FTPD

An FTP Daemon NLM for Novell Netware 386 3.11

Administrators Guide

August 9, 1992

MurkWorks
P.O. Box 631
Potsdam, NY 13676 USA

info@murkworks.com

FTPD V1.31 Administrators Guide

This software is not shareware, freeware nor public domain. It has been provided to you in a limited timed demo format. You may use this software without obligation until the demonstration period expires. When the demonstration period has expired, the software will no longer function. If you wish to continue to use the software you must register it with MurkWorks and pay a per server license fee. See the accompanying order form for pricing and ordering information. Dis-assembly or patching of the software to provide execution beyond the end of the demonstration period is expressly forbidden.

Demo Version -- No Warranty

MurkWorks makes no warranty of fitness or suitability for a particular purpose. Although we have made every effort to ensure the reliable and satisfactory operation of this software, we do not warrant it in any way. You the user assume all responsibility in its use and operation. MurkWorks shall not be held liable for any loss of any kind.

Licensed Version -- Limited Warranty

If you have registered your copy of the software and paid a license fee to MurkWorks, this software is covered by a limited warranty for the first sixty (60) days from the date of receipt of the license fee. Should the software fail or prove unsatisfactory during that time period MurkWorks sole remedy to you will be the reimbursement of your license fee. In no event will MurkWorks be held liable to you for any damages, including lost profits, lost savings or other incidental or consequential damages arising out of the use of this software or the inability to use this software.

Trademarks

Novell and Netware are registered trademarks, and Netware NLM, and Netware Loadable Module are trademarks of Novell, Inc. Other computer and software names are registered trademarks of their respective manufacturers.

This product was developed using Network C for NLMs™, Novell®'s toolkit for developing server-based applications for the Netware® 3.x operating system.

Additional Information

For additional information, write to us via postal mail:

MurkWorks
P.O. Box 631
Potsdam, NY 13676 USA

Or send electronic mail to

info@MurkWorks.com

Table of Contents

Introduction

Welcome to MurkWorks FTP Daemon NLM for Netware 386. This product provides an efficient and useful FTP daemon NLM for your Netware 386 server.

FTP (File Transfer Protocol) is described in RFC959. It provides a mechanism for internet hosts to transfer files in text or binary mode between cooperating systems. FTPD V1.31 supports the most common commands listed in RFC959. For a complete list of supported commands see appendix B.

This manual assumes that you have a basic understanding of the TCP/IP protocol and a thorough understanding of the NW386 environment. If you do not currently have the TCPIP.NLM module loaded on your Netware server, now would be a good time to review the *TCP/IP Transport Supervisor's Guide* provided with the NW386 3.11 manual set.

Features

FTPD.NLM supports the following features:

- Multiple simultaneous service connections
- Real-time monitor display
- Transaction logging
- Access control by remote IP address, target server and user
- Supports NW2.15 servers through 'proxy' ftp service
- Anonymous connections
- Idle and connect timeouts, restricted access times, etc

System Requirements

NLMS

The FTPD NLM uses both CLIB.NLM and the TCPIP.NLM. When loading the TCPIP.NLM, CLIB will be automatically loaded. Follow the procedures outlined in the *TCP/IP Transport Supervisor's Guide* when installing and configuring the TCPIP nlm. You will need to provide at least one IP address for your Netware server.

MEMORY

A typical Netware 386 server with 4 megabytes of RAM is quite sufficient to operate the FTPD NLM. A detailed analysis of the memory required per connection is described in the **Performance** section of this manual.

Installation

Quick Start

If you're impatient and don't want to read through this manual, follow the instructions in this section to get the NLM up and running as quickly as possible. Without a configuration file, the FTPD NLM will not provide any access control beyond standard Novell password checking. This may be sufficient for your needs. You can always follow the full installation instructions later if you so desire.

This program is distributed with the file:

ftpd.nlm - The FTPD NLM

If you have licensed your copy of the FTPD NLM, you will have also received the file:

ftpd.key - Security key file

To install the FTPD NLM on your Netware 386 server, log into the server as *supervisor* and copy the **ftpd.nlm** file to **sys:system**. If you have licensed your copy, you should also copy the file **ftpd.key** to sys:system.

If you're in a hurry, load the FTPD NLM by issuing the follow command on the NW386 console:

load ftpd

otherwise, create a configuration file as described in the **Configuration File** section before loading the NLM with the load command.

Standard Installation

If you wish to make use of the additional security features of the FTPD NLM, you will have to create a configuration file. The next section of this manual describes how to create a configuration file.

After creating the configuration file, follow the instructions described in the **Quick Start** section shown above.

Configuration File

The file **ftpd.cfg** is provided in the distribution as an example only. Do not copy this file into the sys:system directory. Instead, you should print this file on a printer and use it as a reference when creating your own ftpd.cfg file. This sample file is also listed in the appendix of this manual.

The configuration file uses a simple ASCII format with one command per line. Lines which begin with the pound symbol (#) are comment lines and are ignored.

Use a suitable text editor to create a file called **ftpd.cfg** in the **sys:system** directory. This file will control how the FTPD NLM operates. Read this section carefully to get the most out of your software.

If you create a new **ftpd.cfg** file or alter an existing one, you will have to either reload the FTPD NLM or issue the *ftpd reconfig* command before the changes will take effect.

Class Definitions

When a remote client connects to the FTPD NLM for service, its internet address (IP Address) or internet name can be used to place it in an access class. This allows connections to be grouped based on whether they are ‘local’ or ‘remote’. For example, a university may wish to have one access class defined for on campus clients, and a second one defined for off campus users.

Classes can have a list of valid servers and users, as well as default settings for transaction logging, time limits and so forth.

A new class is defined with the class command, as follows

class *classname*

where *classname* is the name of the class. This name may be anything meaningful to the supervisor. There is one special class whose name is **default**. This class is used whenever an incoming connection does not match any other listed class.

Addresses

Following the **class** command is a list of access control commands and the class definition command: **address**. The **address** command controls which remote connections are grouped into the specified class. The address command takes one argument, an IP Address or IP Name:

address *129.134.*.**
address **.murkworks.com*

The **address** command must follow a **class** command, and may appear more than once in a given class. The asterix (*) indicates a wild-card pattern, which means any value may occupy its space. The **address** command is not allowed in the default class.

When a client connects to the FTPD NLM, the class list is searched for a match. The first class found which has an address pattern that matches the client's IP address or name is used for access control.

Novell Netware 386 keeps its IP Name to IP Address mapping information in the file **sys:etc/hosts** This file may not contain all internet names and addresses. Therefore it is important to specify an IP address in addition to the IP name. If the client's IP name is not listed in the **hosts** file, the FTPD NLM will use the client's IP address to match against.

Access Settings

Each class may have default access control settings. The **settings** command determines how clients may access the Netware server. The **settings** command has its own sub-commands, which generally appear as follows:

settings [sub-commands]

Where [sub-commands] may be one or more of the following:

<u>Command</u>	<u>Argument</u>
log	*count*
logfile	*log_file_name*
connect	*connect_time_in_minutes*
idle	*idle_timeout_in_minutes*
timerange	*allowed_access_timerange*
readonly	

All sub-commands and their arguments should appear on the same line as the **settings** command, however settings can be continued on the next line by inserting a plus symbol (+) at the end of each continued line. The plus symbol may not appear between sub-commands and their arguments. See the sample configuration file for an example of setting continuations.

The **log** *count* setting indicates that client transactions for this class should be logged to the logfile. The number of transactions to be logged is not to exceed *count* entries. When this setting is in effect, transactions such as login, logout, read, write, delete, rmdir and mkdir are recorded in the log file **sys:system\ftpd.log**. If you only want a record of login and logout times, use a log count of zero (0).

Example: **log 0**

The **logfile** setting specifies the name of the file in which log entries should be stored for this class. This setting over-rides the system-wide setting *logfile* (see below) if specified. The file name must be a fully specified name which includes the volume name.

Example: **logfile vol1:usr/anon/logfile.log**

The **connect** *connect_time_limit* command specifies the maximum connect time for any client in this class. After the time limit expires, the client will receive an error message and be automatically disconnected by the server at the completion of any pending command. The time limit is specified in minutes. The limited demo version of the NLM enforces a maximum connect time of five minutes.

The **idle** *idle_timeout_limit* specifies the maximum idle time allowed for any client in this class. If the client does not issue a command within the idle time limit, the client will receive an error message and be automatically disconnected. This command is useful in eliminating ‘dead’ connections due to failed internet connectivity. However, too short of an idle timeout may cause unintended service disruption to the client. The time limit is specified in minutes.

The **timerange** *allowed_access_timerange* controls when clients within this class may access the Novell Netware server. The format of *allowed_access_timerange* is:

timerange *startday-endday/starthour-endhour*

For example:

timerange mon-fri/8-17

In the above example, clients are allowed access monday through friday from 8am to 5pm.

The **timerange** command may be specified multiple times if desired.

The **readonly** command specifies that all clients within this class have readonly access to the Novell Netware server. Even if a particular user has trustee rights which grant him write access, this setting disables all commands which may alter data on the server.

Deny

The **deny** command lists those servers and/or users who should be denied access within this class. The format of the command is:

deny [*server/*]*user*

Where *server/* is an optional server name. If the server is not specified, it applies to the server on which the FTPD NLM is operating. Both *server* and *user* may be the wildcard (*), meaning all servers or all users respectively.

For example, suppose you had defined a class for all local clients. That class would have no restrictions of any kind. You may then wish to add access restrictions for the *default* class which would apply to any non-local client. If you don't want to allow any non-local clients to have access to any of your Novell Servers, you could issue the following two commands in the configuration file:

```
class   default
        deny        */*
```

Allow

The **allow** command lists those servers and/or users which should be allowed access within this class. This command over-rides any previous **deny** command. The format of the command is:

allow [*server/*]*user* optional settings

Where *server/* is an optional server name. If the server is not specified, it applies to the server on which the FTPD NLM is operating. Both *server* and *user* may be the wildcard (*), meaning all servers or all users respectively.

The optional settings argument is a list of **settings** sub-commands which should be applied to this *server/user*. If any optional **settings** sub-commands are listed, then all **settings** for the class are ignored for this user. Therefore any settings which apply to the class, which should also apply to the *server/user* must be repeated on this line.

For example, suppose a particular class had a **setting** *timerange* of mon-fri/8-17 If you wanted to grant readonly access for a particular user but maintain the timerange, the class definition would look something like the following:

```
class       example
    settings    timerange mon-fri/8-17
    allow       FS1/guest    readonly, timerange mon-fri/8-17
```

If the *timerange* command were not repeated on the allow line, then the user FS1/guest would not have any timerange limit because any setting sub-command on the allow command line overrides all setting sub-commands for the class.

Deny/Allow Ordering

The order of the **deny** and **allow** commands within the class is important. All class definitions begin with an implied **allow */***. Access checking follows the list of deny/allow commands in the order in which they are encountered within the configuration file. Deny commands mask out access, where allow commands add in access. The first deny command which explicitely matches the target userid terminates the search. Wildcard deny's do not terminate the search, instead subsequent wildcard allows or explicit allows may over-ride the previously encountered deny command.

When the last deny/allow command is encountered in the class, the logical result determines whether or not access is allowed and which settings are applied.

For example, if you wanted to grant all users access to a particular server with readonly access, but you wanted the supervisor to have full access, the following would be a possible configuration file entry.

allow	*/*	readonly
allow	*/supervisor	

If the order of the above commands were reversed, the supervisor would have readonly access because the */* mask also applies to the supervisor.

MaxConnections

This command specifies the maximum number of simultaneous client connections. For the limited timed demo version of this program, the maximum number of connections is silently forced to two (2) or less. In the licensed version, the maximum number of connections is thirty-two (32).

If this command is not specified, the default number of connections is (2) in the limited timed demo version, and (5) in the licensed version.

Example:

maxconnections 3

ScreenDelay

This command specifies the number of seconds between monitor screen updates. If this command is not specified, the default is (2) seconds between screen refreshes. The minimum allowable value is (1). Too low a value may increase server utilization if there are many active connections. A value between (2) and (5) is recommended.

Example:

screendelay **5**

LogFile

This command specifies the name of the system-wide log file. The system-wide logfile is where transactions will be logged for those users or classes for which no logfile command has been specified. The logfile name must be the full-path name of the log file. If this command is not specified, the default is **sys:system\ftpd.log**

Example:

logfile **sys:system\access.log**

Operation

Command Line Options

The FTPD NLM recognizes several command line arguments which can be provided during the load phase.

/config	*config_file*	Specifies the path to an alternate configuration file, instead of the default **sys:system\ftpd.cfg**
/keyfile	*key_file*	Specifies the path to an alternate key file instead of the default **sys:system\ftpd.key**
/delay	*delay_time*	Specifies the monitor screen update delay in seconds. Overrides any value specified in the configuration file.
/max	*count*	Specifies the maximum number of connections. Overrides any value specified in the configuration file.
/display		Enables the monitor display. By default the monitor display is not shown. Using this switch causes the display to appear when the NLM is loaded.

Example:

load ftpd /display /delay 5 /max 3

This command line loads the FTPD NLM, enables the monitor display with an update frequency of 5 seconds and a maximum of 3 connections.

Console Commands

The FTPD NLM provides an additional set of console commands which can be entered at the NW386 console prompt. All of the following commands are prefixed with the keyword **ftpd**, which indicates that the command is for the FTPD NLM.

ftpd disable	This console command disables new incoming connections. Current connections are not effected.
ftpd enable	This console command enables the FTPD, allowing additional incoming connections.
ftpd serialnum	This console command displays serial number and application information required to register your copy of the FTPD NLM.
ftpd displayon	This console command enables the monitor display.
ftpd displayoff	This console command disables the monitor display.
ftpd delay *delay_time*	This console command sets the monitor display delay time to *delay_time* seconds.
ftpd reconfig *config_file*	This console command causes the NLM to reload the configuration file. If an optional configuration file name is provided after the command, then that file is used instead of the default **sys:system\ftpd.cfg** This command is only allowed when there are no active connections.

Monitor Display

The FTPD NLM provides a near real-time display of current connections. To make use of the display you must enable it either by using the */display* command line option or the *ftpd displayon* console command.

The monitor display shows the first eleven (11) connections. Each connection occupies two lines of the display. The format of the display is as follows:

```
client_host_name                              | last_command  | username
files      | bytes    | Kb/sec   | connect   | idle_time     | last arg
```

client_host_name	Displays the client host name, if known, otherwise it displays the client IP address.

last_command	Displays the last FTP command issued by the client.
username	Displays the SERVER/USERNAME under which the client has logged in. If the username corresponds to a no-password anonymous account, the symbol ***A*** also appears in this field.
files	Displays the count of files transferred by the client during this session.
bytes	Displays the total byte count transferred by the client during this session.
KB/sec	Displays the average KB/Sec for all files transferred by the client.
connect	Displays the total client connect time in HH:MM:SS format.
idle_time	Displays the current client idle time in HH:MM:SS format.
last_arg	Displays the argument for the last FTP protocol command issued by the client.

Home Directories

When a client logs in to a Netware server, the FTPD NLM examines the bindery information for that userid. The NLM looks for a bindery property of the name **HOME_DIR**. If this bindery property is found, it executes an automatic chdir to value of that property. This provides a means to specify a 'home directory' for each user. There are several freeware utilities which provide the tools required to set this value, including David Harris' SETHOME package available from most Novell oriented FTP sites and BBSs.

If the **HOME_DIR** property is not found, the FTPD NLM scans the Trustee Paths for that userid. If any Trustee Path has a trailing directory component which matches the userid (or nearly matches) then that Trustee Path is chosen as a home directory.

For example, given the account *anonymous* with the following Trustee Paths:

```
sys:mail/0023023
sys:usr/anony
```

The FTPD NLM would select the Trustee Path *sys:usr/anony* as the home directory for the *anonymous* userid.

User Commands

The FTPD NLM provides the usual FTP command services. This version offers one additional site specific command:

site tp

This site specific command causes the NLM to list the trustee paths available to the user. A typical unix client can issue this command by using the FTP client *quote* command:

quote site tp

Additionally, the user can return to their **HOME_DIRECTORY** by issuing the command:

cwd ~

The FTPD NLM recognizes the token (~) as meaning the home directory chosen for this user during the login sequence. An error message is returned if no suitable home directory could be found.

Anonymous Accounts

The FTPD NLM fully supports so-called ‘anonymous’ client connections. A typical anonymous client accesses the server with the username of ‘anonymous’, at which point the server prompts the user for an email address or other identifier instead of requesting a password. This allows users to access the server without having to know a special password. The FTPD NLM will record the user supplied address/password to the log file if logging is enabled for the anonymous account.

The FTPD NLM considers **any** account which has no password to be an anonymous account. Therefore the actual account name ‘anonymous’ has no special signifigance to the FTPD NLM.

The following steps are recommended when setting up an anonymous account:

A. Use **syscon** to create a user account named ‘anonymous’
B. Remove the user from all groups (including EVERYONE)
C. Create a station restriction for the account which will inhibit any workstation from logging into the account.
D. Make the password for the account be empty, and do not require a password for the account.
E. Make the account a trustee for a secure directory with [R F] rights only.
F. Remote the account’s access rights to it’s *sys:mail* directory.
G. Use the SETHOME freeware utility to set the home directory for this account to match the secure directory.

When a remote user logs into the server by specifying a userid of ‘anonymous’, they will be prompted for an email address because the FTPD NLM will realize that the account has no password. After the user provides an email address the default current directory will be set to the secure directory.

Proxy Connections

The FTPD NLM provides ‘proxy’ FTP service to older, non-Netware 386 servers.

Security Considerations

The FTPD NLM does not inherently make your Novell server less secure. It does, however, add another avenue of attack into your server. Prudent use of the FTPD.CFG file will ensure that remote users do not violate the integrity of your system.

The FTPD NLM attempts to resist password cracking schemes by disconnecting any connection which enters an incorrect password three times in a row. ie:, after the third incorrectly entered password the user is disconnected and a warning message is written to the server console. Also, on the second and third login attempts (after the first incorrect password) the FTPD NLM delays the login process by three and six seconds respectively in an effort to slow down any password cracking program.

If fifteen incorrect passwords are entered without an intervening correct login (on any server, for any account), then the FTPD NLM also broadcasts a warning message to all users attached to the file server. This serves to alert the supervisor or another responsible party that the file server is under attack from a password cracking program.

Finally, accounts whose password has expired are denied access to the server (whereas the Netware shell would normally allow access and prompt for a new password).

Following are some suggestions to improve security on your server.

A. The NLM environment does not recognize station restrictions on the local server. If you have accounts that have station restrictions, and you do not wish those accounts to be able to access the server via FTP, you must explicitely **deny** those accounts access because the NLM can not recognize station restrictions on the local server.

   Station restrictions are recognized on remote servers. If a remote server specifies a station restriction for an account which should have FTP access, you must add the network address of the local server to the station restriction list of the remote server. You can obtain the station address of the local server by executing the *slist* command from the DOS prompt and noting the address for the local server.

B. Intruder detection lockout may disable an account if excessive invalid passwords are entered. The FTPD NLM attempts to access an account twice for each login, the first time with no password (to see if the account is an 'anonymous' account) and the second time to actually log in the user (if a password was required). On remote servers there is no other way to determine if a password is required for an account.

   Therefore, for each attempted login where an invalid password is specified the NLM actually attempts two logins to the specified account. If the intruder lockout detection count value is set too low, accounts may get locked prematurely.

   Additionally, malicous users may intentionally issue invalid passwords in an attempt to lock an account. This problem is not specific to the FTPD NLM, as any user on a workstation may do the same thing.

   To avoid locking the supervisor's account or other important accounts, you should explicitely **deny** access to those accounts in the *ftpd.cfg* file. Accounts which are ‘denied’ are immediately rejected without a login attempt.

C. Be sure that users granted access via FTP have the appropriate rights to directories on the server. This is especially important with ‘anonymous’ accounts which might accidentally be left in the EVERYONE group. Judicious use of the **readonly** attribute will ensure that data is not lost if password secrecy is compromised.

D. Be careful if your server is reachable from the Internet or other wide-area networks. If you have, until recently, counted on the non-routed aspect of IPX for security, be aware that once connected to the Internet your server becomes reachable from places you’ve probably never heard of. The best course of action when first enabling FTP is to **deny** all users, then expressly **allow** only those accounts which require FTP access. Of those accounts requiring access, grant **readonly** access to those accounts which do not need to write to the file server.

E. You should maintain a logfile of all transactions, or at the very least log all login/logout activity. Examine this logfile on a periodic basis and note accesses from address which seem inappropriate.

F. If you want to totally inhibit all ‘non-local’ ftp connections, create a class for ‘local’ addresses. In the ‘local’ class, configure the **deny/allow** settings as described previously. Then create a default class which **deny**’s all users on all servers.

Example:

```
class default
     deny */*

class local
     addresses *.mydomain.com
     addresses 129.136.*.*
     deny      */*
     allow     bill
     allow     tom
     settings  log 10
```

Performance

This section of the manual describes the memory requirements and expected performance of the FTPD NLM. It describes how we tested and developed the NLM. You do not have to read this section to make use of the FTPD NLM.

How We Test

The FTPD NLM was tested ‘in-house’ on two NW386 file servers, one a 386/33 with 4 megabytes of ram, the other a 486/33 with 8 megabytes of ram. In all cases the client was a 486/33 running Dell Computer Corporation’s OEM version of USL SYSVR4 Unix™.

A series of test ‘scripts’ were used (the *expect* package for Unix implemented the scripts) which exercised all functions within the NLM, the expected return codes and the file transfer operations.

These scripts were executed multiple times, simulating simultaneous clients. The scripts also repeatedly used the mget and mput operation in an effort to place the maximum load on the Netware server.

All scripts were executed with the FTPD NLM running with and without **protect.nlm**[1] Additionally, a select group of beta testers have tested this NLM in several different environments.

Memory Requirements

This version has the following memory requirements (values are approximate). This information was gleaned from **monitor.nlm** under the Resource Utilization section.

```
Base Memory    NLM size       13 K bytes
               Base Socket    9.5 K bytes
Each Client                   13 K bytes additional
Monitor Screen                8 K bytes additional
Each Transaction              32 bytes plus filename size
```

Using the above information, the maximum amount of memory used by the NLM can be determined.

---

[1]Protect.nlm is provided with the Netware C for NLMS kit. This NLM detects memory accesses outside the area set aside for the NLM.

For example, suppose a maximum of two connections was allowed. If every client connection was to be logged with a maximum of 100 transactions each, and if the average filename size was 40 bytes, then the memory usage would be as follows:

22.5 K	Base Memory
26 K	Two client connections
(32 + 40) * 200	Maximum log entries stored in memory
-------------	
64 K bytes	Total memory used

The transaction entries are stored in memory until the client logs out, at which time the client thread gains back its supervisor privileges, allowing it to write to the log file in sys:system.

Processor Utilization

Processor utilization information was obtained by loading **monitor.nlm** with the -p command line option

load monitor -p

The test process used a 5 megabyte PostScript file for transfer. The file was read multiple times, noting the maximum utilization figure ever shown on the monitor display, along with the maximum percentage utilization per process. The Netware server was a 486/33 with 8 megabytes of Ram and an ESDI disk running through an Ultrastor 12F controller.

The test file was transferred in both directions using binary and ascii mode. As expected, ascii transfer had a higher utilization due to the extra processing involved.

The test was executed on a private ethernet connection, the server used a Racal-Interlan NI6510 lan card, the client (SYSVR4 Unix on 486/33) used a Western Digital (SMC) WD8003EBT lan card.

Utilization is proportional to throughput. The FTPD NLM contains no throttling component, therefore it will use maximum server resources during transfers in an effort to supply the client as quickly as possible.

Idle connections cause zero server utilization. However, if the monitor display is enabled and the screen update delay is low (less than 3 seconds) and there are several connections, then the utilization from that component will be approximately 2 - 5 depending on screen update delay.

Operation	Client Throughput	Server Utilization	% Utilization from FTPD	% Utilization from TCP NLM	% Utilization from LAN Card Ints	% Utilization from DISK Ints
Binary Put	330 KB/sec	82	13%	4%	9%	51%
Ascii Put	220 KB/sec	90	48%	3%	6%	30%
Binary Get	400 KB/sec	50	30%	4%	12%	--[2]
Ascii Get	190 KB/sec	63	58%	2%	5%	--

[2]Get operations had negligible disk interrupts because the file had been cached in memory.

How to Get Support

At this point in time MurkWorks is unable to provide telephone support. We are keeping our product costs down by reducing the manpower that would be required to man a telephone. Hopefully this will change in the future.

In the meantime, the **best** method for obtaining support is by sending us electronic mail to our Internet address:

support@murkworks.com

If you are on compuserve, you can send us mail using the address:

internet:support@murkworks.com

If you are not a licensed user, we may not be able to answer questions pertaining to installation problems or configuration issues. If you have found what you consider to be a problem with the FTPD NLM, please feel free to write to us with detailed information about the problem, your environment, and the commands issued which caused the problem. Be sure to include the current version number of the NLM when you write.

You can always write to us:

MurkWorks
P.O. Box 631
Potsdam, NY 13676-0631 USA

Appendix A

Sample Configuration File

```
# Sample FTPD.CFG file for FTPD.NLM
#
# lines beginning with # are comments.

maxconnections 5    # specify max # connections
               # forced to 2 or less in DEMO mode

screendelay    2    # set delay in seconds between status
               # screen updates

logfile        sys:system\logfile.ftp
               # specify alternate logfile

# The following lines are examples. Modify to
# suite your needs and delete any remaining lines
# that do not apply

#####################################################
#                DEFAULT CLASS
#
#   The following class conditions will be used if
#   the incoming IP address (or name) doesn't match
#   any other class
#####################################################
# The following class allows access to all users on
# all servers accept for bkc (on any server). Once
# connected, users may be idle up to 2 minutes, and
# connected for at most 10 minutes. They can only login
# mon-fri and they are only allowed readonly access

class     default
     deny */bkc
     settings  connect 10, idle 2 +
               timerange mon-fri/0-23 +
               readonly
```

```
###########################################################
#    CLASS DOM.MURK.COM
#
# The following class is used if the client IP address
# is in subnet 20
# or any name ending in dom.murk.com
###########################################################
# Access to all servers is denied, accept for access to
# server GIMP,  which allows all users. However users can
# only login mon-fri between 10PM and 5AM, or
# any time saturday and sunday

class     dom_murk_com
     address   *.dom.murk.com
     address   113.121.20.*
     deny */*
     allow     gimp/*
     settings  timerange mon-fri/22-5 +
               timerange sat-sun/0-23

############################################################
#    CLASS MURKWORKS
#
# This class is used if the client IP address is in class
# B network 113.121 or its IP name ends in murk.com
# NOTE: Class dom.murk.com has precidence over this class as
# appropriate since it is more specific
############################################################
# This class allows access to any user on THIS SERVER ONLY
# All users except for bkc have readonly access and can only login
# on weekends between 10am and 9pm
# User bkc has access only on the weekends
# In either case, all read/write operations will be logged to
# the file vol1:usr\fred\logfile.ftp

class     murkworks
     address   113.121.*.*
     address   *.murk.com
     deny */*
     allow     *    readonly timerange sun-sat/10-21
     allow     bkc  timerange sat-sun/0-23
     settings       log 25 +
                logfile vol1:usr\fred\logfile.ftp
```

Appendix B
Supported FTP Commands

The following FTP commands are supported. Be aware that these are protocol level commands and are not necessarily the commands that a user would type at the command line of their client.

```
USER   PORT   RETR   ALLO   PASS   STOR   CWD    XCWD   XPWD
LIST   NLST   HELP   QUIT   MODE   TYPE   STRU   ACCT   NOOP
RMD    MKD    DELE   SITE
```

The site specific command supported in this version is:

```
SITE TP
```

This command returns a list of trustee paths.