ftp4all

COLLABORATORS

	TITLE : ftp4all		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

ftp4all

1.1 FTP4ALL - FTP Server

```
                              FTP4ALL (Amiga Version)
                     Version 2.25

           Amiga Version by FireBall Interactive
               http://www.fireball.csti.ru


             Introduction

             Requirements

               Install

            Configurations

            Users & Groups

               History

              Copyright

               Authors
```

1.2 FTP4ALL - Introduction

What is FTP4ALL ?

FTP4ALL is a FTP server program for UNIX systems. Unlike existing FTP servers, e.g. WUFTPD, FTP4ALL does not require an root account to install it. Every user that can compile source code can install it, thereby providing a facility to share files with friends. All that you need is a C compiler, and I dont know a UNIX system that does not have one.

Features.

- Free source code available, compiles on most UNIX platforms
- Server runs under normal user accounts (no superuser privileges needed)
- Permissions can be given to files and directories, similar to UNIX
- User classes: anonymous / registered / superuser
- IP range for login for every user
- Private mode
- Number of simultaneous logins can be set for each user
- Counts upload and downloaded amount for every user
- Download credit system
- Download counter for every file and directory
- Logs user activities, also through external program
- Customizable FTP replies, with variables
- Online server administration (view/add/edit/remove users/groups)
- Passive FTP
- Archive checking through external program/script
- Top ten lists with variables
- Duplicate file checking
- User scripts, new commands with external scripts/programs
- Intelligent ChDir
- Filename / directory name conversion
- Readme files
- DES encrypted user file
- User information field

Files and Directories.

FTP4ALL uses a file protection scheme similar to that of UNIX. Permissions can be given to three instances: owner, group and world (others). Rights can be assigned to directories and files. For files, the rights are almost the same, with the exception that there is no execute right (but this is not needed for an ftp server). For directories, the rights differ from the UNIX rights. There are not three rights, but seven. These seven rights, that can be given to each of the three instances are:

- access The right to change the permissions of this directory and the files within, and to remove the directory.
- chdir The permission to change to this directory.
- list The permission to list the contents of the directory.
- mkdir The right to create a subdirectory in this directory.
- put The permission to upload a new file to this directory.
- no limit This is not really a right. When you set this flag, then uploads and downloads to/from this directory do not increase the user's upload/download amount.
- get If this is set, created files get read permission.
- overwrite If this is set, created files get write permission.

For files, the permissions are mostly like in UNIX:

- read The permission to read the file - heird from the get right.
- write The right to overwrite, delete or rename the file - heird from the overwrite right.
- chmod The right to change the file permissions - heird from the access right.

Access permissions for a directory and the files within are stored in a special file in the same directory, .permissions by default. This file simply contains user id and group id of the owner and the permissions for owner, group and others - for the directory itself and for each file in the directory. If a readme file exists in a directory, its contents is displayed whenever a user changes to this directory.

Users.

There are four classes of users:

- ~anonymous This user may log in anonymously, i.e. with his e-mail address as password. Login is possible from any IP address. There is a maximum number of anonymous users that can be logged in simultaneously. In addition, anonymous access can be fully disabled by setting this number to zero.
- normal/registered This user has an account with the server, and may log in with his user name and password. There is a maximum number of registered users that can be logged in at the same time. An IP range can be set individually for each user, allowing him to log in from certain IP adresses only.
- superuser This user also needs login and password, but has unlimited access to files and directories of the server, i.e. access permissions do not count. Furthermore, he can use all SITE commands, allowing him to modify users, groups and other server parameters. The maximum number of users does not apply for superusers. However, IP check applies, superusers can only log in from the given IP range (like registered users), thereby improving security.
- template This is not an ordinary user account like the other three ones. It is created in order to create other user accounts faster. When creating a normal user account, a template can be specified. Then all user data from the template are copied to the new user account.

Users are defined in a user file.

Groups.

Like in UNIX, users are moved together in groups. Every user must be in one group. This group is called the primary group. Furthermore, a user can be in up to eight secondary groups. There are two differences between the primary group and the secondary groups. The first is, that all files and directories that the user creates get only the primary group id. Second, the user can only run scripts from his primary group. Groups can be set up in a special group file.

Files and Configuration.

The server itself consists of two binaries, named ftpd and ftps. ftpd is the daemon. Its purpose is to set up the FTP server and to wait for connections. It only terminates by request (shutdown or signal). Every time a client connects, ftpd spawns ftps. ftps then handles the session with the client. ftpd reads a configuration file on startup, named ftpd.conf by default. This file contains the basic configuration, e.g. the port number of the server and the name of the base directory, messages to display and so on. Two more configuration files are also very important: The user and group files. They are named .users and .groups by default and contain user and group definitions. They are saved regulary by ftpd, to reflect changes. ftpd also creates two log files by default: ftpd.err and ftpd.log. All errors go in ftpd.err (are there any ??? ;-)). Whenever the server terminates, date, time and reason are also logged in ftpd.err. Session log goes in ftpd.log, i.e. when did a client connect and from where, uploads, downloads and so on. You can define which actions to log in ftpd.conf.

1.3 FTP4ALL - Requirements

Amiga FTP4ALL require:

1) An Amiga :)
2) TCP/IP Stack (AmiTCP/AmiTCP-GENESiS/Miami/MiamiDeluxe/Inet225)
3) Ixemul/Ixnet.library
4) Internet connections or local network.

1.4 FTP4ALL - Copyright

FTP4ALL is FREE software, and distributed under GNU GENERAL PUBLIC LICENSE.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too. When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things. To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it. For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights. We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software. Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations. Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all. The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you". Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program. You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions: a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change. b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License. c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the

Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.) These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program. In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following: a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or, b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or, c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.) The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable. If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to

do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program. If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances. It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice. This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of

1.5 FTP4ALL - Installation

```
Server guide:

ftpd - reads the configuration file and accepts connections from clients.
       also spawn ftps.
ftps - handles communication with the client.


Installing:

1) Make dir ftp4all in your AmiTCP:serv/ or where you want.
2) Copy ftpd, ftps, ftpd.conf, users, groups to AmiTCP:serv/ftp4all
   or where you make your dir.
3) Change ftpd.conf, users, groups what you need.
Warning! Path like Intenet:Miami/ not accepted, use only /internet/miami/.


Running:
Warning! Running only in standalone mode.

cd amitcp:serv/ftp4all (or cd <where your ftp4all dir>)
stack 200000 (not obligatory, but recommended)
run >NIL: ftpd

(for test you can yourself check it by: telnet 127.0.0.1 21)


Stopping:
```

```
1) break <n> (where <n> - number the ftpd task)
2) quote SITE SHUTDOWN (only by superuser)
```

1.6 FTP4ALL - Configuration

```
                FTPD needs a configuration file to set up the service. All  ↩
                   configuration
options are described below. In the configuration file, all empty lines are
ignored, as well as all lines starting with a "#" character. The
configuration file must be in the current directory. If it is not in the
current directory, run ftpd with the full name of the configuration file as
first parameter (e.g. ftpd ftpd.conf).

alternativeserverprogram
Syntax : alternativeserverprogram <path and file >
Example: alternativeserverprogram /usr/sbin/wu.ftpd
Default: disabled
This  is  to  run two services on one port (Service Splitter).  Must turn option
private  on.   When  a non-authorised connection is encountered, the alternative
service  is  started.   (Versions  lower  than  2.09 would close this connection
immediately) Give alternative service program in configuration file with option:
AlternativeServerProgram <path>
Alternative service is invoked in inetd mode.  This means any serive that is run
by inetd can be used.  Therefore, it is not possible to run two ftp4all services
!   Services  that require root privileges require that you run ftp4all as root.
Or use the following trick:  Example:  Alternative service is wu-ftpd / You need
superuser privileges to run wu-ftpd !!!

  1. Compile & install ftp4all as non-privileged user
  2. Include configuration option for alternative service in ftpd.conf
     (AlternativeServerProgram /usr/sbin/wu.ftpd)
  3. Activate set-user-bit on ftp4all ( chmod u+s ftpd )
  4. become root (superuser)
  5. run ftpd

Note:   ftpd  will  run  in  non-privileged  mode and will only temporary aquire
superuser rights for the following two tasks:
1. to bind the service to a socket (ports <1024 are possible!)
2. to execute the alternative service wu-ftpd
ftps  runs  completely  as  non-privileged  user.   All  files  are  stored with
non-privileged user as owner (as usual).

basedir
Syntax : basedir <path>
Example: basedir /home/ftp/root
Default: /home/ftp
This specifies the base directory.  The base directory appears as root directory
when someone logges into the server.


bouncer
Syntax : bouncer <bouncer_ip>
Example: bouncer 192.168.1.1
Default: (none)
With  this  option  you  can use your server with a so-called "bouncer".  F4A is
```

full working with the rftpd-bouncer. You can get it on the rFtpd-Homepage.

cdpath

```
Syntax : cdpath <path> <uid> <gid> <permissions>
Example: cdpath /cdrom 0 1 0x7F1300
Default: (none)
```

With this option you can assign access permissions to directories that are on read-only filesystems (mostly cd roms). The specified directory and all following subdirectories get the specified access permission. <uid> specifies the user id of the owner, <gid> the group of the owner and <permissions> the access permissions. permissions is an integer and is best specified in hexadecimal form (as shown in the example). There, the first byte gives the rights of the owner, the second the right of the group and the third the rights of other users. The next table shows which bits mean which right. Add the values to form the right:

0x01	-	chdir
0x02	-	list
0x04	-	mkdir
0x08	-	chmod
0x10	-	get
0x20	-	put
0x40	-	overwrite
0x80	-	no limit

You can have as many cdpath statements in the configuration file as you want. The paths are checked by the server in the order in which they appear in the configuration file. In the above example, if you do not want that group members can access /cdrom/private but all other parts of /cdrom should be accessible, then add the line

```
cdpath /cdrom/private 0 1 0x7F0000
```

before

```
cdpath /cdrom 0 1 0x7F1300.
```

This will cause the server to check /cdrom/private first, and as it has no access permissions, access to this directory will be refused. If you change the order, /cdrom is examined first and because /cdrom/private is a subdirectory of /cdrom, it has the same permissions and access is granted.

cdupmsg

```
Syntax : cdupmsg
 <message>
 cdupmsg
Syntax : cdupmsgfile <filename>
Example: cdupmsg
 CDUP command successful.
 Directory changed to: %s
 cdupmsg
Example: cdupmsgfile /home/ftp/msgs/cdup.msg
Default: CDUP command successful.
```

This is the first option of many that can be used to customize FTP replies. This message is sent when a successful CDUP command was performed. The message itself can go over multiple lines and can contain variables. Variables begin with a % and are replaced by their values when the message is sent.
You can either write the message in the configuration file or to a message file. In the first case, the message must be included between the name of the message, as shown in the example. In the latter case, the file that contains the message

must be specified. The syntax is the name of the message followed by file and then the filename (with full path) as shown above. The advantage of this method is, that the messages can be easily changed while the server is running. Whenever the server detects that the message file has been modified, it reads it again.

chdirmsg
Syntax : chdirmsg
<message>
chdirmsg
Syntax : chdirmsgfile <filename>
Default: CWD command successful.
This is the message that is sent when the user changes the directory with the CD command. For more details see cdupmsg.

check
Syntax : check <extension> <program> [<parameter>]
Example: check .zip unzip -t
Default: (none)
With this option, the server checks uploaded archives to see if they are ok. This is done by external programs. You have to give the extension of the file and the program that can check files with this extension. parameter gives the parameter that makes the program check the archive. If an archive is corrupt, the upload amount of the uploader does not increase.

checkdupe
Syntax : check <extension> <program> [<parameter>]
Example: checkdupe .zip /home/ftpd/scripts/dupecheck/dupechk
Default: (none)
This is the interface for an external dupechecker. The syntax is the same as in the check-configuration option. Depending on the exitcode of the external dupecheck program the daemon denies or allows the upload.

checktimeout
Syntax : checktimeout <seconds>
Example: checktimeout 10
Default: 0 (disabled)
This sets a maximum time, which the uploadchecker can use before timeout and
gives a error-message to the log and the user that the checker failed. The
files remain on the server but the siteop have to check them manually.

delemsg
Syntax : delemsg
<message>
delemsg
Example: delemsg
Please force the siteop to update the dupelist.
DELE command successful.
delemsg
Default: DELE command successful.
For example you can specify a script here which updates a dupelist.

dirshortcut
Syntax : dirshortcut <shortcut> <target>
Example: dirshortcut td /Incoming/today
With this Option, you can configure shortcuts to often used directories. No matter in wich directory on your server the user is, with a cd <shortcut> he can

comfortable jump to often used locations.

downloadmsg
Syntax : downloadmsg
<message>
downloadmsg
Syntax : downloadmsgfile <filename>
Default: Transfer completed.
This is the message that is sent after a successful file download. For more details see cdupmsg.

dupecheck
Syntax : dupecheck <wildcard>
Example: dupecheck *.zip
Default: (none)
This option prevents files from being uploaded twice. Files matching the wildcard cannot be uploaded when a file with the same filename exists somewhere on the server. You can specify more than one wildcard by entering this option several times with different wildcards.

dupepath
Syntax : dupepath <directory>
Example: dupepath /upload
Default: /
This options specifies a path from that the dupe checking should be performed. Dupe checking is only performed for uploads in directories below the specified one. Only one directory can be given.

errlogfile
Syntax : errlogfile <filename>
Example: errlogfile /home/ftp/ftpd.err
Default: ftpd.err
ftpd stores runtime error information in this file. Specify a full filename, because otherwise the file is created in the current directory. The default value is therefore not useful.

filenameconversion
Syntax : filenameconversion <n>
Example: filenameconversion 0x11
Default: 0
This option sets a filename conversion function, that is called every time a file is uploaded or directory is created. If one of the events occurr, the name of the file or directory is converted in a special way. The number gives says how the name should be converted. Add values for files and directories:

```
0x00 - no conversion
0x01 - convert file name to lowercase
0x02 - convert file name to uppercase
0x03 - convert first letter of filename to uppercase
0x04 - convert all spaces of filename into underscores _
0x10 - convert directory name to lowercase
0x20 - convert directory name to uppercase
0x30 - first letter of directory name to uppercase
0x40 - convert all spaces of directory name to underscores _
```

goodbyemsg
Syntax : goodbyemsg

```
<Message>
goodbyemsg
Example: goodbyemsg
Goodbye, %s !
You downloaded %d in this session (%D total).
You uploaded %p in this session (%P total).
Your upload/download ratio is %r (%R total).
See you later ...
goodbyemsg
Default: Goodbye !
This  message  is shown when a user logges out.  There can be special characters
in the message.  They will be expanded when the message is shown.

group
Syntax : group <name>:<gid>
Example: group root:0
Default: (no default value)
This  is  used to define a user group.  <name> gives the name of the group.  The
maximum  name length is 10 characters.  However, in directory listings, only the
first  eight  characters  are  show,  so  try  to set up group names that have a
maximum  length  of  eight  characters.   <gid> is a numeric value that gives an
unique  group  identifiaction  number.  Creating a group with this option is not
advised.  Use the groupfile command instead.

groupop
Syntax : groupop <rights>
Example: groupop group/list kick user/passwd
Default: (no default value)
This  configuration option sets the rights a user with the groupop status should
have.   A  user with the groupop status can apply all the rights you've set with
this option to all users in his group.  The rights can be one of the following:

operator/edituser rights
-----------------------------------------------------------
enable             - enable user account
disable            - disable user account
group/all          - all group rights (following)
group/add          - add group
group/change       - change group
group/remove       - remove group
group/list         - list all groups
groups             - list/change groups of a user
kick               - kick user
max                - limit maximal user count
user/all           - all user rights (following)
user/add           - add users to the group
user/passwd        - set user (new) password
user/change/all    - all user change rights (following)
user/change/ip     - change user ip(s)
user/change/stats  - change user statistics
user/change/ratio  - change user ratio
user/change/other  - change other user rights (rest)
user/list          - list all users
user/remove        - remove users from site
user/reset         - reset user account transfer statistics

groupfile
```

Syntax : groupfile <filename>
Example: groupfile /home/ftp/groups
Default: .groups
File to read more user groups from. Specify a full filename, because the file is saved by the server at certain occasions.

ident
Syntax : ident [on|off]
Example: ident on
Default: off
This option enables the built in ident-support. If it is enabeled, the server sends on an connection attempt an ident request to the clients host. The response of this request is logged into the logfile, also connections get refused if the server is in private mode and there is no user with a suitable ident@ip added in the userbase.

listmsg
Syntax : listmsg
<message>
listmsg
Syntax : listmsgfile <filename>
Default: Directory listing completed.
This is the message that is sent after a successful directory listing. For more details see cdupmsg.

log
Syntax : log {[!]<action>}*
Example: log login logout get put
Example: log all !list !cd
Default: none
Log certain user activities. An exclamation mark (!) before the action means that this action is not logged. Not all actions can be logged. By now, only actions regarding Login/Logout and Files/Directories can be logged:

```
all    - log everything
cd     - log change of current directory
chmod  - log chmod commands
del    - log removal of files
get    - log file downloads
login  - log user logins and login retries
logout - log user logout
ls     - log directory listings
md     - log make directory commands
put    - log uploads
rd     - log removal of directories
ren    - log rename of files
```

logfile
Syntax : logfile <filename>
Example: logfile /home/ftp/ftpd.log
Default: ftpd.log
File to log user activities.

logprogram
Syntax : logprogram <program> [<parameters>]
Example: logprogram /home/ftp/server/logger /home/ftp/server/ftpd.log
Default: (none)

This specifies a program that should perform the task of a log server. The program is started by FTP4ALL when the server starts. It receives all log messages that go into the log file. The log program must read the messages over stdin and terminate when stdin is closed. It can then perform own functions, e.g. create statistics out of the log information.

loginkick
Syntax : loginkick <kill signal>
Example: loginkick 9
With this option you can enable the "ghost"-kick function for all users. If enabled it allows the user to wipe all their currently logged in clients with logging in as !username (and correct password of course). The kill signal is the signal which is send to the child processes, 9 is secure.

loginretries
Syntax : loginretries <n>
Example: loginretries 3
Default: 10
Specifies the maximum number of login retries. If the number is reached, the control connection to the client is closed.

maxanon
Syntax : maxanon <n>
Example: maxanon 5
Default: 10
Maximum number of anonymous users logged in at the same time.

maxanondomain
Syntax : maxanondomain <n>
Example: maxanondomain 2
Default: 10
Maximum number of anonymous users logged in at the same time from the same domain. maxanonip and maxanondomain are there for preventing an anonymous user from logging in multiple times - and thereby occupying the server. A domain is aaa.bbb.ccc from an IP address aaa.bbb.ccc.ddd. This is not exactly an internet domain, but it does what it is supposed to.

maxanonip
Syntax : maxanonip <n>
Example: maxanonip 1
Default: 10
Maximum number of simultaneous, anonymous logins from an IP.

maxuser
Syntax : maxuser <n>
Example: maxuser 5
Default: 10
Maximum number of registered users logged in at the same time.

mkdirmsg
Syntax : mkdirmsg
<message>
mkdirmsg
Syntax : mkdirmsgfile <filename>
Default: MKDIR command successful.
This is the message that is sent when the user creates a new directory. For more details see cdupmsg.

```
other_ip
Syntax : other_ip <ip>
Example: other_ip 123.123.123.123
Default: none
The  other_ip option will tell the ftp-daemon to which IP it should bind in case
the host has multiple IP’s.

permissionfile
Syntax : permissionfile <filename>
Example: permissionfile perm.$$$
Default: .permissions
File  to  store access permissions in.  Must be a plain filename (without path),
because it is created in every directory.

pidfile
Syntax : pidfile lt;filename>
Example: pidfile /home/ftpd/bin/ftp4all.pid
Default: /var/run/ftpd.pid
The processid (pid) of the serverprocess gets saved in there on daemon start

port
Syntax : port <n>
Example: port 2048
Default: 21
port to accept connections from.

pre_download
Syntax : pre_download <script>
Example: pre_download /home/ftpd/dl_check
Default: (none)
This script gets executed before every download and the result of the script
(better the exitcode) allows or denys the download. If the script returns
a line of text  is is shown as reason for the user otherwise "no special
reason" is shown. The intention is, to allow download i.e. at given times or
don’t allow more than 20 downloads at one time.

pre_upload
Syntax : pre_upload <script>
Example: pre_upload /home/ftpd/ul_check
Default: (none)
This script gets executed before every upload and the result of the script
(better the exitcode) allows or denys the upload. If the script returns
a line of text  is is shown as reason for the user otherwise "no special
reason" is shown. The intention is, to allow uploads i.e. at given times
or don’t allow more than 20 uploads at one time.

private
Syntax : private [on|off]
Example: private on
Default: off
This  option  is  used  to  make  the  server  real private.  When it is on, all
connections  from  hosts  that  are  not in the ip range of all users are closed
immediately.  When you activate this option, make sure that there is no ip range
*.*.*.*  with  a  user.   Furthermore,  anonymous  access  is no longer possible
because of the ip limitation.  But you must remove the anonymous user or disable
the ip range of this user yourself.
```

programname
Syntax : programname <name>
Example: programname ftpd-server
Default: <serverprogram>
name of program, as shown by "ps"

readmefile
Syntax : readmefile <filename>
Example: readmefile readme.$$$
Default: (no default value)
File to display when directory is changed. This option has no default value. This means that readme files are disabled.

resolver
Syntax : resolver [on|off]
Example: resolver off
Default: on
You can turn off dns-lookups in the log and within the "site user lista".

scriptpath
Syntax : scriptpath <path>
Example: scriptpath /home/ftp/scripts
Default: (none)
Users can now run scripts/programs. The output of the script is shown as result. To run scripts, the script path must be set up with this option. In this directory, you can create directories for every group. The directories must have the name of the groups. Now, users can run scripts which are either in the directory of their group, or in the parent directory. The group directory is searched second for a script. The scripts are run with the SITE RUN command.

serverdir
Syntax : serverdir <directory>
Example: serverdir /home/ftp/server/
Default: (none)
This specifies a path, under which all server files go. If you specify this, then you dont have to specify full paths for message files, log files, user and group files... The serverdir must end with a /

serverprogram
Syntax : serverprogram <filename>
Example: serverprogram /home/ftp/ftps
Default: ftps
Name of program to serve client. Must be absolute or relative from <base_dir> Default value is "ftps", that means that "ftps" must be in <base_dir>

shortcut
Syntax : shortcut <command> [<command>] ...
Example: shortcut credit rules help
Default: (none)
This option creates additional SITE commands. Its name is shortcut, because it is only a shortcut to a user script. Normally, a user script is run with SITE RUN <name>. If you include <name> in the shortcut list, the script can be run with SITE <name>. This means you have a new SITE command. Existing SITE commands can be overriden with this function. For example, if you create a shortcut help, this replaces the built-in help function. The built-in function

can still be accessed by typing SITE two times, e.g. SITE SITE HELP.

startupmsg
Syntax : startupmsg
<message>
startupmsg
Syntax : startupmsgfile <filename>
Default: FTP4ALL Server {VERSION} ready.
This message appears when a client connects to the server. For more details see cdupmsg.

timeoutanon
Syntax : timeoutanon <time>
Example: timeoutanon 120
Default: 60
Timeout value for anonymous users. After expiration of this period the control connection is closed. This means that the user is being logged out.

timeoutuser
Syntax : timeoutuser <time>
Example: timeoutuser 600
Default: 300
Timeout value for registered users.

toomanyanonmsg
Syntax : toomanyanonmsg
<message>
toomanyanonmsg
Syntax : toomanyanonmsgfile <filename>
Default: Too many anonymous users are logged in at the moment. Try again
later.
This is the message that is sent when too many anonymous users are logged in. For more details see cdupmsg.

toomanyusermsg
Syntax : toomanyusermsg
<message>
toomanyusermsg
Syntax : toomanyusermsgfile <filename>
Default: Too many users are logged in at the moment. Try again later. This is the message that is sent when too many users are logged in. For more details see cdupmsg.

touch
Syntax : touch [on|off]
Example: touch on
Default: off
When this option is on, the file modification date of a file being opened is updated to the current time. This time will occurr in directory listings, so you can see when a file was last accessed.

umask
Syntax : umask <mask>
Example: umask 022
Default: 077
umask for server daemon.

umaskanon
Syntax : umaskanon <mask>
Example: umaskanon 0x7F0000
Default: 0x230000
Sets the permissions a directory receives when created by an anonymous user. The first byte gives the rights of the owner, the second the right of the group and the third the rights of other users. The next table shows which bits mean which right. Add the values to form the permission:

0x01 - chdir
0x02 - list
0x04 - mkdir
0x08 - chmod
0x10 - get - files uploaded by user get read permission
0x20 - put
0x40 - overwrite - files uploaded by user get write permission
0x80 - no limit

The get and overwrite rights are no directory rights. They are for files only. This means that the right to download (get) and to upload to an exising file (overwrite) are determined by the file permissions and not by the directory permissions. The only exception are directories that are set in the cdpath. There are no separate file permissions, so the directory permissions apply.

umaskroot
Syntax : umaskroot <mask>
Example: umaskroot 0x7F1313
Default: 0x7F1300
Sets the permissions a directory receives when created by a superuser.

umaskuser
Syntax : umaskuser <mask>
Example: umaskuser 0x7F1313
Default: 0x7F1300
Sets the permissions a directory receives when created by a normal user.

uploadmsg
Syntax : uploadmsg
<message>
uploadmsg
Syntax : uploadmsgfile <filename>
Default: Transfer completed.
This message is sent after a successful file upload. For more details see cdupmsg.

user
Syntax : user <name>:<uid>:<gid>:<passwd>: <kb_down>:<kb_up>:<last_login>:
<ip_range>:<limit>:<info>:<class>: <credit>[:<sec_groups>]
Example: user testuser:20:1:ZhQ64.i/uda-l.K: 20480:10240:823722826:
221.23.143-144.*:100:Test:0x11:1024: 100
Default: (no default value)
User definition, No defaults. Special user "anonymous" or "ftp" enables anonymous access. Use uid=0 for superuser (root). Creating a user with this option is not advised. Use the userfile command instead.

useredit
Syntax : useredit <rights>

Example: useredit group/list kick user/passwd
Default: (no default value)
Nearly the same as the groupop option. This option is intended for site co-operators, to give them rights to edit users, while the groupop option is for an operator of a user group (for example a development group which has its own ftp-group and its on directory on the server). The rights are the same as the groupop rights.

userfile
Syntax : userfile <filename>
Example: userfile /home/ftp/users
Default: .users
File to read more user definitions from. Use this command instead of "user". The user file is saved when the server is closing down or the last user logges out, to preserve the changes in the user area. Therefore, specify a full filename. When the file does not exist at server startup, an error message occurs and the server ist not started. Therefore make sure that at least the file has one user (root) in it.

userkey
Syntax : userkey <key>
Example: userkey X-dwe832kIa
Default: (none)
When you set this option, the user file is saved DES encrypted. This is useful when you want to keep the user file confidental, and you are not root on the system. The placement of this option is important. If it is placed after the userfile option, then the user file is read as plain text. If it is placed before, then the server considers the user file encrypted and reads decrypts it. Remember to move the configuration file to a safe place after you started the server. You need to compile FTP4ALL with a special option in order to have DES encryption available.

waitport
Syntax : waitport
This option is used to tell the ftp daemon to wait on startup, if the port (specified with the port option) is already occupied, instead of binding to the port+1.

welcomemsg
Syntax : welcomemsg
<message>
welcomemsg
Syntax : welcomemsgfile <filename>
Default: User %S logged in.
This message appears after a successful login. For more details see cdupmsg.

whoonline_msg_top
Syntax : whoonline_msg_top
<message>
whoonline_msg_top
Syntax : whoonline_msg_top_file <filename>
Default: " !CNR HANDLE GROUP ON-TM AC-TM MUP!H/!CMD N ACTIVITY !H/ (!CLAST ACTIVITY!H)!0"
This is the message that is sent as header on site w / site who

whoonline_msg_body
Syntax : whoonline_msg_body

<message>
whoonline_msg_body
Syntax : whoonline_msg_body_file <filename>
Default: " !G%W[%02li]C!H%W[·]o!D%W[%-9s]N !G%W[%-9s]G %W [!H%H!C:!H%M!C]T %W[!H%M!C:!H%S!C]t!0 !D%W[%3li]M!H/!D%W[%3li]m %W[!H(!0]l!D%W[%s]A%W[!H)!0]l!0"
This is the body part of the 'SITE WHO' output. Beware if you change this. This is not Goofproof !!! Any Errors in this configuration can crash your Server! Never use any empty line in this configuration. If you need a free line, use a '!0' (All Colors off control cookie). Here are the special Site-Who-Cookies that you can use for your Custom Site-Who:

- %W[intformat]C
  Counter for the number of logged in users. Intformat has the same syntax as the printf-command in ansi-c. As example '%W[%02li]C' means to print the counter always 2 characters wide and do not ommit the leading zero.
- %W[stringformat]N
  The cookie for the username. stringformat has the same syntax as used for printing a string in ansi-c's printf-command. As example %W[%-9s]N' means to print the name 9 characters wide and use spaces left to the name if the name has less than 9 characters.
- %W[stringformat]G
  The cookie for the Groupname the user is in. syntax is like in the cookie for the username.
- %W[timeformat]T
  The Onlinetime of the user. Syntax of timeformat is like in ansi-c's stroftime command. As example '%W[%H:%M]T' would give back a 5 character wide timestring with hours and minutes.
- %W[timeformat]t
  The Activitytime of the user. Syntax like in Onlinetime. It tells you for how long the user is doing the current action, or how long he is idle
- %W[longintformat]M
  Uploadamount in MB for this session. As example %W[%3li]M gives you a 3 character wide output.
- %W[longintformat]m
  Downloadamount in MB for this session. As example %W[%3li]M gives you a 3 character wide output.
- %W[stringformat]A
  Activity of the user. As example %W[%s]A gives you back the users activity.
- %W[stringformat]l
  Idletag. A one character wide string that is displayed if the user is idle. If the user is not idle, a space-character is inserted at this position. Example: %W[X]l . An X would mark the user as idle.
- %W[stringformat]L
  Users Infoline. Makes not so much sense right now, the built in infoline is not very flexible. If you want to use it - same syntax as to the users activity applies.
- %W[stringformat]I
  Users current ident. Same syntax as for Activity.
- %W[stringformat]i
  Users Hostname/IP. Syntax as in the other cookies that use stringformat.
- %W[stringformat]o
  Own-login tag. Tags the entry of the user that is calling site who. Format is a single character like in the Idletag.
- %W[stringformat]H

```
     ident@hostname.  Again using the syntax for stringformats.

whoonline_msg_tail
Syntax : whoonline_msg_tail
<message>
whoonline_msg_tail
Syntax : whoonline_msg_tail_file <filename>
Default: " !HFTP!B4!HALL v2.20 !CHH!c:!CMM MM!c: !CSS !HON-TM=ONLINE TIME /
AC-TM=ACTIVITY TIME!0"
This is the message that is sent as trailer on site w / site who


See also
                  User & Group Files
```

1.7 FTP4ALL - History Page

```
Version 1.0 (Beginning of August 1996)

   o This is the original version
   o Done a lot of testing and bux-fixing
   o Added SIZE command
   o Added MDTM command
   o Added passive FTP

Version 1.1

   o Released 3 patch versions
   o Added IP checking
   o Added private mode
   o Added touch feature
   o Added SITE KICK command
   o Added recursive directory listings

Version 1.2

   o Added download limit system
   o Improved SITE USER CHANGE command

Version 1.3 (24th September 1996)

   o Changed upload and download amount handling
   o Completely rebuild log system
   o Added SITE EXEC command

Version 1.4 (30th September 1996)

   o Added read-only filesystem (CDROM) support
   o Added archive checking
   o Added server reconfiguration command
   o Added rankings

Version 1.4x (October 1996)
```

```
   o Directory listing function rewritten, listings are not compatible with
     UNIX ls command
   o You can now assign up to four IP numbers to a user
   o Added new field "user information", this can be displayed in the top
     tenlists
   o Fixed some Linux bugs that caused the server to crash when connected
   o Improved signal handling under Linux
   o Added SITE W command
   o Fixed bug that caused the server to crash on ls commands on certain
     systems

Version 1.5 (November 1996)

   o Added file searching command (SITE FF)
   o Added dupe checking

Version 1.6 beta (December 1996)

   o Supports more systems (now: SunOS 4.x & 5.x / AIX / Linux / HP-UX /
     generic SVR4 / generic BSD)
   o File permissions added
   o Download counter added
   o User scripts added

Version 2.0 (05th January 1997)

   o External log program
   o More custom messages added
   o user classes changed / multiple login / templates / multiple group
     membership
   o SITE USER functions improved
   o Credit system changed
   o "Intelligent" ChDir

Version 2.01 (14th January 1997)

   o Fixed login bug that occured on all systems except my Linux :-)

Version 2.02 (10th February 1997)

   o Anonymous users do not get credit any more
   o Added message variable that runs scripts (%#!)

Version 2.05

   o Fixed cd /.. bug
   o Changed command line parsing, spaces in filenames are now possible,
     conforming to WuFTPd
   o Added MaxAnonIP and MaxAnonDomain limits

Version 2.06 (06/May/1997)

   o Improved configure script
   o Added SITE CHMODR command
   o Added ServerDir configuration option

Version 2.07 (15/May/1997)
```

```
   o Bugfixes

Version 2.15 (November 97)

   o Several bugs fixed in v2.15. Works fine.


Version 2.16 (03/December/97)

   o Fixed a few bugs, which causes the server to crash under
     several OS'es. (e.g. AIX).

Version 2.17 (27/January/98)

   o new options for 'site user change ...': KBUP[+|-]=<num>,
     KBDOWN[+|-]=<num>, MBUP[+|-]=<num> MBDOWN[+|-]=<num>, MBCRED[+|-]=<num>
   o colors in ftp4all ! 'site color' switches between color and monochrome
     mode. A helpscript for colors is under the addon-section of this
     website. You will find the color definition codes also under the
     variables section of the config file.
   o configuration option 'CHECKDUPE' as interface for an external
     dupechecker
   o configuration option 'PIDFILE' - server stores its own process-id there
   o configuration option 'MKDIRMSG' - can be used to give the information
     about new directories to other programs (e.g. irc-bots)
   o configuration options 'TOOMANYUSERMSG' and 'TOOMANYANONMSG' - custom
     messages if the server is full
   o configuration option 'WHOONLINEMSG' - header message for 'site who/site
     w'
   o changed layout for 'site who'/'site w'
   o connections that fail the login now dont get the logoutmsg anymore on
     disconnect, they just get a 'Goodbye.'
   o changed timeformat in the logfile (HH:MM:SS instead of HH:MM)


Version 2.18b1 (11/March/98)

   o Minor update: each user can now have 10 ip's/ip-ranges

Version 2.18b2 (15/March/98)

   o Fixes a bug where connections from an ip that was not in ip0-3 of
     any user were shut down immediately.

Version 2.18b3 (25/March/98)

   o Site run is now disabled for normal users - use shortcuts instead
      (thx to superfly for discovering and fixing this bug)

   o Fixed a small bug that substracted users on downloads in race situations
     too less credits

Version 2.19 (04/Aplir/98)

   o ident support: additional configuration options 'ident on' and 'ident
     off' in ftpd.conf. Default setting is ident off.
   o changed layout for 'site user list': Ident@ip's are only shown if
     information about a single user is requested. Additional information
     about users implemented (color on/off, # of simultaneous logins).
     Format of user's last login got more compact, ratio field is now left
     out (limit filed is still there).
   o changed layout for 'site user lista': idents of the logged in users are
     now also shown.

Version 2.20 (22/May/98)

   o Directory Shortcuts (e.g. 'cd td' gets you to '/Incoming/Today')
   o Configurable 'Site Who' layout
   o Additional Commands for maintaining IP's
   o Added 'SITE USR' as synonym for 'SITE USER'
   o Bugfixes and minor enhancements (see CHANGES inside the package)

Version 2.20b1 (07/June/98)

   o small bug fixed, that allowed users to delete the current permission
     file if they had +a access, thanks to equinox for pointing it out and
     thanks to crestor for fixing it :-)

Version 2.21 (14/June/98)

   o Added other_ip feature. You can now specify a ip to which the server
     will bind. The feature was implemented to bind the server to a certain
     ip when on the host multiple ip's are configured, but it also can be
     used when no name-entry is made for the host (I thought it would be
     helpful for some people).
     When no other_ip is given in the ftpd.conf the server uses the normal
     configuration of ftp4all.
     The changes has been made by our beloved SenfGurke, thanks ! :)

Version 2.21b1 (17/July/98)

   o small uid bug fixed, that allowed the superuser to change a users uid to
     an already existing uid, so that two users with the same uid exist,
     thanks to crestor for detecting and fixing it :)
     ah, and sorry for the wrong version number of the 2.20b1 release, it
     should have been 2.21b1, but now i'll just leave this number. :)

Version 2.22 (11/September/98)

   o New variable: %k is the last speed of the user's upload/download (kb/s).
   o Added site command: COUNT
   o Added extra user options: Invisible users are not showed in SITE WHO nor
     SITE COUNT.
   o Moved Color to Extra parameter (instead of class)
   o Login with !login kills ghosts on the site (if loginkick is set in the
     config).
   o New site command: ADDUSER to add users on one line
   o The default whoonline_msg_tail shows now the actual version
   o Added special user class: groupop and useredit class.
   o Add waitport-option in the config file and ftp4all will wait for the
     port to be closed if it's already open (instead of incrementing the
     port number)
   o Scripts can now read stdin to get subclasses and extra parameters:
     1st line: User name
     2nd: Class (s|n|a)
     3rd: Subclass (g|e|-)
     4th: Extra (ic|i-|-c|--)

Version 2.23  (13/October/98)

   o ftp4all tries now to 'reget' the old port if the daemon is restarted and
     the old port is still in use
   o all gets() got replaced with appropriate fgets() so compilers don't
     complain anymore about the usage of gets()
   o new command: site version - guess what it does
   o The "random disconnect bug" should be fixed now, unfortunately many
     external scripts will not work anymore with 2.23 now. They have to get
     updated to use the new (clean) method of passing userinformation to the
     script
     please read the CHANGES file for details!
   o SITE COLOR works again (was broken in v2.22).
   o logins=0 bug fixed.
   o User and Group are displayed in 9 char columns, thus now being
     compatible with the maximum group- and username-length.
   o a few more small bugs were fixed too, see CHANGES.

Version 2.24 (22/November/98)

   o F4A_EXTRA is splitted into F4A_INVISIBLE and F4A_COLOR as separate
     environment variables
   o backwards-compatibility has been removed
   o You can switch resolver ON or OFF in ftpd.conf to prevent dns-lookups in
     the logs or while doing a "site user lista". Default is on
   o You can switch waitport-option now needs to switched ON or OFF in
     ftpd.conf
   o Support for rFtpd's so-called "bouncer", allowing you to move your
     ftp-server to another host and setting up this bouncer on the old host,
     thus allowing transparent forwarding to the new host.
     You find the syntax at the config-page
     The rFtpd-bouncer can be found on the rFtpd homepage.
     Thanks to reanimator for it, and thanks to Bleachboy for the original
     version of this bouncer
   o Fixed a bug in the password-routine, which prevented FTP4ALL to work on
     FreeBSD systems using the MD5 encryption-libraries
   o Fixed a bug which prevented the binding to an explicitely given IP
     (other_ip) to work on some systems
   o If other_ip is specified, the data connection will now originate from
     this IP-address too
   o The code remains now SysV-compatible, setenv was replaced by putenv
     (special thanks to Garry Glendown for his help)

Version 2.25 (07/Mar/99)

   o Added the pre_upload and pre_download config-switches. The given script
     is not searched in the script-folder and is executed before the given
     transfer. The script has to produce one line of output, the daemon don't
     give any response on aborts. The intention is, to allow download or uploads
     i.e. at given times or don't allow 20 up/downloads at one time. Or to
     specify upload-only users (see new ENV-Variables below). The script has
     as first parameter the uploaded or downloaded filename.
   o Now you can set the limit to -1 and set a credit-value which will never
     increase or decrease. With that account a user can download every file
     smaller than credit-value, but upload all he wants. Thanks to Jan Menzel
     for this idea and patch.
   o The delemsg-feature was implemented. Syntax is in mkdirmsg-style.
     For example you can specify a script there which updates a dupelist.
   o New EXTRA-flag added. With this you can specify, that this user can’t
     change his password. Just add the flag with "site usr change username
     extra+=p".
   o Timeouting uploadchecker-scripts won’t stall the server anymore. In
     ftpd.conf you can from now give the timeoutvalue in seconds with
     "checktimeout <sec>". The server responses with a short message that the
     file was not checked, also in the server-log. The files won’t get
     deleted, the admin have to check them manually. When checktimeout
     is 0 the timeout is disabled (default). Thanks to Mindtrip for his help
     implementing this.
   o Error 24 (Too many files open) included in internal error-list (that it
     shows this message up when it occures).
   o F4A_GROUP is a new environment-variable. It contains the name of the
     group of the calling user. More variables to come. Suggestions ?
   o Slighty better parsing of ftpd.conf, just checks more stuff.
   o Large upload amounts-bug (>2gb) fixed.
   o SGI-chmodallowed-char-to-int changed.
   o The upload permission is now "user:incoming". On ’dele’ of this file, it
     don’t removes credits.
   o Login-and-color-data is now stored in correct and static values. (thx to
     mindtrip :)
   o Now it should compile on hpux too (this error with h_error is removed now).
   o Unused variables removed.
   o Site idle in progress.
   o FAQ-Changes (and corrections by Kafka).
```

1.8 FTP4ALL - Authors Contact

```
Original Version:

Copyright 1996,1997,1998 by Christoph Schwarz

WWW: http://www.ftp4all.de
MAIL: ftp4all@ftp4all.de


Amiga Version:

Copyright 1998 by FireBall Interactive

WWW: http://www.fireball.csti.ru
MAIL: bug@fireball.csti.ru / support@fireball.csti.ru

WWW: http://beta.math.msu.su/~zabrod (port author page)
MAIL: nickolayz@usa.net (port author)
```

1.9 FTP4ALL - Users and Groups Files

User and group files play an important role in the running of the server, beacuse they contain the users and groups. They are plain text files, like in UNIX and have a similar structure (see below). User and group files are read by the server on startup, to initialize the initial group and user list. When the server is running, log in as a superuser and use the special user and group manipulation commands to edit users and groups. Do not modify the user and group files manually while the server is running, because they will be overwritten by the server at certain conditions. These two conditions are:

1. The last user logges out of the server (i.e. a user logges out and there is no other user logged in anymore)

2. The server is closing down, either via the "shutdown" command or by receiving a signal (SIGHUP, SIGTERM ...)

Because the server saves user and group files, specify an absolute filename (including the path, starting with "/") in the configuration file to make sure that the files are saved in the right directory. With this system, you do not have to worry much about user and group files. Just set up the initial files, you can use the examples below. Then start the server. When you shut the server down, the files will be saved and reflect the last changes. When starting the server again, the files are read again, so the server has the latest user and group data.

Example of a user file

```
# 4 user(s)
root:0:0::0:0:0:174.21.2.45,spici@174.21.3.1:0::0x10:0
anonymous:1:1::0:0:0:*.*.*.*:0::0x02:0
user_1:2:2::0:0:0:134.2.*.*:50::0x11:0
user_2:3:2::0:0:0:211.182.45.128-195,
199.2.3.*:20:0x91:0
```

This sets up four users. The first one (root) is a superuser, because his user class (0x10) has the lowest 3 bits zero. He is only allowed to log in from 174.21.2.45 or from host 174.21.3.1. where his username is ’spici’. Replace this with your IP address, to increase security. The second one (anonymous) is an anonymous user. He may log in from any IP address, the IP_range field has no effect for this user. The passwd field has no effect, too. The third and the fourth one (user_1 and user_2) are sample users, who can only log in from the specified IP range.

Example of a group file

```
# 3 group(s)
root:0
anon:1
user:2
```

This sets up three groups. They are for superusers, guest and registered users. The group ids are used in the user file above.

The user file in detail

The user file consists of lines in the following format:

```
<username>:<uid>:<gid>:<passwd>:<kb_down>:
<kb_up>:<last_login>:<ip_range>:<limit>:<info>:
<class>:<credit>[:<gid>[,<gid>]...]
```

There are nine fields in a line, separated by a colon.

- <username> The username has a maximum of 9 characters. Longer names are not possible and cause an error when the server reads the user file. The special name "ftp" or "anonymous" enables anonymous access to the server.

- <uid> The user id (uid) is a numeric value.

- <gid> The third field is the group id (gid). Every user must belong to at least one group. He can be in up to eight secondary groups.

- <passwd> The fourth field is the password of the user. It is crypted with the standary UNIX crypt funtion. When the field is empty, the user can log in with any password. To change a password, use the SITE PASSWD command of the server. Every user can change his own password, but only the superuser can change passwords for other users.

- <kb_down> This field holds the amount of data (in KB) the user has downloaded so far.

- <kb_up> This field holds the amount of data the user has uploaded so far (in KBytes).

- ~<last_login> This field contains date and time of the last login of the user. It is in fact a long integer, giving the number of seconds from January 1st 1970 to the last login time (this is the way a date is stored in UNIX).

- <ip_range> This is the Ident and IP range from which the user is allowed to log in. You can specify up to ten Ident@IP ranges separated by a comma. This field has no effect for anonymous users. Ident can have up to 11 characters. You can also ommit the ident part - then all idents are valid. The four parts of the IP address can countain three kinds of numbers:

  1. a single number (from 0 to 255)
  2. a range in the form <n>-<m>, where <n> and <m> are single numbers and n < m
  3. a "*" character, which is a short form for "0-255"

- <limit> This field gives the user's dowload limit. It is a percent value, and relative to the download amount of the user. If it is 100, the user must upload at least 100% of what he downloaded. If 50, he must upload at least half of the download amount. If it is 0, the user can download without limit.

- <info> This can hold a maximum of fifteen characters. It can be displayed in the Top Lists, so it can hold some short information about the user. The maximum length is 15 characters, spaces are not allowed.

- <class> This field containss three parameters: the user class, the number of simultaneous logins and the account disabled flag. It is always written in hexadecimal. The higher four bits contain the maximum number of simultaneous logins for the user, zero means unlimited. The lower four bits contain the user class and the disabled flag. The user classes are: 0=superuser, 1=normal, 2=anonymous, 3=template. Add 8 if the account is disabled.
  A superuser has all rights in accessing files and directories. He also has special commands available, with which he can modify users and groups. He may also close, reopen and shut down the server, or change the maximum number of users to be logged in and so on.

- <credit> This contains the user's download credit in Kilobytes (KB). It increases when the user uploads a file, and decreases when he downloads one. The variable limit determines how much the credit increases. The credit always decreases by the size of the downloaded file (in KB). If the limit is zero, the credit variable has no effect and doesn't change at all.

- [<secondary_groups>] This field contains a the list of secondary groups, the user is in. It is a list of the group ids, seperated by commas, e.g. 1,2,3. The user can be at most in eight secondary groups.

The group file in detail

The group file consists of lines in the following format:
<groupname>:<gid>
There are just two fields in a line, separated by a colon.

- <groupname> The groupname has a maximum of 9 characters. Longer names cause an error.

- ~<gid> The second field is the group id (gid). This is a non-negative integer. This value can be used to assign a user to a group, if it is entered in the third field of a user definition.