other utilities.

## addname & targname

In order to gain access to a printer (or any character device) which physically resides on another computer you must inform the network of the remote device's existence. NETBSHAR.SYS is used to do that. Under DOS, all devices you use have a name. Printers are usually called LPT1, LPT2, and LPT3. If you have a serial printer it may be known as COM1. In order to use a remote networked device you must tell the system the device's name, and where it is. For each such device you must enter into your CONFIG.SYS file:

```
DEVICE=NETBSHAR.SYS addname targname
```

This tells the system you are going to be using a device called targname located on another machine. Targname is the "real" name of the device, i.e., this is the name you already know and use. For instance, if you have a printer known as LPT1, then LPT1 is the targname. The addname is the name by which you will be accessing that device from the machine which does not really have it. Addname is the new name you are going to call it. Addname can be any name. Targname and addname can be the same, or can be different. But don't worry. The network can keep things straight if you can. So the same printer may be known as LPT1 on one machine, but known as LPT2 from another machine.

For example, suppose you have a printer hooked up to LPT1 on each of two computers. The "real" name of each device is LPT1. When you print to LPT1 you would want the output to go directly to the locally attached printer. But with the network you would want to be able to print to either

printer from either computer. To do this add the following line in your CONFIG.SYS file on both computers:

```
DEVICE=\LBL\NETBSHAR.SYS LPT2 LPT1  
DEVICE=\LBL\NET00000.SYS
```

This tells Little Big LAN that you are going to use LPT1 hooked to the other computer, and you are going to call it LPT2 on this local computer. This line has no effect when printing to LPT1, so as before, printing to LPT1 will print to your local printer. However, printing to LPT2 will print to the remote printer (LPT1).

Note that NET00000.SYS must be loaded after NETBSHAR!

If you had entered:

```
DEVICE=NETBSHAR.SYS LPT1 LPT1
```

then printing to LPT1 could be routed away from your local LPT1 and would instead go to the remote's LPT1. This is okay but your local printer would not be available to you since LPT1 would be redirected to refer to the remote computer's printer and not yours. You might very well want LPT1 to refer to both the local and remote printers. In this case you could add the OFF parameter which has the effect of disabling this driver's redirection until it is later enabled by the LBL program.

## BIOS or INT17 or INT14

If entered, each of these parameters tell the NETBSHAR driver to route all appropriate low level BIOS requests to the network. This is the default mode of operation if addname is LPT1..LPT3, or COM1..COM4, or PRN.

This low level mode of operation is necessary to allow some programs to print to remote printers. For instance, LOTUS 123 bypasses the normal higher level DOS function calls and instead calls the BIOS printer service routines. Print Screen also does this.

These parameters force low level redirection if addname is close to the standard BIOS name. Only names of type L??n, or C??n will work, i.e., the first character must be "L" or "C" and the fourth character must be numeric.

## ON:nn

Use this to specify the node where the printer is. To find the proper device the network must be told to which node to send the request. ON:7 would specify that the targname device is found on the computer known as node #7

## OFF

Use this to keep the driver dormant. Later it can be awakened by the LBL program. This is useful when using LPT1 to refer to a local printer sometimes, and a remote printer at other times.

## BUFFER:nn

Use this to set the number of characters buffered before the network sends the request to the target node. This parameter adjusts effective speed of transfer. Maximum buffer size is 128 bytes. The default is 8. It will speed up printing when using a larger buffer size but it will also take more time away from the remote computer. These are tradeoffs you are best able to judge on your equipment. But it is usually best to use the default value.

### Examples:

```
DEVICE=NETBSHAR.SYS LPT1 LPT2 ON:10  
DEVICE=NETBSHAR.SYS LPT2 LPT2 ON:2 OFF BUFFER:16  
DEVICE=NETBSHAR.SYS JACK CON ON:1  
DEVICE=NETBSHAR.SYS JILL CON ON:2
```

### NOTE: Redirecting PrtScr

Since Print Screen always prints to printer 1, you can redirect PrtScr only when addname=LPT1.

## Why NETBSHAR?

This driver is an optimized version of NETSHARE.SYS. It is meant for faster printing to remote printers, so it has the ability to output only. As NETBSHAR collects characters it buffers up to 128 before sending a character packet to the remote device. If over 2 seconds elapse between characters then the current accumulation of characters is automatically flushed to the device. You will see this at the end of most print jobs as a slight delay before the last few characters are printed. This buffering significantly increases the throughput to printers over that of NETSHARE.SYS. But this time delayed buffering makes NETBSHAR.SYS a poor choice for passing character streams to some devices. One case is NCON/RCON. Using NETBSHAR to redirect to the NCON driver would give strange results.

NOTE: NETBSHAR.SYS must be loaded before NET00000.SYS

# NETCLOCK.COM

**Purpose:** Synchronizing the clock

**Syntax:** NETCLOCK NODE:x WAIT:y  
x = node# where to read the clock  
y = wait time in 1/18 ths of a second

Sometimes you may want to set your clock to the same time as a "master" clock on a particular node. This utility will read the clock on any node and then set the local computer's clock to the same date and time.

NETCLOCK can be run from the DOS prompt or you can run it from a batch file such as AUTOEXEC.BAT.

For those writing batch files, NETCLOCK will report exit codes depending on its operational status. The exit codes are:

- 0 = clock set ok
- 1 = link did not respond
- 2 = network not installed
- 3 = remote clock read was bad
- 4 = error writing local clock

You can use batch IF ERRORLEVEL to test the exit status.

NODE parameter specifies which node's clock to read.  
This is a number, 1..250.

The WAIT parameter is rarely needed. It specifies how long to wait for the remote node to respond. The default is 9 ticks which is about a 1/2 second.

# NETCOM.SYS

**Purpose:** To setup a COM port server

**Syntax:** DEVICE=NETCOM [COMx] [INTy] [IRQz]  
x = COM port number  
y = CPU interrupt number 11=irq3, 12=irq4  
z = IRQ#

If you have a COM port device such as a modem or FAX and you wish to permit other nodes to use that device then you must load NETCOM.SYS to setup the COM port for server operation.

NETCOM traps the INT14 BIOS and replaces it with an interrupt driven BIOS more suited for network use.

NETCOM merely provides a facility for use. Your communication package must be redirectable. Most current packages make direct use of the COM ports. They read and write directly to the hardware so it is impossible for such a package to use a remote modem. But more and more communication packages are starting to appear which allow INT14 redirection, or some other scheme. You must use one of these packages on any LAN if you want to use a modem that is not physically on the computer running the software.

This driver will be used in conjunction with the NETMSHAR driver which communicates over the LAN with NETCOM.

Place NETCOM in CONFIG.SYS.

**EXAMPLE:**

```
DEVICE=C:\LBL\UTILS\NETCOM.SYS COM2 INT11  
DEVICE=C:\LBL\UTILS\NET00000.SYS .....
```

Note that NETCOM.SYS must be before NET00000.SYS.

# NETMSHAR.SYS

**Purpose:** Provides ability to use remote COM ports.

**Syntax:** DEVICE=NETMSHAR.SYS localcom remotecom <ON:nn>

NETMSHAR.SYS is very similar to NETBSHAR.SYS but it is optimized for modems and other serial port devices. It should not be used to redirect printers.

Parameters "localcom" and "remotecom" must always be entered. Additional parameters may be entered AFTER these two device names.

This driver will probably be used in conjunction with the NETCOM driver which directly controls the COM port hardware.

## localcom & remotecom

"Localcom" is the COM name you wish to use to access the remote modem. The name should be COM1, COM2, COM3, or COM4. "Remotecom" is the true name of the COM port the modem is using.

"Remotecom" is the default target modem on startup, but can be changed on-the-fly by LBL.COM.

## ON:nn

Use this to specify the node where the target modem is. To find the proper device, the network must be told to which node to send the request. ON:7 would specify that the remotecom device is found on the computer known as node #7.

## EXAMPLE:

Lets say you want to use a COM2 modem on node# 7, but you already have a mouse on COM2 on your machine so you want to use the name COM3 to reference the modem. Add this to your CONFIG.SYS file:

```
DEVICE=NETMSHAR.SYS COM3 COM2 ON:7  
DEVICE=NET00000.SYS ....
```

This tells the system that when you use the name COM3 you really want to use a modem called COM2 located on node 7. This should be on node# 7:

```
DEVICE=NETCOM.SYS COM2  
DEVICE=NET00000.SYS ....
```

Note that NET00000.SYS must be located after all other devices!

# NETSHARE.SYS

**Purpose:** Provides ability to use remote character based i/o devices.

**Syntax:** DEVICE=NETSHARE.SYS addname targname <ON:nn>

NETSHARE.SYS is very similar to NETBSHAR.SYS but it is oriented more towards single character transfers without buffering. It should not be used to redirect printers. It will be used for various future utilities.

Parameters "addname" and "targname" must always be entered. Additional parameters may be entered AFTER the two device names. These are the default values you wish your computer to have after booting. Most can be changed on-the-fly by LBL.COM or other utilities.

Remember, use of NETBSHAR or NETSHARE is fairly interchangeable so do not use both to redirect the same device.

## addname & targname

For each new network device you want to add to your computer you must enter into your CONFIG.SYS file:

```
DEVICE=NETSHARE.SYS addname targname
```

This tells the system you are going to be using a device called targname located on another machine. Targname is the "real" name of the device, i.e., this is the name you already know and use.

The addname is the name by which you will be accessing that device from the machine which does not really have it. Addname is the new name you are going to call it. Addname can be any name. Targname and addname can be the same, or can be different. But don't worry. The network can keep things straight if you can.

For a more thorough description of the meaning of addname and targname read the NETBSHAR.SYS file section.

## ON:nn

Use this to specify the node where the target device is. To find the proper device the network must be told to which node to send the request. ON:7 would specify that the targname device is found on the computer known as node #7.

# NETSPOOL.SYS

**Purpose:** Permits multiple nodes to use one printer simultaneously

**Syntax:** DEVICE=NETSPOOL.SYS [LPTx] [COMx] [pathname]  
          [LOCK:nn] [WAIT:nn]  
          [SLICE:t1,t2,t2] [CTS] [DSR] [ONLINE:a,b]

NETSPOOL solves the printer sharing problem you would have if two or more computers try to print to the same printer at the same time. Without NETSPOOL you would get a garbled mess consisting of a random mixture of both print jobs. With NETSPOOL the print jobs are spooled to your hard drive and then printed one at a time. Install NETSPOOL via your CONFIG.SYS file on the computer which has the shared printer, as follows:

```
DEVICE=\LBL\NETSPOOL.SYS  
DEVICE=\LBL\NET0000.SYS ....
```

Then make a subdirectory called SPOOL in the home LBL subdirectory. This new subdirectory will keep temporary files used during the spool process. Optional parameters are defined as follows:

## LPTx

This specifies the printer. LPT1, LPT2, or LPT3 are valid.

## COMx

This specifies the printer. COM1, COM2, COM3 or COM4 are valid.

## Pathname

NETSPOOL creates Files named "spooljob.n" for a spooling node, and creates files named "printjob.n" for printing. The default path used is "C:\LBL\SPOOL". If you prefer a different path you can change it to anything else. Specify the whole path including drive letter.

## LOCK:nn

This specifies the time period in seconds NETSPOOL waits between print jobs. NETSPOOL uses a very simple but effective method to tell when your print job is finished and someone else can start printing. It merely watches printer activity and assumes if your computer has not sent anything to the printer lately that you are finished printing. That time period of inactivity is automatically set to 10 seconds unless you specify differently with this parameter. Laser printers may need 60 seconds.

## **WAIT:nn**

This specifies the time period NETSPOOL waits for the printer to indicate it is ready to receive another character. If this time is exceeded then NETSPOOL allows DOS some time before NETSPOOL retries to print the character. This parameter is an arbitrary loop count which is normally left to NETSPOOL to calculate on its own. However you can try other values. 25 is a good number for XTs, while 100 might be good for ATs.

## **CTS**

Use CTS for a ready indicator with some serial printers.

## **DSR**

Use DSR for a ready indicator with some serial printers.  
(default = either CTS or DSR)

## **SLICE:t1,t2,t3**

Specifies internal timing values, best left alone

- t1 = time per print timeslice in 1/18 seconds  
default = 3
- t2 = time between print timeslices  
default = 9 (9/18 seconds)
- t3 = time between not-ready retries  
default = 3 (3/18 seconds)

## **ONLINE:a,b**

If you install NETSPOOL and discover you can no longer print to your local printer then it may be due to an error in your hardware's reporting of when it is ready. Try putting the following parameter on the NETSPOOL line: **ONLINE:80h,80h**  
For those interested, the first parameter is the mask used on the status byte returned by the INT17 bios call, the second parameter is the value which indicates "ready" after the mask is applied.

## **DEBUG:n**

If it appears the spooler is not working, try adding: **DEBUG:9** to the NETSPOOL line. The spooler will print reports on internal activity. This may help locate the problem.

**NOTE:** NETSPOOL.SYS must be loaded before NET0000.SYS

# NETUNITS.SYS

**Purpose:** This specifies the number of remote drives.

**Syntax:** DEVICE=NETUNITS.SYS [@n]  
"n" is any number 1..24

NETUNITS.SYS tells DOS how many remote drive letters you plan on using. It must be placed into CONFIG.SYS. Without it the network may not be able to access any remote drives. That's because it is by way of NETUNITS that you tell DOS there are any remote drives at all. This driver supersedes DOS's LASTDRIVE parameter. LASTDRIVE may work with some DOS versions but not all. It is always safer to use NETUNITS.

The optional parameters are defined as follows:

## @n

Specify the remote drive count here. If this parameter is unused all drive letters from your current last drive letter through Z are assumed to reside on other computers. In most cases this is just fine.

**NOTE:** Most DOS versions allocate 112 bytes of RAM for each remote drive you add.

## DOS 6.x + DBLSPACE

If you get a message at boot time saying there are too many block devices you will have to specifically tell NETUNITS how many drive letters to reserve. Otherwise it may attempt to reserve more than DOS can handle. Find out the last drive letter DBLSPACE has reserved. Usually this is the last "host" drive letter. Then calculate the number of drive letters remaining through "Z:" That number is entered with the "@" parameter above.

## Disk compression programs

As a general rule, any disk compression device drivers should be loaded BEFORE NETUNITS.

# NODE.COM

**Purpose:** Use this to report network status and setup.

**Syntax:** NODE [?]

Sometimes it's nice to be able to verify the network is setup just as you think it is. This is especially true for diagnostic reasons. This program reports many things about the internal status of the network. This program is kept in the \LBL subdirectory, so get into that directory, then type NODE at the DOS prompt. Some of the things reported will be:

- 1) SYSREQ interrupt
- 2) The node # and name
- 3) A list of all installed link drivers & their setup
- 4) NET21 MODE options
- 5) NET21 DEBUG options
- 6) Local drive count
- 7) remote drive count
- 8) NET21 BUFSIZE
- 9) Current drive mapping

**Parameter: ?**

Entering a question mark as a parameter forces NODE to print a "help" screen. This is a common parameter on many LBL utilities.

# PARMUX.COM

**Purpose:** Allows several LPT ports to share the same IRQ line.

**Syntax:** PARMUX [INTx] [IRQx] [LPTa] [LPTb] [LPTc]

This utility allows Little Big LAN to share the same IRQ lines between two separate parallel ports. This makes it easier to set up multiple node configurations. Normally you have to specify a separate interrupt line for each LPT port you are using with PAR\_LINK. This means the network might need both IRQ5 and IRQ7, which are the normal LPT port interrupt lines. Sometimes it is not possible to jumper a parallel port for another irq. And when it is possible, it is likely you will need the irq for something else.

The multiplexer simplifies this problem.

PARMUX.COM shares one IRQ line with up to three networked LPT ports. To do this you must set your LPT port's hardware jumpers to the appropriate interrupt, then specify the same interrupt number on each PAR\_LINK command line. Lastly you must enter a new parameter on the PAR\_LINK line which tells each LPT port to place it's interrupt line in an inactive state. Then run PARMUX.COM once, specifying the same interrupt and all of the printer ports sharing that interrupt. For example, to use LPT1 and LPT2 for the network sharing IRQ7, enter the following in the AUTOEXEC.BAT file:

```
PAR_LINK LPT1 INT15 MUX
PAR_LINK LPT2 INT15 MUX
PARMUX INT15 LPT1 LPT2
```

Be sure that PARMUX is run after all the PAR\_LINK lines. Also, PARMUX should not be used if fewer than two LPT ports are sharing an interrupt. This multiplex program is designed to run with the Little Big LAN, not alien programs or device drivers.

The INSTALL program will automatically install PARMUX if two or more network linking LPT ports are sharing the same interrupt.

## INTx or IRQx

INTx specifies the CPU interrupt being used. Optionally, IRQx specifies the hardware interrupt being used. These parameters are interchangeable. Only use one of them. The INSTALL program uses INTx, not IRQx.

Default: INT15

# PAR\_LINK.COM

**Purpose:** Provides ability to link computers via parallel ports (printer ports).

**Syntax:** PAR\_LINK [LPTx] [INTx] [DEBUG] [MODx]  
[LINKTO:nn,nn,...]

Every computer that uses parallel ports to link to another needs to install this driver. It provides a standard interface to the card so that Little Big LAN can use the parallel port for communication. Place it in AUTOEXEC.BAT prior to NET21.

## LPTx or LPT:hh or LPTx=hh

You can specify the parallel port you are using in one of three ways. LPTx specifies via LPT port number, LPT:hh specifies via port hardware address, and LPTx=hh specifies via LPT port and the card's hardware address.

There can be up to four possible LPT ports in your computer, so the PAR\_LINK driver needs to know which one you are using to link to another computer. If you are using several then you must have several PAR\_LINK lines in AUTOEXEC.BAT.

You can specify an LPT port hardware address instead of an LPT port number. The address is the hex hardware address of your card. If you had a card using base address 300 then you can specify LPT:300. This is not the preferred way of specification so it should be avoided unless you fully understand it. It does force you to enter the MODx parameter.

You can also enter the LPT number and the card's address together. For example, suppose you want to use LPT4 to link to another machine but your BIOS does not recognize LPT4. Just enter:

```
PAR_LINK LPT4=2E0
```

This forces the LPT4 port address to be 2E0 hex.

## INTx or IRQx

Parallel ports are interrupt driven so you must configure your card to use one of the IRQ lines your card permits. Most cards allow selection of IRQs 5 or 7. Some also permit IRQs 2,3, and 4.

LPT1 is usually set by the manufacturer to IRQ7. LPT2 and LPT3 could be either IRQ5 or IRQ7.

**PAR\_LINK** can use any of these interrupts but you must know how your hardware is jumpered. Software is not capable of changing your board's jumpers. There are utilities in the **DIAGNOSE** directory which help you determine your cards current configuration.

Once you have determined your LPT port's interrupt setting, **PAR\_LINK** must be told which **IRQ** line you are using. You do this with either the **INT** or **IRQ** parameters. **INT** specifies the CPU interrupt which is the **IRQ** interrupt plus 8 for **IRQ** 0 thru 7, and is the **IRQ** number plus 104 for **IRQs** 8 thru 15. **IRQ5** is equivalent to **INT13**, **IRQ7** is equivalent to **INT15**. The **IRQ** parameter specifies the true interrupt line. These are interchangeable parameters. You need only use one of them. Why do we offer two parameters for the same purpose? Mainly because if you think in terms of software internals, as we do, you may prefer to use the CPU **INT#**, but if you prefer to think in terms of hardware you may prefer to use **IRQ#**.

## **MODx**

**MODx** is a parameter you should stay away from using. The system generates an internal module number and it is best just to leave it be. It specifies the Link Module Number used by Little Big LAN utilities to address the **PAR\_LINK** module directly. By default the parallel port driver module number is the LPT port number plus 30. So **LPT1** is module 31, **LPT2** is module 32, etc.

## **LINKTO:list**

**LINKTO:list** specifies other nodes reached directly via this driver. Since a parallel port is directly linked to only one computer the list should be only one node number. This parameter is one method you can use to tell **LBL** how to find a particular node. The list is either "ALL", or a list of node numbers. When this module detects a request to any of the node numbers in this list it will know it is supposed to send the request and does so. For example:

**LINKTO:2**

This indicates a direct connection exists to node 2. This is an optional method of supplying a connection list. The preferred method is via the **CONNLIST** file described elsewhere.

## **DEBUG**

This option turns on debugging capabilities for use with **WATCH**. It is helpful if there are problems communicating over the network.

Defaults = **LPT1 INT15 MOD31**

# PKT\_LINK.COM

**Purpose:** Provides ability to link computers via ethernet packet drivers

**Syntax:** PKT\_LINK [DEBUG] [INTx]  
[LINKTO:a,b,...] [MODx]

There are two basic methods of supporting ethernet cards. The best method is for us to write a driver to directly "talk" to the card. This method offers the best speed and the least memory usage. But it also requires us to write the driver. We will supply direct drivers for some cards, but there will always be cards we cannot support directly. However, there is another option. Most cards are shipped with software drivers providing a standard interface. Among those drivers you will usually find a packet driver. These drivers are also called the Clarkson Packet drivers, TCP/IP packet drivers, FTP drivers, or the Crynwr drivers. PKT\_LINK uses this driver.

This driver must be loaded prior to NET21 but after the packet driver itself. For more information on loading packet drivers read the diagnostic section of this manual referring to ethernet.

## INTx

This specifies the software interrupt the packet driver uses. It is not related to the IRQ in any way. Packet drivers can use interrupts 60H to 80H to communicate with upper level software. The default is normally 60H. Since INT needs the decimal equivalent, the default id INT96. Do not enter the IRQ or INT of the card.

## BUFSIZE:x

PKT\_LINK must allocate two internal memory buffer for reception and transmission of packets. These are set to 1024 bytes unless you tell PKT\_LINK you want a different size via this parameter. The buffers should be set to the same size as is specified on NET21.

## DEBUG

DEBUG forces extra code to load which allows the driver's activity to be monitored. It is required if using the WATCH diagnostic utility to detect ethernet status when debugging a non-functioning network. When running WATCH with PKT\_LINK, for best results specify the hardware irq of the ethernet card, as in: WATCH IRQ3

LINKTO and MOD are as described on ETH\_LINK.

# PORT.COM

**Purpose:** Manipulate port addresses, etc...

**Syntax:** PORT [COMx=hhh] [LPTx=hhh] [IRQ=on/off]  
x = port# to set  
hhh = hex address

PORT provides a means of modifying various port assignments. It can be used to modify the BIOS table, as in:

```
PORT COM3=3E8
PORT COM1=0
```

The first example would add the COM3 hardware address to the BIOS table which is necessary before some programs can use COM3. The second example blanks COM1 from the BIOS table which may be helpful to trick Windows 3.0 into leaving that port alone so that Little Big LAN can keep control.

Printer ports can be modified the same way:

```
PORT LPT4=3BC
PORT LPT1=0
```

You can also turn on or turn off the interrupt request line (IRQ) on a specific COM port. The PC/XT/AT hardware was not designed to permit interrupt lines to be shared. Since COM1 and COM3 share the same interrupt, and COM2 and COM4 share the other communications interrupt, it is possible for a COM port to get in the way of another COM port's use of its IRQ line. PORT will disable (tri-state) a COM port's IRQ line to permit another COM port to use it. One likely use is with PROCOMM. If you have the network running on COM3 and then load PROCOMM on COM1 you may find either PROCOMM does not work or the network does not work when you exit PROCOMM. Run PORT to disable the network's COM port IRQ line before loading PROCOMM, then disable PROCOMM's COM port when you exit it, remembering to re-enable the network's COM port.

```
PORT COM3 IRQ=OFF
PORT COM1 IRQ=ON
PORT COM3 IRQ=OFF COM1 IRQ=ON
```

# STIO

**Purpose:** Add Misc I/O Traps to LBL

**Syntax:** STIO

STIO.COM provides CONSOLE device read and write trapping to prevent operations such as TYPEing to the screen or COPYing from the CONSOLE from hogging DOS thus preventing LBL accesses.

There are a very few programs which temporarily disable the network when they are waiting for keyboard input. Also, there is one way you can do this in DOS by typing:

```
A>COPY CON CON
```

This disabling occurs because the programs are executing a DOS function call 3Fh with bx=0. This does an input of the Standard Input device (STI). Since DOS has been irrevocably entered by that function call and is using the wrong internal stack, no other DOS function call can be executed. This prevents other nodes from getting service.

The short utility STIO.COM will correct the problem. It traps this function call and looks for STI input. If there is a troublesome call then the utility emulates the call using a buffered keyboard input (Function 0Ah).

Just run STIO after the network loads:

```
C>\LBL\UTILS\STIO
```

It becomes resident, using about 1000 bytes. This fix will probably not be needed since very few programs require it.

Another problem this utility fixes is the TYPE problem. If you TYPE a big ASCII file or COPY it to the CON device then again DOS is in the wrong state and the network becomes inactive. This utility checks DOS function call 40h for this possibility.

This utility will also trap the keyboard Interrupt Service Routine to check for the Ctrl+Alt+Del key combination. If you have any opened files on a remote machine or if any other node has opened files on your machine, this utility will not allow you to reset the computer - otherwise you could destroy someone else's work.

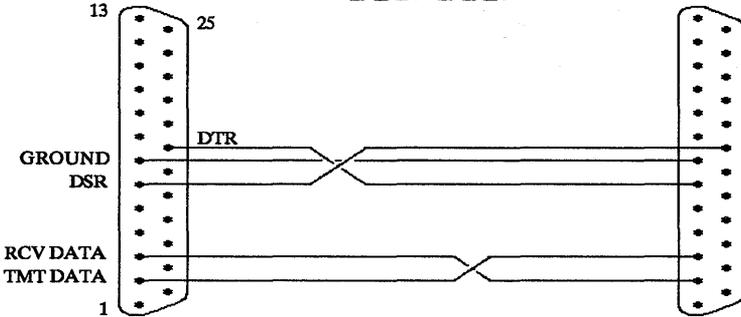
The Ctrl-S key combination and the Pause key are also trapped. An internal loop will stop screen scrolling as you would expect yet still allows remote nodes to access your machine.

# CABLES - serial

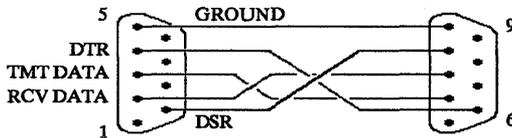
The following diagrams will help those who wish to construct custom serial cables for special installations. This may be helpful for users who wish to custom wire within an office or home.

Be aware that abnormally long cable lengths may cause problems. These problems can often be remedied by reducing the transfer speed of the network. (i.e.: to 57.6K or less) For best results, keep the cable lengths as short as possible. Also, try and route the cables away from AC power lines, extension cords, long printer cables, etc. Our experience has been that modular 6 conductor telephone cable runs reliably to lengths of up to 80 to 100 feet. Heavily shielded (and expensive) cable may not perform as well.

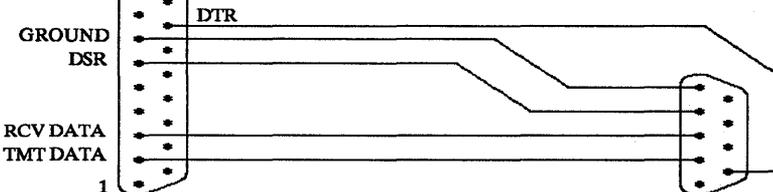
### DB25-DB25



### DB9-DB9



### DB25-DB9



## **NON-STANDARD CABLES**

Two special cable specific parameters may be entered on the COM\_LINK command line. DTR:x indicates that the DTR line is being specified to a particular bit in the 8250 Modem control register. The default is DTR:1, which uses the cable's DTR line as DTR. The other possibility on an unmodified serial card is DTR:2 which uses the RTS line as DTR. Similarly, DSR:2 indicates you are using the Modem status register's DSR as DSR. DSR:1 would use CTS, DSR:4 would use RING, and DSR:8 would use RLSD. These parameters are useful if you have a prewired cable which passes RTS/CTS instead of DTR/DSR, for instance. Also, if you have a bad DSR or DTR bit on your serial port then you can use a modified cable and bypass the bad bit. The above is only applicable when using a modified or specially wired cable.

## **READY MADE CABLES**

Cables and adapters are available for those customers who are having trouble either finding them at your local dealer or building their own. Each cable is made according to customer specifications.

Current pricing is as follows:

10 Foot modular cable .....\$ 15.00

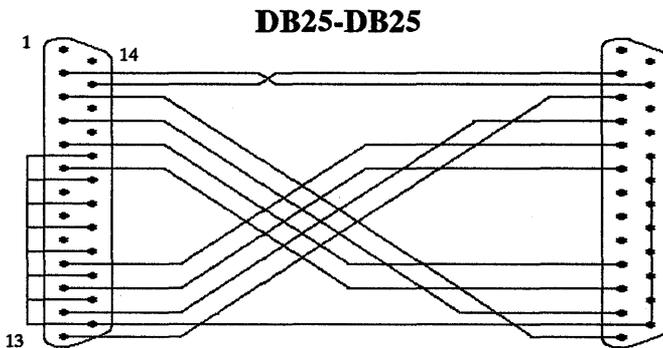
Specify gender and DB25 or DB9 for both ends.  
For longer lengths, add \$.25 per foot

Little Big LAN has been tested with cable lengths of up to 80 feet running at full speed. No problems were detected. Abnormally long cable lengths may not work in all environments.

# CABLES - parallel

The following diagrams will help those who wish to construct custom parallel cables. Imagine the DB25-DB25 diagram on this page as the back of your computer (outside). Your cable should "plug in" to this diagram and the wiring on your connector should match up with the wiring on this page.

The length of the cable must be kept to less than 15 feet. It is possible to use longer cables but be aware that parallel ports are not as electronically protected as are serial ports. It is possible to destroy parallel ports if your cable links two computers separated by a great distance. If you want to risk longer cables make sure your parallel port is of old TTL design, and on a separate board easily replaced. Do not risk long cables with laptops!

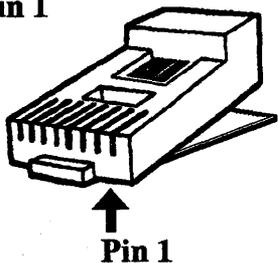
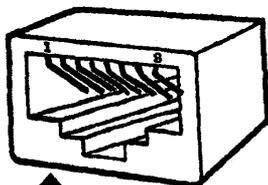
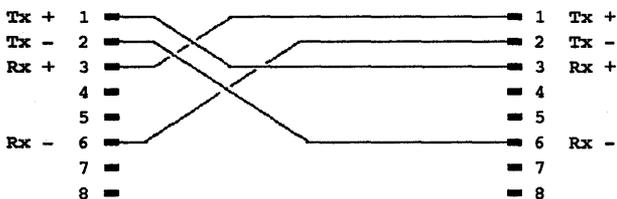


	FROM DB25P		TO DB25P	
	PIN #		PIN #	
Strobe	1	NO CONNECTION	1	
Data Bit 0	2	-----wire-----	15	
Data Bit 1	3	-----wire-----	13	
Data Bit 2	4	-----wire-----	12	
Data Bit 3	5	-----wire-----	10	
Data Bit 4	6	-----wire-----	11	
	7	NO CONNECTION	7	
	8	NO CONNECTION	8	
	9	NO CONNECTION	9	
Acknowledge	10	-----wire-----	5	
Busy	11	-----wire-----	6	
Out of Paper	12	-----wire-----	4	
Select	13	-----wire-----	3	
	14	NO CONNECTION	14	
Error	15	-----wire-----	2	
	16	NO CONNECTION	16	
	17	NO CONNECTION	17	
Ground	18	---1 wire---	18	Ground
Ground	19	---	19	Ground
Ground	20	---	20	Ground
Ground	21	---	21	Ground
Ground	22	---	22	Ground
Ground	23	---	23	Ground
Ground	24	---	24	Ground
Ground	25	---	25	Ground

# CABLES - ethernet

If you are using ethernet twisted pair cards (10baseT) and want to link two computers without buying a concentrator, you must wire a cable as documented here. Do not use this cable with a concentrator! This diagram should be used only in a two computer setup with the computers wired directly together through this cable.

Computer 1			Computer 2	
	PIN #			PIN #
Tx +	1	-----wire-----	3	Rx +
Tx -	2	-----wire-----	6	Rx -
Rx +	3	-----wire-----	1	Tx +
	4	NO CONNECTION		
	5	NO CONNECTION		
Rx -	6	-----wire-----	2	Tx -
	7	NO CONNECTION		
	8	NO CONNECTION		



# Technical notes

## File Sharing

If you are using a multiuser database or other multiuser software you need to install the DOS SHARE command to permit record locking. Read the DOS manual for more information on this. The fact that you have a network does not necessarily mean you are going to share data files among several nodes, therefore there may not be a reason for you to install SHARE.

If you have a single user application you should not try to modify the same data, or add to the same file from two nodes at the same time. Only one node should be in the application unless others are only reading data. There is no way for any network to make a single user program into a multiuser program.

Be sure you have enough files specified in CONFIG.SYS.

Also, if you need to open more than 27 remote files you need to use the HANDLES parameter on NET21. Some accounting packages and database systems will require this. See NET21.COM.

## SUBST

You can use the DOS SUBST command to limit remote users to a few subdirectories on a hard drive. Let's say you don't want remote nodes to use your D: drive, but one path on the drive would be okay for them to use. Let's call that path D:\PUBLIC. Now with SUBST you can reference that path as a drive letter then share that new drive letter, but not D:, on the NET21 line using the SHARE:list parameter. So put into AUTOEXEC.BAT:

```
SUBST P: D:\PUBLIC
NET21 SHARE:ABCP
```

The SUBST line must be before the NET21 line. Now remote users can only access your A:, B:, C:, and D:\PUBLIC (which will be your drive P:).

# Trouble Shooting

Make sure the diagnostics tests pass before wasting any time on checking for software installation problems. If your hardware fails the diagnostic tests then there is no way the network will work.

If the hardware fails try new or different combinations of hardware. If possible, try different combinations of computers and limit your testing to pairs only. So, find which computer is causing the trouble, then it is much easier to isolate the problem. Its always best to get the network working under any circumstance and then work backwards from there until you find the troublesome component. This may require pulling cards. Remove all cards which are not absolutely necessary for testing. **DO NOT ATTEMPT THIS UNLESS YOU ARE ABSOLUTELY SURE OF WHAT YOU ARE DOING!** In some cases this may void the warranty on your computer. If you are at all in doubt, don't !

When you get the network running then you can start putting things back together one at a time, testing after each addition. This way you will probably find the culprit.

If the hardware tests pass but the software fails try removing all TRSs, device drivers, menu systems, and DOS shells. Run only DOS and the network. Again, it's best to get the network working any way possible and then start restoring your software setup to its original state, testing all the while until the problem shows up.

The main sources of problems are likely to be:

- 1) No terminators on ethernet
- 2) IRQ conflicts between ethernet card and other cards  
(like Sound board, COM port, FAX board, Scanner)
- 1) Cable!
- 2) Serial cables too long or (rarely) ports not capable of full speed.
- 4) Incompatible software.  
Test with only what is needed to boot.
- 5) The network is not installed or partially installed.  
(due to menu running before LAN, or exiting to another batch)
- 6) Human error.  
Double and triple check yourself.

# GLOSSARY / Definitions

## Application program

Any program you normally run under DOS. For example, a word processor, spreadsheet, or database program.

## ARCNET

Attached Resource Computer system Network; originally developed by Datapoint Computer, based on CSMA/CD packet communication, with a data rate of 2.5Mbit/sec.

## ASCII

American Standard Code for Information Interchange; an ASCII file is a simple text file which can be displayed with the DOS TYPE command.

## Asynchronous

Something that can occur at any time, without warning - like a telephone ringing at dinner time.

## Baseband

A transmission technique in which only one transmission signal at a time can be present - like a one lane road.

## Baud rate

A measurement of transmission speed. The baud rate is how many times per second that switching can occur.

## BIOS

Basic Input/Output System; The code built into system ROM the provides the lowest level of functionality in a computer system.

## Block device

A device which is file and directory oriented and lumps information together in large blocks. A disk drive is a block device

## Broadband

A transmission technique in which many signals can coexist on one wire -like a four lane highway, or the airwaves which allow many stations to coexist. This is generally more complex and expensive than baseband techniques.

## Bus

As regarding a network topology, a bus is a single physical communication channel which is simultaneously connected to all nodes so that any node can speak directly to any other node but only one node can speak at any given time (baseband transmission). Inside a computer a bus is a collection of parallel lines that pass data, address, and control information between devices.

## Character device

A device which is single character oriented like a printer or keyboard. It is not file oriented - one file name refers to the whole device.

## Collision

A collision occurs in baseband transmission when any more than one node attempts to transmit on a single medium at one time. This causes disruption of the transmission signals.

## **Communication link**

An electrical and logical connection between two devices. In the case of a network, a communication link is the path through which the source and destination nodes communicate.

## **CSMA/CD**

Carrier Sense Multiple Access with Collision Detection. In a network system using this scheme, when a collision is detected transmitting stations stop transmitting and wait for a predetermined interval before trying again.

## **Debug**

When computer people are trying to locate problems (bugs!)

## **Default**

That which is selected for you if you are not specific in your request for service. e.g. default device, parameter, path, file, operation etc...

## **Device Driver**

A program that interfaces the unique characteristics of a computer hardware device to the requirements of the operating system.

## **DOS**

Disk Operating System; commonly referring to Microsoft or IBM PC DOS

## **Email**

A program which allows many users to send mail and leave messages to one another.

## **EtherNet**

A network protocol developed by the Xerox Corporation, utilizing a bus topology, baseband coax medium, with 10Mbit/sec data rate.

## **Interrupt**

A hardware signal or microprocessor instruction that halts processing momentarily so that a specific operation may take place. When the operation is finished, processing resumes. Interrupts may be hardware or software generated. Hardware interrupts are asynchronous events usually serviced by a fairly brief sequence of instructions to be performed without hogging much processor time. Think of a hardware interrupt as a child interrupting you to ask for something while you are in a heavy conversation with an adult. You quickly service the child, then pick up your adult conversation where you left off. Software interrupts are often used as system function calls or subroutines and the interrupt service routine may be quite extensive.

## **IRQ**

Interrupt ReQuest;

## **LAN**

Local Area Network; group of PCs connected together for common access.

## **LBL**

Little Big LAN

## **Local**

A resource attached to the computer you are sitting at is local

## **Memory resident**

A program or data that, once loaded into memory, stays there ready for action until the computer is powered down, reset, or deliberately released from memory.

## **MODEM**

MOdulator/DEModulator; used for converting digital data to analog signals for transmission over telephone lines .

## **NETBIOS**

NETwork Basic Input/Output System. This is a standard protocol many networks use to communicate among nodes. A computer network "language."

## **Network interface card**

Sometimes abbreviated "NIC". A printed circuit board that plugs into a bus slot in a computer and permits direct connection to a network cable.

## **Node**

One individual computer on a network.

## **Node Number**

The address by which any other node knows your computer; Each node number must be unique.

## **Null modem**

A cable used to directly connect two terminal devices to each other without going through the usual intervening modems. As a minimum, the cable usually swaps the transmit data (TxD) and receive data (RxD) lines and may swap other signals as well, in order to make each terminal device to appear as a modem to the other.

## **Operating System**

Software that manages all resources of a computer and performs various interfacing chores between software and hardware.

## **Packet**

A bunch of data transmitted as a unit, usually with addresses and control codes for identification when transmitted on a computer network.

## **Parallel**

Concurrently; more than one at a time; a parallel interface on a printer typically receives 8 bits at a time.

## **Peripheral Device**

A hardware device such as a video monitor, disk drive, printer, or modem -used in conjunction with a computer and under its control.

## **Physical layer**

The physical layer is the physical connection between the nodes of a network system. It is composed of the cables, connectors, electrical specifications and modulation techniques as well as the bandwidth (speed) and network topology.

## **Polling**

A system of regularly checking the status of attached devices. This method of looking for work to perform is orderly and predictable however, if there are many devices and infrequent transactions then the interrupt approach may be better suited for control.

## **Pop-up**

A memory resident program that opens a window on to your console device usually by means of a hot-key sequence and allows execution of some routine task or perhaps provides some information like system status etc..

## **Protocol**

A set of defined parameters for establishing and controlling communications.

## **Redirector**

The part of the network software which determines a request refers to a remote resource and sends the request to the remote machine.

## **Remote**

A resource attached someone else's computer is a remote resource.

## **Repeater**

A device used to amplify and pass along a signal.

## **Resource**

A computer's resources include software, disk drives, or disk space, CPU time, Clock, keyboard, monitor, programs, printers, FAX boards, mice, etc. A network is designed to allow you to use not only your computer's resources, but also some other computer's resources.

## **ROM**

Read Only Memory; special memory chips with a program permanently burned into them.

## **Serial**

One after another; one at a time; typically serial data is transmitted one bit at a time.

## **Server**

The part of the network software which "services" incoming DOS requests.

## **Star**

A network topology in which two or more nodes are connected to a common hub node such that no transactions may take place without involving the hub node. The hub will be involved directly in communicating, or indirectly, by having messages passed through it to reach another node.

## **Terminator**

An impedance matching device that prevents signal reflections from traveling backwards along a cable from the destination back toward the source.

## **Topology**

With respect to a computer network, designates the mode of connectivity of all or part of a particular network. Some networks consist of composites of different topologies e.g. star, bus etc... Topologies fall into two basic categories: centralized and decentralized. In a centralized topology such as a star network, a central computer controls all access to the network. In a decentralized topology such as a bus network, each node can access the network independently and communicate directly with the other nodes.

## **TSR**

Terminate but Stay Resident; a program which stays in memory after exiting to DOS.

## **Twisted pair**

A low bandwidth connecting cable used in telephone systems which has a characteristic impedance of 600 - 900 ohms in the voice frequency range. The cable includes two wires which are twisted together to minimize interference from neighboring wires. Sometimes satin cord is substituted for actual twisted pair cable and while it is not as good at crosstalk rejection, still it performs adequately in most cases for "twisted-pair" network cabling.

