

AdeptNet User's Guide

This is a printable version of the AdeptNet User's Guide online documentation. A Table of Contents is provided so that you can locate the desired topics. Because this document is designed for online viewing, cross-references will not have page numbers. For easier navigation use the online version of this document, which can be accessed from the Adept Document Library.

Table of Contents

AdeptNet User's Guide.....	1
Introduction to AdeptNet.....	3
1.1 What Is the AdeptNet Product?.....	3
1.2 Using This Manual?.....	3
1.3 Documentation Conventions.....	3
Typographic Conventions.....	4
Keyboard Conventions.....	4
Abbreviations.....	5
List of Figures.....	7
System Overview.....	9
2.1 What Is Ethernet?.....	9
2.2 Network Hardware.....	9
Adept SmartController.....	10
Adept MV Controller.....	10
PC/UNIX Server or Host Machine.....	11
2.3 Network Software.....	11
The AdeptNet Software Model.....	11
AdeptNet Software.....	12
Software Required for Non-Adept Equipment.....	13
Installing AdeptNet.....	15
3.1 Installing AdeptNet.....	15
Installing AdeptNet in an 040-Based Adept MV Controller.....	15
Instructions for MV-5, MV-8, and MV-19 Controllers.....	15
Instructions for MV-10 Controller.....	17
3.2 Connecting to a Network.....	18
3.3 Software Installation.....	19
Installing V+ as an Upgrade.....	19
Installing AdeptNet Software Licenses.....	19
Network Configuration.....	21
4.1 IP Network Address.....	21
4.2 Adept Controller Configuration Using CONFIG_C.....	21
Explanation of the Default Configuration.....	22
Configuration Limits.....	23
4.3 Network Functionality Check.....	24
4.4 Network Configuration of Non-Adept Equipment.....	24
PC-Based Server.....	24
AdeptNFS:Network File System.....	25
5.1 Introduction to AdeptNFS.....	25
5.2 System Requirements.....	25
5.3 Mounting a Remote Disk.....	25
Default Mount at System Initialization.....	25
Mounting by Using the FSET Monitor Command.....	26
Mounting by Using the FSET Program Instruction.....	26

Table of Contents

- AdeptNFS: Network File System**
- 5.4 Referring to Remote Drives.....26
- 5.5 V+ Monitor Command Support.....27
 - Examples.....27
- 5.6 V+ Program Instruction Support.....28
 - Example Code: Writing to and Reading from a Remote File.....28

- AdeptFTP and AdeptTFTP: File Transfer Protocol.....31**
- 6.1 Introduction to AdeptFTP and Adept TFTP.....31
- 6.2 System Requirements.....31
- 6.3 How AdeptFTP Works.....32
 - Related Topics.....32
- 6.4 Using AdeptFTP Server.....32
 - Loading and Executing the Server Program.....33
 - Stopping the Server Program.....34
- 6.5 Using AdeptFTP Client.....34
 - Loading and Executing the Client Program.....34
 - Help Command.....34
 - Connection Commands.....34
 - Directory and File Commands.....35
- 6.6 AdeptFTP and File Types.....37
- 6.7 Using AdeptTFTP.....37
 - Parameters.....37
 - Examples:.....38

- Writing Custom Protocols Using AdeptTCP/IP.....39**
- 7.1 Introduction.....39
- 7.2 System Requirements.....39
- 7.3 Ports and Sockets.....39
- 7.4 How a TCP Connection Is Established.....40
 - Server Side.....40
 - Client Side.....40
- 7.5 Example V+ Code.....40
 - AdeptTCP/IP Echo Server.....40
 - AdeptTCP/IP Client.....42

- AdeptUDP: User Datagram Protocol.....45**
- 8.1 AdeptUDP Introduction.....45
 - Sending and Receiving Data with UDP.....45
 - Related Topics.....46

- Using AdeptNet with V+.....47**
- V+ Keywords Related to AdeptNet.....47
 -48
 - System Messages Related to AdeptNet Software48

AdeptNet User's Guide

This is a printable version of the AdeptNet User's Guide online documentation. A Table of Contents is provided so that you can locate the desired topics. Because this document is designed for online viewing, cross-references will not have page numbers. For easier navigation use the online version of this document, which can be accessed from the Adept Document Library.

Introduction to AdeptNet

[What is AdeptNet](#)

[Using this Manual](#)

[Documentation Conventions](#)

1.1 What Is the AdeptNet Product?

AdeptNet is Adept's Ethernet networking solution for the Adept SmartController and MV controller. AdeptNet allows multiple Adept controllers to operate as nodes on a Local Area Network (LAN) that may include other non-Adept devices such as PCs or UNIX-based workstations. AdeptNet software allows for peer-to-peer communications, file transfer, and sharing of a (non-Adept) file server.

1.2 Using This Manual?

The following sections of this user guide provide information on installing, configuring, and using the AdeptNet product:

1. The [System Overview](#) section outlines the hardware and software required for the AdeptNet option and introduces basic network terms for those not familiar with the technology.
2. [Installing AdeptNet](#) gives instructions on installing the AdeptNet hardware in your Adept MV controller.
3. [Network Configuration](#) gives all the necessary information required to configure the Adept MV controller as a node on a Local Area Network (LAN).
4. [AdeptNFS: Network File System](#) tells you how to use the network services provided by AdeptNFS, Adept's optional Network File System software.
5. [AdeptNet FTP: File Transfer Protocol](#) describes how to use the file transfer facilities provided by AdeptFTP, Adept's optional File Transfer Protocol software.
6. [Writing Custom Protocols Using Adept TCP/IP](#) gives detailed information on how to use Adept's TCP/IP and sockets driver to write your own custom protocols in V+.
7. [Addendum to the V+ Reference Guides](#) details the additions and changes to the V+ Language and Operating System program instructions, functions, and monitor commands that are specific to the AdeptNet option. More detailed information on these topics is available in the [V+ Operating System User's Guide](#) and the [V+ Language Reference Guide](#).

1.3 Documentation Conventions

This section discusses:

- Typographic conventions
- Keyboard conventions
- Abbreviations

Typographic Conventions

The following typographic conventions are used throughout this manual:

This	Represents
ALL CAPITALS	file names, directory names, commands, keywords, and attributes; also acronyms.
regular monospace	Monitor displays, code examples, non-placeholder terms in formal syntax definitions, and case-sensitive words required for a UNIX-like setting associated with the AdeptNet product.
italic monospace	Placeholders for information that you provide in formal syntax definitions. You must replace such a placeholder written in bold weight but need not replace an optional one, which is written in regular weight.
bold	<p>In a typing or entering instruction, anything that you type exactly as it appears. For example, if you are asked to type <code>execute 1 a.diskcopy</code>, you would type all the bold characters exactly as they are printed. What you type is shown in lowercase letters unless it must be typed in uppercase letters to work properly. You may always substitute a currently valid shortcut form when typing a V+ command. In order for the V+ system to process your typing, you must conclude your entry by pressing the enter or return key.</p> <p>Bold type is used for subroutine names, variable names, and program names, such as <code>a.diskcopy</code>. Bold type also is used for window items that you choose and window items that do not have initial capital letters in all principal words.</p>
italics	Placeholders that you must provide in typed input. This font also indicates new terms and other emphasized words.
Initial Capitals	The name of an object such as a window, screen, menu, button, dialog box, or dialog box component. Examples are the Display menu and the Task Profiler window.
small capitals	A physical key or button that you must press, such as the enter key.

Keyboard Conventions

Key combinations appear in the following format:

--	--

Introduction to AdeptNet

Notation	Meaning
key1+key2	A plus sign (+) between keys means that you must press the keys at the same time. For example, "Press ctrl+z" means that you press ctrl and hold it down while you press z.

Abbreviations

The following abbreviations may appear in this manual:

Abbreviation	Meaning
MB	Megabyte (1,048,576 8-bit bytes)
Mb	Megabit (1,048,576 bits)

Introduction to AdeptNet

List of Figures

The AdeptNet User's Guide contains the following figures:

Figure 2-1 AdeptNet Hardware Installed in an Adept MV-8 Controller

Figure 2-2 The AdeptNet Software Model

Figure 3-1 Industry Pack and Ribbon Cable on 040 Module

Figure 3-2 Ribbon Cable on Transition Module

Figure 3-3 AdeptNet Hardware Installed in an Adept MV-8 or MV-19 Controller

Figure 3-4 AdeptNet Hardware Installed in an Adept MV-10 Controller

Figure 3-5 AdeptNet 10BASE-T Installation Requires a Hub

List of Figures

System Overview

This section outlines the hardware and software required for the AdeptNet option and introduces basic network terms for those not familiar with the technology.

What is Ethernet?

Network Hardware

Network Software

2.1 What Is Ethernet?

Ethernet provides a method for high-bandwidth communication between devices on a Local Area Network (LAN). Ethernet had its historical beginnings as a joint development between Digital Equipment Corporation and Xerox Corporation. It is sometimes referred to by its Institute of Electrical and Electronics Engineers (IEEE) standard number: IEEE 802.3. The term Ethernet designates both physical transmission media and the method of transmission along the media.

There are three commonly used physical media for Ethernet:

- ◆ 10BASE-5, or ThickNet. This is the coaxial cable standard used by the original Ethernet standard.
- ◆ 10BASE-2, ThinNet or CheapNet. As the name implies, this uses a thinner, cheaper coaxial cable.
- ◆ 10BASE-T, or twisted pair. It uses unshielded twisted-pair wire.

The physical medium used also defines the topology of the network. ThickNet has a bus topology. Network devices are attached via transceivers mounted on the cable itself. ThinNet also has a bus topology but devices attach directly to the bus via BNC T-connectors. Twisted pair has a star topology, where only one network device can be attached per cable. A concentrator (or hub) is required to make the final connection to the network.

The physical media also differ in their capacity for number of nodes and transmission distance. In all cases, the Ethernet specification dictates transfer of data at up to 10 megabits per second (Mbps).

Ethernet uses what is termed a Carrier Sense, Multiple Access, Collision Detect (CSMA/CD) transmission algorithm. As the name suggests, CSMA/CD allows multiple devices, or nodes, on the network to attempt to access the network at any time. If a collision is detected on attempting to transmit, the node waits for a short period of time and then attempts to re-transmit the message.

There are many suitable texts on the subject of networking and Ethernet. For more detailed information on the subject you may wish to consider referencing such a text.

2.2 Network Hardware

Adept SmartController

See *SmartController Connectors and Indicators* in the *Adept SmartController User's Guide* for details.

Adept MV Controller

For Systems Using an 040 System Processor

The AdeptNet hardware includes a daughterboard for the 4Mb or 8Mb 68040 system processor board. This daughterboard is termed an Industry Pack and plugs directly into connectors on the 040 board.

In addition to the Ethernet Industry Pack, a Transition Module is required, which converts the signal output from the Industry Pack into a standard Ethernet connection, in Adept's case providing direct connectivity for 10BASE-T twisted pair. Note that the slot occupied by the Transition Module is not a backplane slot.

The Industry Pack Ethernet daughterboard and Transition Module are linked by a ribbon cable connection. [AdeptNet Hardware Installed in an Adept MV-8 Controller](#) shows the completed assembly in an MV-8 controller. Note that in an MV-10 controller, the Transition Module is installed on the right-hand side of the 040 module, and the ribbon cable remains inside the controller. For more details, see [Installing AdeptNet](#).

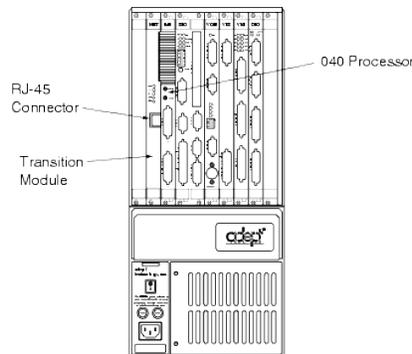


Figure 2-1. AdeptNet Hardware Installed in an Adept MV-8 Controller

If the AdeptNet option was specified at the time of purchase, Adept has installed the option for you. If you purchased AdeptNet as an upgrade option for an existing Adept MV controller, you need to install AdeptNet yourself or have a qualified Adept Field Service Engineer do so for you. For details, see [Installing AdeptNet](#).

For Systems Using an 030 System Processor

AdeptNet is not currently available as an option for systems using the 030 system processor. If you have an existing 030-based installation and require the AdeptNet option, please contact your Adept Sales Engineer to discuss upgrade options.

PC/UNIX Server or Host Machine

AdeptNet provides a method to store and retrieve data or programs from a non-Adept system server or host, such as a PC or UNIX workstation.

UNIX was designed with network connectivity in mind, and your UNIX workstation should already have the necessary hardware required. Ethernet hardware in the form of a Network Interface Card (NIC) is commonly available for the PC platform. Most NIC cards offer a choice of connectivity with a 15 pin AUI connector (for 10BASE-5 or Thicknet), an RJ45 telephone-style socket for 10BASE-T, and/or a BNC connector for Thinnet. For direct media compatibility with the AdeptNet option, you need to purchase a NIC fitted for RJ45 twisted pair (10BASE-T).

2.3 Network Software

The AdeptNet Software Model

The AdeptNet Software Model describes Adept's network software architecture.

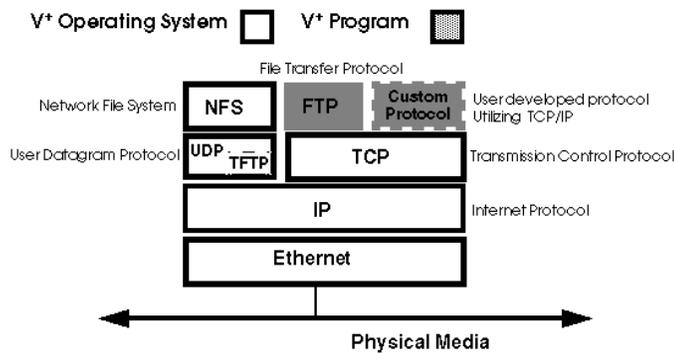


Figure 2-2. The AdeptNet Software Model

In this layered model, each software layer uses the services provided by the lower layers. The lowest layer shown is the Ethernet hardware layer. This layer defines the physical connectivity of the network and the low-level software required to transmit and receive messages over the network.¹

Above this layer resides a standard protocol called the Internet Protocol (IP) layer.² Every network node has what is termed a Network IP address. The IP address uniquely identifies the network node either in the LAN (Local Area Network) or WAN (Wide Area Network). In the case of transmission, the IP layer establishes connection to the desired destination node. In the case of a received message the IP layer recognizes the IP address of the node, and accepts or rejects the message accordingly.

Above the IP layer reside two different protocols: UDP and TCP.³ Functionally these layers achieve the same result: they allow a consistent interface to applications needing access to the network. UDP (User Datagram Protocol) provides what is termed an unreliable service; that is, message delivery is not guaranteed. This implies greater simplicity and improved speed of

System Overview

operation since the receiving node in the network need not send an acknowledgment that the message has been received. For this reason UDP is typically used in applications not requiring absolute reliability, and for broadcasting to several nodes. An application using UDP can of course perform its own error checking and status monitoring. For details, see the [AdeptUDP](#) documentation.

TCP (Transmission Control Protocol) provides reliable point-to-point service. TCP was designed for applications involving transmission of information across Wide Area Networks (for example, Internet), where reliable communication is essential. AdeptNet uses UDP/IP or TCP/IP depending on what application layer protocol is being used.

At the top or Application layer, Adept offers three packages for Network File System and File Transfer (Protocol) functionality. For details, see the [AdeptNFS](#), [AdeptFTP](#), or [AdeptTFTP](#) documentation. Note that NFS uses the UDP protocol and FTP uses the TCP protocol. Adept also offers direct access to TCP with AdeptTCP/IP.

The upper-layer protocols in the software model use the functionality provided by the lower layers. In the case of transmission, each layer attaches its own header and trailer information to a message being sent. In the case of receiving a message, each layer strips off the header and trailer information it needs and passes the remaining message to the next layer above.

AdeptNet Software

AdeptNet is available as a software option with V+ version 11.2 or later. There are three software licenses associated with the AdeptNet product: the AdeptNFS license (optional), the AdeptFTP license (optional), and the Adept TCP/IP license (included as standard with AdeptNet).

Network File System – AdeptNFS

Adept's Network File System software allows an Adept controller to appear as a client on a LAN and share the resources of a non-Adept server with other devices in the network. AdeptNFS does not allow an Adept controller to be used as an NFS server. A server device typically is a PC or UNIX workstation.

AdeptNFS allows the disk resources of the server to be accessed by the Adept client much as if it were a local drive inside the Adept controller. This allows the Adept controller to do the following:

- ◆ Load or store files from/to the server.
- ◆ Create or delete files and directories on the server.
- ◆ Attach, read, and write files on the server under program control.

File Transfer Protocols – AdeptFTP and AdeptTFTP

AdeptFTP allows the peer-to-peer transfer of complete files from one network node to another network node. FTP transfers can occur across a LAN or a WAN.

The AdeptTFTP device provides a way to read and write files from a host machine without requiring any additional configuration steps, such as those required for the NFS device. (However, the NFS device provides greater functionality.) The TFTP device has only one channel which supports READ and WRITE access. Because the device has one channel, there can be only one active transfer at a time.

Direct TCP/IP Access – AdeptTCP/IP

AdeptTCP/IP allows direct access to the TCP Transport Layer protocol. This allows developers to create their own application-specific protocols for high-bandwidth peer-to-peer communications between Adept controllers or other network devices (PLCs, PCs, and workstations).

Software Required for Non-Adept Equipment

For UNIX Stations Such As SUN Workstations

UNIX has functionality built into the operating system to support both NFS Server and FTP transfer. Refer to your UNIX manual for more details.

For PCs

There are many networking software products available for the PC platform. Some of these effectively replace the PC's MS-DOS operating system (for example, Novell's Netware). Others run as application packages under MS Windows (for example, NetManage's Chameleon). Any networking software that provides a standard (UNIX-style) TCP/IP implementation should work with AdeptNet. A good rule of thumb is to consider the Adept MV controller as a UNIX emulator. If the package you intend to purchase offers connectivity to UNIX, then it should be compatible with AdeptNet.

For Other Network Devices

Some other industrial control products utilize Ethernet and the TCP/IP protocol for network communications. Any third-party device that uses compatible physical media and the standard TCP/IP protocol should be compatible with AdeptNet. For peer-to-peer communication between an Adept controller and another TCP/IP device, you need to write an application layer protocol in V+. The purpose of this program is to emulate the protocol required by the third-party equipment. For more information on network compatibility, contact Adept's Application Engineering group.

1For people familiar with the OSI seven-layer model, this layer is equivalent to the Physical and Data Link layers.

2Referred to as the Network layer in the OSI model.

3UDP and TCP are Transport layers in the OSI model.

System Overview

Installing AdeptNet

The Installation documentation contains instructions on how to install AdeptNet hardware and software.

3.1 Installing AdeptNet in an 040-Based Adept MV Controller

Instructions for MV-5, MV-8, and MV-19 Controllers

Instructions for MV-10 Controller

3.2 Connecting to a Network

3.3 Software Installation

Installing V+ as an Upgrade

Installing AdeptNet Software Licenses

3.1 Installing AdeptNet

If you have an Adept SmartController, an AWC-based MV Controller, or specified the AdeptNet option during controller purchase, the AdeptNet hardware has been installed at the factory. In this case, proceed to

Installing AdeptNet in an 040-Based Adept MV Controller

If you specified the AdeptNet option during controller purchase, Adept has installed AdeptNet for you. If this is the case, you can proceed to the next section.

If you have purchased AdeptNet as an upgrade, you have the option of having an Adept Field Service Engineer perform the upgrade for you or you can perform the upgrade procedure yourself.

 **CAUTION:** This procedure should be attempted only by personnel familiar with handling and servicing electronic equipment. The procedure requires the removal of a module from the controller chassis. Please observe normal electrostatic discharge (ESD) precautions while performing this procedure. Adept recommends that you wear a static strap during the entire procedure. Observe the practices for network safety as detailed in section 14.7.2 of the ISO/IEC international standard 8802-3 and ANSI/IEEE standard 802.3. These standards define some of the electrical safety hazards that must be avoided or protected against for proper network installation and performance.

Instructions for MV-5, MV-8, and MV-19 Controllers

NOTE: If you are installing AdeptNet in an MV-10 controller, go to the next section for instructions.

Installing AdeptNet

1. With the Adept MV controller turned off, and wearing the disposable static strap provided, unscrew the captive screws at the top and bottom of the 040 primary processor module (located in slot 1 of the Adept MV controller). Lay this module down on a smooth, clean, static-free surface.
2. Replace the front plate of the 040 with the front plate provided in the upgrade kit.
3. Remove the Ethernet Industry Pack Module from its ESD bag and install it on the 040 processor board in the position shown in Industry Pack and Ribbon Cable on 040 Module.
4. Remove the ribbon cable from the kit, and plug the longer end (farthest from triangular fold) into the connector receptacle on the 040 as shown in Industry Pack and Ribbon Cable on 040 Module.
5. Remove the AdeptNet Transition Module from its ESD bag, and plug the shorter end of the ribbon cable into the rear of the small printed circuit board as shown in Ribbon Cable on Transition Module.
6. Remove the front-panel cover plate immediately to the left of the 040 Adept MV controller. Note that this slot does not have connections to the VME backplane.
7. Simultaneously install the 040 and the Transition Module into the Adept MV Chassis. Tighten the captive screws at the top and bottom of each front panel.

AdeptNet Hardware Installed in an Adept MV-8 or MV-19 Controller shows the completed assembly in MV-8 or MV-19 controller. Note that in an MV-5 controller, the modules are all positioned one slot to the left of that shown in AdeptNet Hardware Installed in an Adept MV-8 or MV-19 Controller, and there is no blank slot position at the far left.

With this procedure complete, restart the Adept MV controller and ensure that it works correctly.

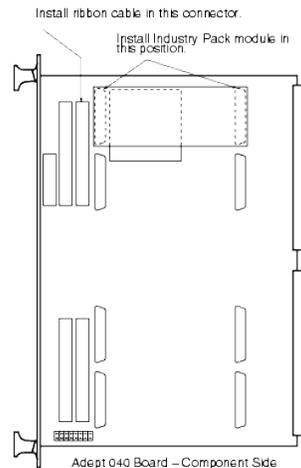


Figure 3-1. Industry Pack and Ribbon Cable on 040 Module

Installing AdeptNet

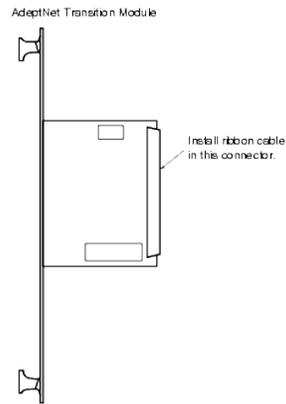


Figure 3-2. Ribbon Cable on Transition Module

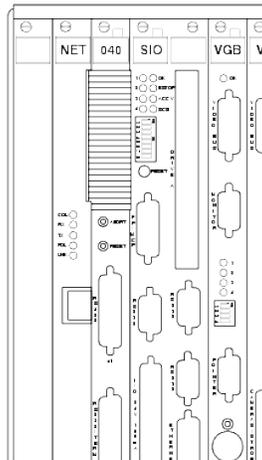


Figure 3-3. AdeptNet Hardware Installed in an Adept MV-8 or MV-19 Controller

Instructions for MV-10 Controller

NOTE: In an MV-10 controller, the AdeptNet Transition Module is installed on the right-hand side of the 040 module. You must move existing modules one slot to the right to make space for the AdeptNet module.

Installing AdeptNet

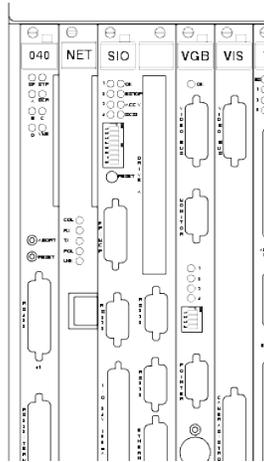


Figure 3–4. AdeptNet Hardware Installed in an Adept MV–10 Controller

1. With the Adept MV controller turned off, and wearing the disposable static strap provided, unscrew the captive screws at the top and bottom of the 040 primary processor module (located in slot 1 of the Adept MV–10 controller). Lay this module down on a smooth, clean, static–free surface.
2. Remove the SIO module in the same manner as described above.
3. Move any additional modules over one slot to the right.
4. Remove the five backplane jumper plugs from slot 3 and install them in slot 4. See the Adept MV Controller User's Guide for information on jumper plugs.
5. Install five backplane jumper plugs (shipped with AdeptNet kit) in slot 2 of the backplane.
6. Replace the front plate of the 040 with the front plate provided in the upgrade kit.
7. Remove the Ethernet Industry Pack Module from its ESD bag and install it on the 040 processor board in the position shown in [Industry Pack and Ribbon Cable on 040 Module](#).
8. Remove the ribbon cable from the kit, and plug the longer end (farthest from triangular fold) into the connector receptacle on the 040 as shown in [Industry Pack and Ribbon Cable on 040 Module](#).
9. Remove the AdeptNet Transition Module from its ESD bag, and plug the shorter end of the ribbon cable into the rear of the small printed circuit board as shown in [Ribbon Cable on Transition Module](#). Fold the ribbon cable so that it fits between the two modules without causing any crimping or interference.
10. Simultaneously install the 040 and the Transition Module into the Adept MV Chassis. Tighten the captive screws at the top and bottom of each front panel.
11. Install the SIO module into slots 3 and 4.

[AdeptNet Hardware Installed in an Adept MV–10 Controller](#) shows the completed assembly. With this procedure complete, restart the Adept MV controller and ensure that it works correctly.

3.2 Connecting to a Network

The Adept controller, once fitted with AdeptNet, offers direct connectivity to a ThinNet Ethernet network. The following are guidelines for the connection of 10BASE–T:

- ◆ Use RJ–45 10BASE–T ethernet cable to form the trunk segments of the network.
- ◆ The total length of a cable should not exceed 100m (328 ft.).
- ◆ If distances in excess of this limitation are required, use a device known as a repeater.

Installing AdeptNet

- ◆ All network interconnections must go through a hub.¹ See AdeptNet 10BASE-T Installation Requires a Hub.

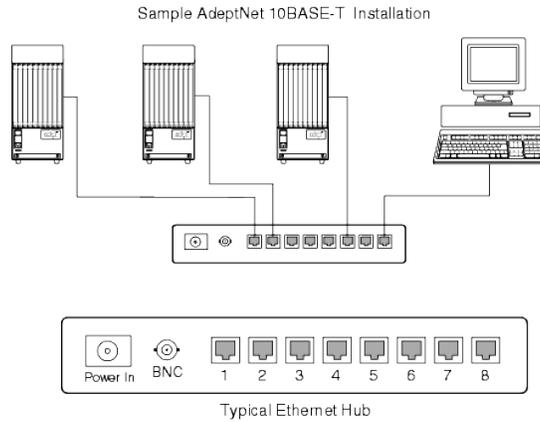


Figure 3–5. AdeptNet 10BASE–T Installation Requires a Hub

⚠ **CAUTION:** To minimize susceptibility to interference, take extra precautions when installing Ethernet in industrial environments. Take care to physically distance the Ethernet coaxial cable from noise sources such as motor cables, robot power cables, crosstalk from 10BASE–T circuits, ringing and dialing signals from telephones, high–voltage lines, switch–mode power supplies, and other sources of high–frequency or high–transient currents or voltages. Factory routed Ethernet should ideally be installed in grounded metal conduit or trunking to minimize the influence of noise and to provide physical protection.

¹

The hub in a 10BASE–T network functions as a multi–port repeater: it receives, retimes, and regenerates signals received from any attached stations.

3.3 Software Installation

Installing V+ as an Upgrade

If you have a V+ version earlier than 11.2, you need to upgrade your software to a later version. If you do not have this software at hand, contact Adept Customer Service.

To install new V+ system software you need to use the DISKCOPY utility. For instructions on using this utility, refer to revision B of the manual titled Instructions for Adept Utility Programs.

Installing AdeptNet Software Licenses

If you purchased an Adept SmartController or an AWC–based Adept MV Controller, the AdeptNet hardware is included with the system. To use AdeptNet, you need to install the AdeptNFS

Installing AdeptNet

If you purchased AdeptNet as an upgrade option, you need to install the various software options you purchased. In the package you received there is a License Agreement for AdeptTCP/IP. If you purchased either or both of the licenses for AdeptNFS or AdeptFTP, agreements for the appropriate options are also in the package. Use the INSTALL monitor command to install these software options, using the passwords given on the license agreements for your controller.

NOTE: The TCP/IP License must be installed if you are going to use any options, including NFS and FTP.

Network Configuration

The Network Configuration documentation provides instructions on how to configure the Adept controller for use on an Ethernet network.

4.1 IP Network Address

4.2 Adept Controller Configuration Using CONFIG

4.3 Network Functionality Check

4.4 Network Configuration of Non-Adept Equipment

4.1 IP Network Address

Every Adept controller, workstation, PC, or network router on a network must have a unique Internet Protocol (IP) address. An IP address is a 32-bit number, usually represented in dotted decimal notation, that is stored as part of the network configuration for the device. The IP address can be considered to consist of two portions:



For example, the default IP address for AdeptNet node 1 is 192.168.144.129. The network can be described by the network number 192.168.144.0, and the node by the number 129.

Alternatively, the network number can be 192.168.144.128, and the node number 1. The trade-off here is the number of networks versus the number of nodes in each network. In order to make it clear as to whether network numbers or node numbers are involved, AdeptNet supports the concept of subnet masking.

A subnet mask is a 32-bit number, represented in the same dotted decimal notation as an IP address. The subnet mask is used to indicate which bits of an IP address belong to a network number. The default subnet mask for AdeptNet is 255.255.255.0. This mask implies the former interpretation of the default AdeptNet IP address. (In the second example earlier in this section, the subnet mask 255.255.255.128 was assumed).

There are certain IP addresses that are reserved by convention. Also, if you intend to link your Adept controller(s) and LAN to a public WAN, such as Internet, you need to obtain a registered network number for the purpose. You can get a network ID address from a local Internet provider.

The default IP addresses chosen by Adept are suitable for connection of multiple Adept controllers onto a LAN.

4.2 Adept Controller Configuration Using CONFIG_C

The network configuration for the Adept controller is held in the system configuration file and can be viewed and modified using the CONFIG_C utility.

Explanation of the Default Configuration

Section Header

The default network configuration begins with the header:

```
.NETWORK
```

Local Machine ID (Optional)

```
LOCAL_ID = "/MACHINE `CELL1' /UID 200 /GID/200"
```

The machine-ID and the user-ID and group-ID are used for permission checking on an NFS server. There are two conventions used for mounting an NFS server:

- ◆ In UNIX-style authentication the NFS client is required to pass a machine identification and user and group IDs. This checking allows access to the server to be limited to certain nodes on the network and nodes that know the appropriate user-ID and/or group-ID to use.
- ◆ Some other systems accept what is termed null authentication when this information is not required. If this statement is omitted (the default case), null authentication is used.

If the /MACHINE keyword is not specified, then null authentication is used.

Default Remote Disk Mount

```
MOUNT 1 = "/MOUNT 'D1' /NODE 'SERVER1' /c"
```

This is the default user-selectable name used to access a disk on a server on the LAN. This server may be a file server or a gateway or any other type of server.

The default NFS remote disk has the default user-selected name D1 and refers to drive c on the server that identifies itself as SERVER1. This drive has been exported as /c. Note that the /PATH definition is case sensitive and in server-native syntax. The /PATH definition can include a subdirectory specification. The example given is the syntax used for a PC server. The IP address for the server is defined by a NODE statement for the server name.

Additional mounts can be defined at run time using the V+ FSET program instruction or monitor command.

Default Node Names and Addresses

```
NODE 1 = "/NODE 'CELL1' /ADDRESS 192 168 144 1"
```

CELL1 is the default AdeptNet user-selectable name for one of the nodes on the LAN. This name may be used anywhere an IP address is required in V+. The default configuration data file also contains the definitions for several other nodes:

```
NODE 2 = "/NODE 'CELL2' /ADDRESS 192 168 144 2"  
NODE 3 = "/NODE 'CELL3' /ADDRESS 192 168 144 3"  
NODE 4 = "/NODE 'CELL4' /ADDRESS 192 168 144 4"  
NODE 5 = "/NODE 'CELL5' /ADDRESS 192 168 144 5"  
NODE 6 = "/NODE 'CELL6' /ADDRESS 192 168 144 6"  
NODE 7 = "/NODE 'CELL7' /ADDRESS 192 168 144 7"  
NODE 8 = "/NODE 'CELL8' /ADDRESS 192 168 144 8"  
NODE 9 = "/NODE 'SERVER1' /ADDRESS 192 168 144 101"
```

Network Configuration

Note that additional nodes can be defined at run time using the V+ FSET program instruction.

Default Router

```
ROUTE_DEFAULT = "/ADDRESS 192 168 144 101"
```

This is the default address for any packet with an unknown destination.

Default Gateway

```
ROUTE_GATEWAY 1 = "/ADDRESS 192 168 199 0  
/SUBNET 255 255 255 0 /GATEWAY 192 168 144 100"
```

A gateway links two networks together with a network router. In this example the address of the remote network is 192.168.199.x. The subnet mask indicates how to interpret this address. The router through which passes all information for this remote network has the address 192.168.144.100 on the local network. Note that this configuration statement is not necessary if all remote networks are accessible through a single gateway. In this case, a ROUTE_DEFAULT statement is sufficient.

Default Local Address

```
ROUTE_SELF = "/ADDRESS 192 168 144 1 /SUBNET 255 255 255 0"
```

This defines the IP address of the local system. The network address of this particular node is 192.168.144.1, that is, node 1. The subnet mask indicates that the default address for this network is 192.168.144.x, where the x represents the node number.

For networks incorporating multiple Adept controllers, you need to change the address defined in the ROUTE_SELF statement for the additional nodes.

Note: On an Adept SmartController, the default IP address is displayed on bottom side of the controller chassis.

Configuration Limits

Number of Disk Mounts

There can be a maximum of nine remote disk mounts for each V+ System. Only two of these can be specified in the configuration file; others can be specified using the FSET program instruction or the FSET monitor command. Refer to Mounting a Remote Disk for more information.

Number of Nodes

There can be a maximum of 32 node names specified to V+. Only 9 of these can be specified in the configuration file; the rest have to be specified (only once at power-up) in a program using the FSET program instruction or the FSET monitor command.

Number of Gateways

There can be a maximum of two gateways defined by V+. These must both be specified in the configuration file.

4.3 Network Functionality Check

Perform the following checks to verify the network is functioning properly.

- ◆ on the Adept MV controller, the POL and LNK LEDs should be illuminated, on the Adept SmartController, the LAN LED should be illuminated.
- ◆ use the NET monitor command to verify the IP address and the options installed.
- ◆ use the PING monitor command to check communications.

4.4 Network Configuration of Non–Adept Equipment

PC–Based Server

The procedure for setting up and configuring a PC server depends on the network package you purchased for the PC. Typically you need to specify two things:

1. The network IP address for the server. This must be the same as that defined on the Adept side for the server. The AdeptNet default IP address for Server #1 is 192.168.144.101.
2. For NFS or FTP, the drive or drives that you wish to make available as accessible network resources. In some PC NFS packages this specifying is termed drive mapping or drive exporting. In the default Adept MV configuration example of the section Default Remote Disk Mount, the MOUNT directive specified that the PC's c drive was to be used as a server drive.
3. For FTP you may also need to specify user and password information for the clients. Refer to documentation provided by your server software supplier for more details.

AdeptNFS:Network File System

This documentation describes Adept's Network File System software.

[Introduction to Adept NFS](#)

[System Requirements](#)

[Mounting a Remote Disk](#)

[Referring to Remote Drives](#)

[V+ Monitor Command Support](#)

[V+ Program Instruction Support](#)

5.1 Introduction to AdeptNFS

AdeptNFS allows an Adept controller to be an NFS client, accessing remote disk resources from some non-Adept server. Such a server is typically a PC or UNIX workstation.

5.2 System Requirements

AdeptNFS requires the following components:

- ◆ An Adept MV Controller with the AdeptNet Ethernet option, an AWC-based Adept MV controller, or an Adept SmartController¹
- ◆ The Software Licenses for AdeptTCP/IP and AdeptNFS
- ◆ V+ version 11.2 edit G or later

¹If you have an AWC-based Adept MV Controller, or an Adept SmartController, the Ethernet option is not required because the AdeptNet hardware is included with the system.

5.3 Mounting a Remote Disk

Default Mount at System Initialization

At boot up, provided that the NFS server software is already running, the Adept controller performs the necessary network initialization and mounting procedure to mount the external drive(s) defined by the MOUNT statements contained in the configuration data file. See [Network Configuration](#) for details about the MOUNT statement.

If the NFS server cannot be recognized, this mount fails. A failure occurs, for example, if the server software is not running or if the server is disconnected from the network. Use the NET monitor command after startup to determine which mounts were successful. For details, see the [NET monitor command](#) documentation.

You cannot initialize the network after the V+ system has completed its initialization. You can

AdeptNFS:Network File System

use the PING monitor command to verify network connectivity. If a problem exists with the network connection, correct it and restart the V+ system.

Mounting by Using the FSET Monitor Command

You can use the FSET monitor command to mount a remote disk. For example, to define a new NFS mount with the device name DISK2 to access the exported directory /c of the node (server) called PC1, type

```
fset nfs /mount 'DISK2' /node 'PC1' /path '/c'
```

Mounting by Using the FSET Program Instruction

Alternatively, you can use the FSET program instruction to mount a remote disk. Consider the following code example:

```
; Attach logical unit number for file

ATTACH (lun, 4) "NFS"
error = IOSTAT (lun)
IF error < 0 THEN
    TYPE "ATTACH error: ", $ERROR(error)
    GOTO 100
END

; Mount remote drive

FSET (lun) "/MOUNT 'DISK1' /NODE 'SERVER1' /PATH '/c'
error = IOSTAT (lun)
IF error < 0 THEN
    TYPE "MOUNT error: ", $ERROR(error)
    GOTO 100
END
```

NOTE: You must complete the FSET instruction on one line in the actual code.

5.4 Referring to Remote Drives

Use the following syntax to fully specify any file in the V+ environment:

```
device>disk_unit: directory_path\filename
```

For example, the full specification of a file called MY_FILE.DAT, residing in the directory NEW_DATA on an Adept controller's local C drive, is:

```
DISK>C:\NEW_DATA\MY_FILE.DAT
```

With the NFS functionality, the device name NFS has been added to V+. If the same file is in a directory with the same name but on a remote disk mount called DISK1, the full description is:

```
NFS>DISK1:\NEW_DATA\MY_FILE.DAT
```

AdeptNFS:Network File System

You can use the DEFAULT monitor command to set the default disk device, unit, and path to reference an NFS device. For more detail, see the documentation for the DEFAULT monitor command

NOTE: Directories and files on a local disk can be specified with any combination of uppercase and lowercase letters. However, for compatibility with the NFS server, all directories and programs on a remote disk should be specified with lowercase letters. The name of a remote mount is processed locally by V+ and thus can be specified with uppercase or lowercase letters.

5.5 V+ Monitor Command Support

The FSET monitor command can be used to define nodes and mounts in addition to those defined in the V+ configuration file. After a remote disk has been mounted, you can use the standard V+ monitor commands to display directories, create or delete directories, and copy, rename, or delete files. NFS automatically tries any request up to three times with a ten second timeout between retries. The monitor commands supported are:

Task	V+ Monitor Command
File creation, copying and deletion	<u>STORE</u> , <u>FCOPY</u> , <u>FDELETE</u>
File accessing, reading, writing	<u>LOAD</u> , <u>FLIST</u> , <u>STORE</u>
File directory operations	<u>FDIRECTORY</u>
Subdirectory creation and deletion	<u>FDIRECTORY/C</u> , <u>FDIRECTORY/D</u>

The FORMAT command is not supported for remote disk operation.

Examples

To set the default disk to the local Adept controller C drive, type

```
def = disk>c:
```

Assuming a remote mount named DISK1 exists, give a directory listing of the files on the server in the default directory of the mounted drive. Type

```
fdir nfs>disk1:
```

To create a new directory called backups on the server, type

```
fdir/c nfs>disk1:\backups\
```

To change the default disk drive to the server DISK1, type

```
def = nfs>disk1:
```

To copy the file example.v2 found in the programs directory on the Adept controller's c drive to the directory called backups on the server, type

```
fcopy \backups\example.v2 e:\ps\programs\example.v2
```

5.6 V+ Program Instruction Support

All the standard V+ file instructions are valid for remote files located on a server drive. NFS automatically tries any request up to three times with a ten second timeout between retries. The FSET instruction can be used to allow mounting remote drives (as illustrated is [Mounting a Remote Disk](#)) and to allow the specification of new nodes on the network. For more details, refer to the documentation for the [FSET program instruction](#).

Example Code: Writing to and Reading from a Remote File

```
$wr.message = "This is a test message"
$rd.message = ""
; Attach disk logical unit to NFS device

    ATTACH(lun, 4) "NFS"
    error = IOSTAT(lun)
    IF error < 0 THEN
        TYPE "ATTACH error: ", $ERROR(error)
        GOTO 100
    END
; Mount drive

    FSET(lun) "/MOUNT 'DISK1' /NODE 'SERVER1' /PATH '/c'"
    error = IOSTAT(lun)
    IF error < 0 THEN
        TYPE "MOUNT error: ", $ERROR(error)
        GOTO 100
    END
; Open the file "txt_file.dat" in the directory "programs" on the
; server so that information can be appended to it

    FOPENA(lun) "disk1:\programs\txt_file.dat"
    error = IOSTAT(lun)
    IF error < 0 THEN
        TYPE "ERROR opening file: ", $ERROR(error)
        GOTO 100
    END
; Write the test message

    WRITE(lun) $wr.message

; Close the file

    FCLOSE(lun)
; Re-open the file for read-only access

    FOPENR(lun) "disk1:\programs\txt_file.dat"
    error = IOSTAT(lun)
    IF error < 0 THEN
        TYPE "ERROR opening file: ", $ERROR(error)
        GOTO 100
    END
; Read the message back from the file

    READ(lun) $rd.message
; Type the message out - it should be the same as $wr.message

    TYPE $rd.message
; Close the file and detach
```

AdeptNFS:Network File System

FCLOSE (lun)
DETACH (lun)

100 RETURN

AdeptNFS:Network File System

AdeptFTP and AdeptTFTP: File Transfer Protocol

[Introduction](#)

[System Requirements](#)

[How AdeptFTP Works](#)

[Using the AdeptFTP Server](#)

[Using the AdeptFTP Client](#)

[AdeptFTP and File Types](#)

[Using AdeptTFTP](#)

6.1 Introduction to AdeptFTP and Adept TFTP

FTP is a standard file transfer protocol designed to work on top of the standard TCP/IP protocol. The FTP protocol provides a means for moving files from one system to another. Moving can be between Adept controllers or non-Adept equipment that also has the FTP protocol. The transfer can occur between nodes in the same Local Area Network (LAN), or nodes on a Wide Area Network (WAN).

FTP transfers between two nodes on a LAN or WAN require that one of the nodes be in server mode while the other node is in client mode. Unlike the situation with NFS (Network File System), where the term server applies to a node that shares its disk resources with the client nodes, an FTP server merely responds to a client's requests for file transfers. File transfer can occur either from the client to the server, termed a PUT operation, or from the server to the client, termed a GET operation.

Adept's implementation of FTP, AdeptFTP, is in the form of V+ application utilities. These utilities are found on the Adept Utility Diskette supplied with your system. These utilities are written in V+ and utilize the direct TCP/IP protocol access offered by AdeptTCP/IP. Refer to [*Writing Custom Protocols Using AdeptTCP/IP*](#) for more information on AdeptTCP/IP.

AdeptTFTP is a simplified form of the AdeptFTP protocol which uses the User Datagram Protocol (UDP) and provides no security features. It is often used by servers to boot diskless workstations, X-terminals, and routers. For details, see [Using AdeptTFTP](#).

6.2 System Requirements

To use AdeptFTP you require the following:

AdeptFTP and AdeptTFTP: File Transfer Protocol

- ◆ An Adept MV Controller with the AdeptNet Ethernet option, an AWC-based Adept MV controller, or an Adept SmartController¹
- ◆ The Software Licenses for the AdeptTCP/IP and AdeptFTP options
- ◆ V+ version 11.2 edit G or later
- ◆ The Adept Utility Disk version 11.2 or later. For V+ versions, 14.0 or later, the Adept V+ Software CD.

¹If you have an AWC-based Adept MV Controller, or an Adept SmartController, the Ethernet option is not required because the AdeptNet hardware is included with the system.

6.3 How AdeptFTP Works

An FTP server listens on TCP port number 21 for TCP connection requests from FTP clients. When a TCP connection is established, the FTP server starts accepting standard FTP commands from the FTP client. The FTP client program provides a user interface that converts user commands to standard FTP commands. The FTP server always sends replies to the FTP client to indicate the status of commands. Some client commands elicit multiple server replies.

During file-transfer operations, a second TCP connection is established to transfer the file data. This second connection is established with the FTP client providing a TCP server on the TCP port number 20. This detail is transparent to the user, who is provided with a command-line text interface in the V+ monitor window.

There are two utility programs for AdeptFTP included on the Adept V+ Software CD-ROM¹:

- ◆ FTP_SRVR.V2 is the server utility program.
- ◆ FTP_CLNT.V2 is the client utility program.

¹ For V+ versions prior to V+ 14.0, the AdeptFTP Utility programs are provided on the Adept Utility Disk.

Related Topics

[Using the AdeptFTP Server](#)

[Using the AdeptFTP Client](#)

6.4 Using AdeptFTP Server

The AdeptFTP server program allows an FTP client (an Adept controller or other FTP device) to perform the following functions:

- ◆ Log in. Any user is accepted, and no password is necessary.
- ◆ List a directory.
- ◆ Get a detailed listing of a directory.
- ◆ Type the name of the current directory.
- ◆ Change the current directory. A physical device may be specified, so that the user can in fact specify the current directory as NFS>disk: and transfer files (inefficiently) from the network back to the network.

AdeptFTP and AdeptTFTP: File Transfer Protocol

NOTE: This is a special note to users of UNIX FTP clients. Because the Adept subdirectory delimiter, "\", is also the UNIX prefix for special characters, subdirectories have to be specified with "\\". For example, if the desired subdirectory is DISK>c:\test\, then the correct name to type to the UNIX FTP client is DISK>c:\\test\\.

- ◆ Using the GET operation, transfer a file from a local disk on the Adept controller running the AdeptFTP server utility.
- ◆ Using the Put operation, transfer a file to a local disk on the Adept controller running the AdeptFTP server utility. As specified in the FTP protocol standard, any existing file with the same name is overwritten without warning.
- ◆ Log out.

The common file operations not supported are file deletion and subdirectory creation.

As described in the following sections, perform these steps to place an Adept controller in FTP server mode:

- ◆ Load and execute the server program.
- ◆ Stop the server program.

Loading and Executing the Server Program

If not already done, create a utility (\util\) directory on the C drive of your Adept controller. For V+ versions For V+ version 13.x or earlier, copy the entire contents of the Adept Utility Disk received with your system into this directory using the DISKCOPY utility program. For V+ versions 14.x or later, copy the V+ Utility programs from the V+ software CD.

Load the file FTP_SRVR.V2 from the \util\ directory by typing the command

```
load \util\ftp_srvr
```

The file contains the utility program a.ftp_srvr. Typing the following command executes the server program as task 1:

```
execute 1 a.ftp_srvr
```

You can choose any user task you wish for the server program. The program displays the title line:

```
AdeptFTP Server Program (Version....)
```

in the V+ monitor indicating that the Adept controller is in server mode.

The system is now running and will respond to requests from an FTP client. An FTP client can be another Adept controller running the AdeptFTP client utility, or some other device running FTP.

NOTE: Because the Adept controller is multi-tasking, you can use the controller to perform other tasks while the AdeptFTP server program is running.

Stopping the Server Program

To quit the AdeptFTP server, use the ABORT monitor command. For example, typing

```
abort 1
```

aborts the server running as task 1.

6.5 Using AdeptFTP Client

The AdeptFTP client program provides a text-based user interface. Use uppercase or lowercase when entering the commands. This section describes the available user commands.

Loading and Executing the Client Program

1. To load the file ftp_clnt.v2 from the \util\ directory, type

```
load \util\ftp_clnt
```

The file contains the program a.ftp_clnt.

2. To execute the client program in task 1, type:

```
execute 1 a.ftp_clnt
```

The system then asks:

```
use the V+ monitor window (Y/N)?
```

Type N to create another window. (N = create another window for FTP, Y = use the current monitor window for FTP.)

After the window opens, the *Adept FTP*> command prompt is displayed. You can type FTP commands after this prompt. Press ENTER or RETURN to execute the command.

Help Command

```
HELP command
```

Display a one-line description about the specified command.

Examples:

To display a list of the available commands, type help

To provide help information about the FTP CONNECT command, type help connect

Connection Commands

CONNECT

```
CONNECT host
```

Connect to a specified host, where host can be the name of the server node (such as server1) or a dotted–decimal address (like 192.168.144.101). (If necessary, the user is asked to supply a name, password, and account number. This is dependent on the server host's security arrangement.)

NOTE: The user name and password may be case sensitive.

Example:

```
connect server1  
username:
```

DISCONNECT

```
DISCONNECT
```

Disconnect from the host to which the client is currently connected. It is then possible to connect to a different host with another CONNECT command.

QUIT

```
QUIT
```

Exit the FTP client program. Note that this command includes an implicit DISCONNECT.

Directory and File Commands

DIR

```
DIR -L
```

List directory contents for the current directory. Specifying `-L` gives a more detailed listing that includes file size, and date and time of file creation.

PWD

```
PWD
```

Display the current directory on the server.

CD

```
CD directory
```

Change the current directory on the server to the specified directory.

Example:

```
cd \programs
```

AdeptFTP and AdeptTFTP: File Transfer Protocol

Changes the directory path in the server to \programs.

GET

```
GET remotefile localfile
```

Copy the file remotefile from the server and place it in the local file with the same name or with the name localfile, if specified.

Example 1:

```
get \programs\widget.v2
```

Transfers (copies) the file widget.v2 from the server directory \programs to the \programs directory of the Adept controller.

Example 2:

```
get \programs\widget.v2 \programs\widget.v2
```

Performs the same transfer as does the GET command in example 1.

Example 3:

```
get \programs\widget.v2 \mydir\newfile.v2
```

Transfers (copies) the file widget.v2 from the server directory \programs\ to the file newfile.v2 in the directory \mydir\ of the Adept controller.

PUT

```
PUT localfile remotefile
```

Copy the file localfile from the local V+ system to the server. The file on the server is called localfile unless remotefile is specified, in which case the file is called remotefile.

Example 1:

```
put widget.v2
```

Transfers (copies) the file widget.v2 from the Adept controller to the current directory of the FTP server.

Example 2:

```
put \mydir\widget.v2 \programs\newfile.v2
```

Transfers (copies) the file widget.v2 in the directory \mydir\ to the FTP server directory \programs\ with the name newfile.v2.

NOTE: This note is for users connecting to a UNIX FTP server. UNIX FTP servers time-out a connection usually after about five minutes of inactivity. When this occurs, the AdeptFTP program reports a V+ error like File not opened for most commands attempted. The solution is to

use the DISCONNECT command and then the CONNECT command.

6.6 AdeptFTP and File Types

The remote FTP client may provide a file-type selection, either ASCII or binary. ASCII files are correctly preserved when transmitted as binary files, but binary files must be transmitted only as binary files. The AdeptFTP server accepts both types of files but stores them both as binary files, because ASCII files may be stored without error as binary files. Similarly, an AdeptFTP client program transmits all files, whether ASCII or binary, as binary files.

6.7 Using AdeptTFTP

The TFTP device allows read and write access to files on a TFTP server. This device provides a way to read files from a host machine without requiring any additional configuration steps, such as those required for the NFS device. (However, the NFS device provides greater functionality.)

The TFTP device has only one channel and supports both READ and WRITE access. Because the device has one channel, there can be only one active transfer at a time.

Note: The AdeptTFTP server only permits writing to a new file. If you attempt to WRITE to a file that already exists, V+ returns the error message *Nonexistent file*.

File already exists error defined in the protocol.

When accessing the TFTP device, the syntax is:

```
TFTP>server:file_spec
```

Parameters

server The server name or IP address (in dotted-decimal format)

file_spec Full or partial pathname of the desired file, the syntax of which is determined by the type of the server. For example, a Unix server will require the use of "/" for path delimiters, and a Windows server will require "\" instead. In addition, some servers may allow the use of relative pathnames while others may always require the use of full pathnames.

AdeptFTP and AdeptTFTP: File Transfer Protocol

Examples:

Accessing a Unix server:

```
LOAD TFTP>172.16.1.1:/home/user/file.v2
```

Accessing a Windows server:

```
FLIST TFTP>172.16.1.2:c:\autoexec.bat
```

Accessing the Adept TFTP server which supports relative pathnames:

```
LOAD TFTP>172.16.1.2:file.dat
```

Using the CD monitor command to set a prefix (which is not interpreted in any way by the CD command, and is prepended to subsequent file specifications):

```
CD TFTP>it_server:
```

```
LOAD d:\data\util.v2
```

Writing Custom Protocols Using AdeptTCP/IP

This documentation provides information on using the TCP/IP protocol for peer-to-peer communication from a user V+ program.

Introduction

System Requirements

Ports and Sockets

How a TCP Connection is Established

Example V+ Code

7.1 Introduction

TCP/IP support on the Adept controller allows a V+ program to communicate with another TCP/IP device (Adept controller or other) in either a client or server mode. In client mode, the Adept controller solicits information from a particular node on the network, termed the server. In server mode, the Adept controller responds to requests for information from one or more client nodes.

7.2 System Requirements

To use AdeptTCP/IP you require the following:

- ◆ An Adept MV Controller with the AdeptNet Ethernet option, an AWC-based Adept MV controller, or an Adept SmartController¹
- ◆ The Software Licenses for AdeptTCP/IP
- ◆ V+ version 11.2 edit G or later

¹If you have an AWC-based Adept MV Controller, or an Adept SmartController, the Ethernet option is not required because the AdeptNet hardware is included with the system.

7.3 Ports and Sockets

TCP uses what is termed port addressing to deliver information to the appropriate application-layer program. A port is a 16-bit address. The term socket refers to the concatenation of the IP address of the node and the port number used by the application-layer program. The concept of opening a socket in an application layer program facilitates multiple sessions on a single network node.

For example, two V+ tasks, or sessions, on the same controller may communicate simultaneously with a network server. Because both tasks run on the same network node, they have the same IP address. However, through the use of sockets, the combination of the IP address and a port number uniquely identifies the task to which TCP needs to route the information, provided that the port numbers are different for each task.

NOTE: Some standard application-layer services use defined port numbers in the range of 0 to 255. For example, it is conventional for an FTP server application program to use port #21 as a control channel. Users developing their own protocols and applications should use port numbers out of the range 0 to 255.

7.4 How a TCP Connection Is Established

TCP is naturally partitioned into a client side and a server side. The user's program instructions differ for the client and server sides.

Server Side

To establish a TCP server connection, a V+ TCP server task informs the local TCP driver that it wishes to accept all connection requests on a particular local port number. Any client node on the network now is able to connect to this server using the server IP address and local port number. Refer to [Client Side](#) below for information about this local port number.

Note that the server does not need to know the remote IP address or remote port number of the client. As you will see, a client does need to know the address of the server node. When a client makes a connection to the server a client handle is allocated on the server. From that point on, the handle number is used to identify the client. When a client disconnects from the server, the handle is freed up for use by any new client wishing to make a connection.

In some cases a simple single-client server is required. This is often referred to as peer-to-peer communication.

Client Side

In order to establish a connection on the client side, the client must know the IP address and the port number of the desired server. The client can arbitrarily select the local port number, and typically the client's TCP driver automatically generates it. With this information, the client can request the establishment of a connection by transmitting a connection request to the server.

When the server's TCP driver receives this request, it verifies that the client's remote port designation matches the server's local port. If so, a successful connection message is returned to the client, and the server task is notified.

Once the connection is successfully established, both the server task and the client task can begin bi-directional data transmissions.

7.5 Example V+ Code

AdeptTCP/IP Echo Server

The following example program is for an echo server. An echo server is useful for testing network connections. This program allows up to five clients to connect to the server. Any message received from a client is echoed back. The TCP server employs existing V+ program instructions, which have been enhanced to interface to the TCP driver.

For more details on the V+ program instructions in the example program, see [Using AdeptNet](#)

Writing Custom Protocols Using AdeptTCP/IP

with V+.

NOTE: TCP "server" program must be running before the "client" program is started.

```
.PROGRAM tst.tcp.srvr()  
-----  
; ABSTRACT: Example TCP/IP server program. The server program reads any  
; incoming strings, displays them on the monitor, and echoes them  
; back to the client node.  
;  
; The program may be terminated by aborting.  
;  
; INPUT PARMS: None  
; OUTPUT PARMS: None  
-----  
  
    AUTO do_wait, handle, lun, no_wait, repeat_loop, status  
    AUTO $in.str, $out.str  
  
; Initialize constants.  
  
    no_wait = 1  
    do_wait = 0  
  
    lun = 7  
  
; Attach the TCP device (mode=4 means allocate next available LUN).  
  
    ATTACH (lun, 4) "TCP"  
    status = IOSTAT(lun)           ;Check status of ATTACH  
    IF status <0 THEN  
        TYPE "Error from ATTACH:", $ERROR(status)  
        GOTO 100  
    END  
  
; Set up a socket with local port #1234, to accept 5 client  
; connections, with a buffer size of 1024 bytes.  
  
    FOPEN (lun, 16) "/LOCAL_PORT 1234 /CLIENTS 5 /BUFFER_SIZE 1024"  
    status = IOSTAT(lun)           ;Check status of ATTACH  
    IF status <0 THEN  
        TYPE "Error from FOPEN:", $ERROR(status)  
        GOTO 100  
    END  
  
    repeat_loop = TRUE  
    WHILE repeat_loop DO  
        WAIT  
        READ (lun, handle, do_wait) $in.str  
        status = IOSTAT(lun)  
        CASE status OF  
            VALUE 1:           ;Success--echo string back  
                $out.str = $in.str  
                TYPE "Received string: ", $out.str  
                WRITE (lun, handle) $out.str, /N  
  
            VALUE 100:         ;New connection opened  
                TYPE "Newconnection established. Handle =", handle  
  
            VALUE 101:         ;Connection closed  
                TYPE "Connection closed. Handle =", handle
```

Writing Custom Protocols Using AdeptTCP/IP

```
FCMND (lun, 600) $INTB(handle) ;Deallocates handle
status = IOSTAT(lun) ;Check status of FCMND
IF status <0 THEN
    TYPE "Error from FCMND:", $ERROR(status)
    GOTO 100
END
WAIT
VALUE -526: ;No data received
ANY ;Some other error
TYPE "Error during READ: ", $ERROR(status)
GOTO 100
END
END
END

; Close all open client connections, and release the logical unit.

100 FCLOSE (lun)
DETACH (lun)

RETURN

.END
```

AdeptTCP/IP Client

The following program example is a simple client user program for use in conjunction with the previous `tst.tcp.srvr` example. The program makes a connection to a TCP server (`server1`) and sends a message string. The echoed string is then read and displayed on the screen.

The TCP client employs existing V+ program instructions, which have been enhanced to interface to the TCP driver. For full documentation on the V+ program instructions refer to [Using AdeptNet with V+](#).

```
.PROGRAM tst.tcp.clnt()
;-----
; ABSTRACT: Example TCP/IP client program.
;
; NOTES: This program will continuously ask for an input string. The
; input string is displayed, the string is then sent to the server
; controller, and the server response is displayed. If the string
; is "quit", the client program stops and the connection is broken.
; Reexecuting the program, reestablishes the connection with the
; server.
;
; INPUT PARMS: None
; OUTPUT PARMS: None
;-----
    AUTO do_wait, lun, mlun, repeat_loop, status
    AUTO $inp_str, $read_str

; Initialize constants.

    do_wait = 0

; Attach to the TCP device

    ATTACH (lun, 4) "TCP"
    TYPE "lun: ", lun
    status = IOSTAT(lun) ;Check status of ATTACH
    IF status <0 THEN
```

Writing Custom Protocols Using AdeptTCP/IP

```
TYPE "Error from ATTACH: ", $ERROR(status)
GOTO 100
END

; "cell2" is the name this controller uses to refer to the server machine,
; as defined either in the default controller configuration or by using an
; FSET command or instruction. A local port number is automatically
; established, since the /LOCAL_PORT keyword is absent. The remote port
; number for the server must be known--"1234" in this case. The buffer
; size is set to 1024 bytes.

FOPEN (lun, 0) "cell2 /REMOTE_PORT 1234 /BUFFER_SIZE 1024"
status = IOSTAT(lun) ;Check status of FOPEN
IF status <0 THEN
TYPE "Error from FOPEN: ", $ERROR(status)
GOTO 100
END

; Attach the V+ Monitor and prepare for I/O.

mlun = 4
ATTACH (mlun)
FOPEN (mlun)

; Loop until string is "quit". Send the str and read the echo from server.

repeat_loop = TRUE
WHILE repeat_loop DO
TYPE "Input a string: ", /S
READ (mlun) $inp_str ;Get input from Monitor
WRITE (lun) $inp_str ;Send to the server
TYPE "The string typed is: ", $inp_str
TYPE
IF $inp_str == "quit" THEN
repeat_loop = FALSE ;We're going to stop
END
READ (lun, do_wait) $read_str ;Get reply from the server
TYPE "The echoed string is: ", $read_str
END

; Close open connection and access to Monitor, and release logical units.

100 FCLOSE (lun)
DETACH (lun)
FCLOSE (mlun)
DETACH (mlun)

RETURN

.END
```

Writing Custom Protocols Using AdeptTCP/IP

AdeptUDP: User Datagram Protocol

This documentation describes Adept's implementation of the User Datagram Protocol.

Introduction to AdeptUDP

Sending and Receiving Data

8.1 AdeptUDP Introduction

The UDP device is used for datagram communications. UDP delivers data a packet¹ at a time, and delivery is not guaranteed, nor is the order of delivery guaranteed. This contrasts to the TCP device which delivers data in a byte stream, in the correct order, with delivery guaranteed.

Although data delivery is not guaranteed, UDP is still extremely useful for applications such as live update of status info. In applications of this nature, loss of data is not critical, because old data is quickly invalidated by newer data. Other advantages of UDP are that it has a lower overhead than TCP. In addition, in V+ UDP has faster delivery than TCP.

UDP communication is naturally structured as between a client and a server. The server listens for packets from a client. In this respect, the V+ UDP device uses a programming model somewhat similar to the TCP device.

Sending and Receiving Data with UDP

In V+, the WRITE instruction sends the contents of the parameter strings. The READ instruction receives the next packet into the parameter string variables, filling them with up to 512 bytes of packet data. Any additional bytes of data are lost.

The following examples illustrate how to send data between the UDP server and the client.

Example 1: Client-Initiated Communication

Typical client program instructions (which should include checks of the status after every I/O operation):

```
ATTACH(lun) "UDP"

; Client connect to port 1234 on server 10.12.34.56.

FOPEN(lun) "10.12.34.56/REMOTE_PORT 1234"

WHILE TRUE DO
    ...

; Send a packet of data to the server.

WRITE(lun) $out[0], $out[1], $out[2], $out[3], /S

; Read a packet from the server.

READ(lun) $in[0], $in[1], $in[2], $in[3]
    ...
```

AdeptUDP: User Datagram Protocol

```
END  
  
FCLOSE(lun)  
DETACH(lun)
```

Example 2: Server-Initiated Communication

Typical server program instructions (which should include checks of the status after every I/O operation):

```
ATTACH(lun) "UDP"  
  
; Start having the server listen on port 1234.  
  
FOPEN(lun, 16) "/LOCAL_PORT 1234"  
  
WHILE TRUE DO  
    ...  
  
    ; Read a packet from the client.  
  
    READ(lun) $in[0], $in[1], $in[2], $in[3]  
  
    ; Send a packet of data to the client.  
  
    WRITE(lun) $out[0], $out[1], $out[2], $out[3], /S  
    ...  
END
```

Related Topics

V+ Keywords Related to AdeptNet

¹A packet is a group of bytes that are transmitted as a unit. The V+ system can send and receive up to 512 bytes in each packet.

Using AdeptNet with V+

This documentation contains a list of V+ keywords and system messages relevant to the AdeptNet software.

V+ Keywords Related to AdeptNet

Click the underlined keyword name in the following table for more detailed information on syntax, parameter descriptions, and usage considerations.

NOTE: V+ monitor commands are documented in the [V+ Operating System Reference Guide](#). Program instructions and real-valued functions are documented in the [V+ Language Reference Guide](#).

V+ Keyword	Function
FSET monitor command	Set or modify attributes of a graphics window, serial line, or network device related to AdeptNet.
NET monitor command	Display status information about the AdeptNet option. Also display details about the remote mounts that are currently defined in the V+ system.
PING monitor command	Test the network connection to a node.
ATTACH program instruction	Make a device available for use by the application program.
FCMND program instruction	Generate a device-specific command to the input/output device specified by the logical unit.
FOPEN program instruction	Create and open a new graphics window or TCP connection, or open an existing graphics window for subsequent input or output.
FSET program instruction	Set or modify attributes of a graphics window, serial line, or network device related to AdeptNet.
ID real-valued function	Return values that identify the configuration of the current system.
IOSTAT real-valued function	Return status information for the last input/output operation for a device associated with a logical unit.
NETWORK real-valued function	Return network status and IP address information.

Using AdeptNet with V+

<u>READ program instruction</u>	Read a record from an open file or from an attached device that is not file oriented. For an AdeptNet device, read a string from an attached and open TCP connection.
<u>WRITE program instruction</u>	Write a record to an open file, or to any I/O device. For an AdeptNet device, write a string to an attached and open TCP connection.

System Messages Related to AdeptNet Software

***Directory not empty* (-571)**

Explanation: The operation attempted to remove an NFS directory that was not empty.

User action: Delete the directory's contents before deleting the directory.

***File name too long* (-570)**

Explanation: The file name in an NFS operation was too long.

User action: Shorten the file name.

***File too large* (-569)**

Explanation: The NFS operation caused a file to grow beyond the server's limit.

User action: Close the file, open a new file, and retry the previous operation.

***Invalid network address* (-561)**

Explanation: This error occurs when an NFS server has not correctly exported the path being accessed or when an IP network address specified is not of class A, B, or C.

User action: Check the IP addresses used to refer to network nodes.

***Invalid network resource* (-560)**

Explanation: This error occurs when referring to a node that has not been defined.

User action: Check the node definitions.

***Is a directory* (-568)**

Explanation: The caller specified a directory in a non-directory NFS operation.

User action: Specify a directory in the operation.

***Network connection closed* (101)**

***Network connection opened* (100)**

***Network connection terminated* (-565)**

Explanation: This error occurs when input or output operations are attempted on a network connection that has already been terminated.

User action: Re-establish the network connection, and retry the original operation.

***Network resource name conflict* (-564)**

Explanation: A network name as specified for an NFS mount matches an existing network name such as an NFS disk name.

User action: Choose a different name.

***Network timeout* (-562)**

Explanation: This error occurs when a network transaction is initiated but no reply is received from the server.

User action: Check network integrity. Make sure the server is up and running. Make sure the correct IP address is being used.

***NFS error* Code n (-1200 to -1299)**

Explanation: Because NFS returns errors that do not have corresponding meaning in V+, many NFS errors are returned as a variable NFS error. Errors in this range have the following interpretation: If error number $-(1200+n)$ is returned, the NFS error code n was returned. Following are the currently known NFS error codes:

NFS error code n	Explanation
19	No such device.
30	Read-only file system. A write operation was attempted on a read-only file system.
69	Disk quota exceeded. The client's disk quota on the server has been exceeded.

99	The server's write cache used in the WRITECACHE call was flushed to disk.
----	---

***Not a directory* (-567)**

Explanation: The caller specified a non-directory in an NFS directory operation.

User action: Specify a directory in the operation.

***Not owner* (-566)**

Explanation: The caller does not have the correct ownership to perform the requested NFS operation.

User action: Modify the local_id statement in the V+ configuration file (using the config_c utility) as required to gain access to the server. You may also need to change the access setup on the server itself.

***Out of network resources* (-559)**

Explanation: This error applies to many circumstances. Listed below are several possible cases:

1. Too many ports are simultaneously in use for TCP and NFS; there are no more buffers to use for incoming and outgoing packets.
2. Too many drives are being mounted.
3. Too many NFS calls were made simultaneously from separate tasks to a nonfunctional NFS server.
4. Too many node names are being defined.
5. An incoming IP packet was fragmented into too many pieces and V+ was unable to reassemble it. (This is a highly unlikely occurrence.)

User action: Correct the problem generating the error.

***Remote has not exported network resource* (-563)**

Explanation: The NFS server has not exported the designated drive for use by the client.

User action: Check the NFS server setup.