FTP Serv-U
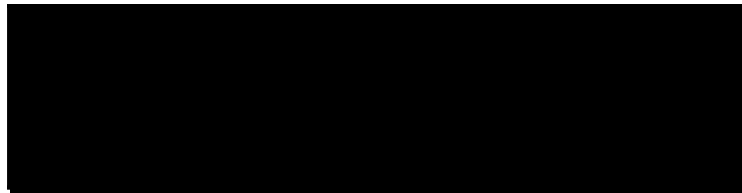
FTP-Server Daemon for WinSock

Version 2.1

Revision 1

November 1996

Cat Soft

Disclaimer

I know, it's not the nicest way to start off. So let's just get this part over with, OK?!

The FTP server program Serv-U and its documentation are copyright Rob Beckers. It is distributed as shareware, giving you the right to try it for a period of 30 days. If you intend to use Serv-U after the initial try-out period, you are obliged to pay the registration fee.

The next paragraph is a beautiful piece of prose. In just two sentences it says it all. Alas, with all the sue-happy people in this world it is unfortunately necessary, so please bear with me.

This software is provided by the regents and contributors 'as is' and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the regents or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

Contents

4. Getting In Touch - Bugs & Registration..

Registration Form Serv-U...

Introduction

Thank you for giving this program a try!

With Serv-U, your PC will be turned into a FTP server. This means that others on the computer network that you are connected to (Internet for most people) can access your PC to copy, move, make, and delete files and directories, using the FTP protocol (FTP = **F**ile **T**ransfer **P**rotocol). This protocol dictates standard ways of communication between computers, so that many different types of computers, using different operating systems and file formats, can exchange files.

Serv-U is a 'server' program, also called a 'daemon'. The term daemon comes from ancient Greek mythology. There, the Daemons were half-gods, acting as messengers between the people on earth and the gods. This FTP server acts, likewise, as a messenger for file transfer between FTP clients and your computer. Once started it sits in the background waiting for a client to contact it and after communications are established, acting out the client's commands.

There are FTP servers (and clients) for many different systems. This particular program is meant for PC's running MS-Windows that have a WinSock version 1.1 compatible TCP/IP stack installed.

Why use this program and not one of the many other FTP servers that are available? For this I have to take you back in time a little, to a little over two years ago. I needed a FTP server to make some files available to others and tried out a number of server programs. One simply didn't work. Another would work, but as soon as someone started transferring a file from my PC it would lock up the whole machine until the transfer was complete. And then there was one that worked fine, but lacked all but the most basic security. So, after endless frustration I decided to write my own, figuring it couldn't be that hard. As usual things got a bit out of hand, but after a year and 11000 lines of C++ there was version 1.0. The current version, v2.1, was made two and a half years later, and consists of over 22000 lines of code. I hope you like the result!

So what has this FTP server to offer?

- Available in 16- and 32-bit versions for Win3.1, WFW3.11, Win95, and NT.
- At the time of this writing there were over 3500 registered users of Serv-U, with many of those for multiple copies!
- Access for multiple clients at the same time. Access for 'Anonymous' users. With the possibility to limit the number of clients at any given time, so your PC remains workable.
- Lots of security! On a directory and even file basis. Allowing different settings for each user, and by putting users into 'groups' permitting easy maintenance for large numbers of users. There's even an option to allow or prohibit clients on the basis of their IP-number. Ideal if you want to let certain people roam around your computer, but you don't want the whole world knocking at your door (that is to say: they can knock, but they won't get in).
- Support for 'multi-homed IP' sites that need a single server for servicing several IP numbers.
- Supports upload/download ratios, and disk quota limitations.

- Lots of other features, like configurable sign-on and sign-off messages, per user configurable login messages, directory change messages, and UNIX-style 'links'.
- Serv-U lets you execute programs remotely though your FTP client.
- It allows transfer to or from ports, like PRN:, LPT1:, COM1:, and AUX:. This allows you to setup your PC as a print server by simply transferring the file to be printed to the desired port! Since the ports are part of the regular security system, a user needs permission to transfer to/from a port, allowing you to control who can use it.
- A quite complete implementation of, and very strict adherence to the FTP standard (found in documents RFC 959 and RFC 1123). Supports the 'passive' command PASV. This is needed by WWW browsers and proxy agents (something required when there is a 'firewall') for FTP transfer. Another feature is that you can let 'Anonymous' users always see the root directory ('/') as their login directory. This, again, is needed by some WWW browsers to make FTP transfers work.
- Support for the 'resume' command.
- Support for external user access verification DLLs.
- Easy to setup and maintain. Everything is accessible through menus, and for automated maintenance the settings are stored in an .INI file of simple format.
- There is lots of logging! You can select how much to log and where (to screen and/or file). The logfile is of a standard format to make machine reading easy, in case you need to do that for accounting purposes.
- It is fast! The file transfer speed you'll get will be very close to the maximum your TCP/IP stack is capable of (well, assuming your FTP client is at least as fast of course).
- Life-time free updates when registered (As long as Serv-U exists).
  - Compared to the commercial implementations, Serv-U is dirt cheap: less than 0.11 › per line of code!

If you didn't register this program, then you're looking at the try-out version of Serv-U. When started for the first time, it'll let you choose between a fully functional program or a somewhat crippled one. The text during startup will explain these options clearly. If you choose the fully functional version then there is absolutely no difference with the registered version. But (yes, there had to be a 'but'), after a little over 30 days it will stop working. Counting starts the first time you run the program. I warn you beforehand: re-installing it will not help you!

And finally: A big 'Thanks' to all the beta testers of Serv-U. The list is now so long it is no longer possible to mention each one individually, but you guys definitely helped to make Serv-U a better product! Any bugs that remain are of course my own fault. Special thanks to Ryan and Kevin, for revising the manual! And, Alun, sorry again! Seems you'll have some catching up to do (The monopoly is now really gone I fear)…

OK, enough sales talk. Let's continue with the actual documentation. First thing will be how to install Serv-U to get you in business.

1. Making It Work - Installation

You're eager to get things going, but a little afraid of what lies ahead. Never fear, it couldn't be simpler. So let's get started!

1.1 Installation

Nothing is simpler than installing Serv-U: just unzip it in the directory of your choice and run it. There is no need to change your 'PATH' or put anything in other directories.

The Serv-U zip-file comes with the following files:
SERV-U16.EXE   - The 16-bit version of the FTP-server executable itself
- and/or -
SERV-U32.EXE   - The 32-bit version of the FTP-server executable
SERV-U.DOC   - The documentation in MS-Word version 6.0 format
SERV-U.TXT   - The documentation in ASCII format
README.TXT   - Something you have to read
REGISTER.TXT   - A registration form in ASCII format
VERSION.TXT   - A list of changes between the various versions.
FILE_ID.DIZ   - Short description file for use by bulletin boards

When Serv-U is started for the first time, it creates the file:
SERV-U.INI   - File containing all settings and user information

The 16-bit version of Serv-U uses Microsoft's 3D-controls to make the dialog boxes look better. If for some reason you get flat-looking dialog boxes it means that the file CTL3DV2.DLL is missing. This file should be in your Window's 'system' directory. It is rare if this file is missing, since just about anything these days uses it. However, if you need a copy then drop me a line. The 32-bit version uses Windows 95 style controls. This means things will look flat on NT 3.51, this is normal. In NT 4.0 and Win95 you will automagically get the 3D-look.

Serv-U offers two very distinct ways to try it out. Each with its own advantages and disadvantages. When started for the first time (or in absence of a SERV-U.INI file), it will show you a screen explaining the two try-out options and allowing you to choose one of them. The first choice is a fully functional try-out version exactly equal to the real thing, but it will stop working after 30 and some days. It does this by contacting a permission server over the Internet every time Serv-U is started. The permission server keeps track of when the program was started for the first time and uses that information to determine if it should be allowed to run again, or not. It then sends this answer back to Serv-U and the program will consequently work or stop. To this effect, a network packet is sent to the permission server containing a version code, the IP number of your PC, ID, and a run/norun flag. It receives back the same packet with the flag set to run or stop. This is all that is communicated between the systems, nothing more, nothing less!

The second choice is a try-out version that is somewhat crippled. It will only allow a total of 10 file transfers (5 PUTs and 5 GETs), it will show a message saying it is not registered to anyone who logs into the server, and finally, it will only stay on-line for one hour. You have to restart it again

after that to get another hour. The advantage is that it will <u>not</u> contact my permission server over the network.

Just to make sure this is clear: **Once Serv-U is registered it will <u>NEVER</u> send anything over the network other than what is needed for regular FTP traffic! In short: The registered program will <u>NOT</u> contact my permission server!**

1.2 Network Installation

To find the setup information, Serv-U first looks in the program directory (i.e. where SERV-U.EXE is located) for the file SERV-U.INI. If no .INI file is found, the program looks for the existence of an environment variable with the name SERV-U. If this variable exists, it should be set to the path for SERV-U.INI. The program will then use the .INI file this path points to. For network use with a single executable shared between many users that need their own settings this is a convenient way to set things up. If this environment variable does not exist, the whole DOS path is searched including the Windows directories. If a file SERV-U.INI is found somewhere it will be used, this way also allowing for a single copy of the executable on a network server while having individual settings for each user. Finally, if SERV-U.INI was still not found, it will be created in the default program directory.

If a SERV-U.INI file is available it does not have to be writable per se. The program will work when it is read-only, but setup changes and the current position of the Serv-U window will not be saved.

1.3 Command Line Options

Serv-U recognizes a number of command line options that may be convenient to tailor it to your particular use. The following is recognized:

        / h   = Hide the server after startup.
        / i   = Start the server as an icon.
        / u   = Stop 'unconditionally'.

Hiding the server makes it disappear, there will be no icon nor window and no way to interact with it. All you can do to stop the server after that is either kill the task in the task manager, or reset the computer. The 'unconditional' option means that if the server is stopped it'll quit without any warning, even if there are still users logged in. This can be convenient if you want to be able to stop the server from a script.

1.4 Quick Setup

When you run Serv-U for the first time there will be no users and access will be restricted. This section will describe the bare necessities to allow access. In addition you can also go through the 'Setup' menu items and put in your heart's desires. The next chapter will explain the various options and how to use them. You might want to take a look at it, since the setup is simple but not totally intuitive (Sorry for that, I tried hard but . . .).

Now, to quickly get Serv-U up & running, just start it. Leave all the settings at their defaults, since in most cases that will be OK. By design Serv-U was made to be 'safe' out of the box: Unless you specify otherwise your PC is well protected by the default settings, giving you one thing less to worry about. Right! Let's have a go at it.

At this point you have a functioning server but no defined users, so no one can log in. To define users, go to the 'Setup - Users' section in the menu and you'll see a blank user setup screen. Now I'll assume you want to setup your server for anonymous access. To make that work, do the following:

- At 'User name', type the name we want to add: '`anonymous`'
- At 'Home directory', type the directory where you want anonymous users to be. Let's assume we want them on '`d:\anonftp`', so enter this (or press 'Browse' and choose your favorite home directory for anonymous users).
- Press the 'Add' button of the 'File/Directory access rules' section.
- Anonymous users need access to their home directory, so enter '`d:\anonftp`' in the dialog box and press 'OK'.
- To hide the rest of the PC from anonymous users we'll make them see '/' as their home directory. To do this select the 'Miscellaneous' button. In the dialog box that pops up check the 'Show path relative to homedir' button and press 'OK'.
  - That's all, just leave the other settings at their default values and press 'Store' to save your setup.

You now have a user with name 'Anonymous' that can log into the server. Of course, if you want anonymous users in a different directory then 'd:\anonftp', then feel free to change this. Just make sure that the directory you specify actually exist and that it is part of the access rules.

If the zip-file you have came with both the 16- and 32-bit version of Serv-U (SERV-U16.EXE and SERV-U32.EXE respectively - the first is for Win3.1 and WFW3.11, the latter for Win95 and NT), and you only need one of them, then you can of course delete the other to free up disk space.

1.5 De-installation

It is hard to imagine why, but if for some reason some day you want to get rid of Serv-U then that is just as easy as was installing it. Just delete the whole directory where you put it, and that's it! Serv-U does <u>not</u> change any system files and does <u>not</u> place any files in other directories.

1.6 Upgrading

Upgrading from an earlier version of Serv-U is not difficult, but there are a few things to keep an eye on. Here are the steps to follow:

- If you are running Serv-U: Exit the program.
- Make a backup of the old Serv-U files. Just in case…
- Unzip the new Serv-U zip-file in a temporary directory and copy the contents over the existing files in your Serv-U directory (Watch out that you get the direction of copying right!).
- Go to the Serv-U directory and delete the file BWCC.DLL if it exists, version 2 no longer needs this file. **Note:** <u>Only</u> delete BWCC.DLL if it is in the Serv-U

directory, <u>do</u> <u>not</u> delete the one in the Windows\System directory as other programs might use this.
- If you are using '%' directives in signon and signoff messages, login messages or directory change message, then you might have to change those. The directives of version 2.0 no longer work and some of the directives of 2.0c have been changed. Take a look at the 'How to setup sign-on and/or sign-off messages' section for the new directives.

Your old SERV-U.INI file will be used by the new program so you <u>don't</u> have to re-install any users. All your settings, and your registration key (which also resides in SERV-U.INI - in case you registered), stay intact.

The 16-bit version of Serv-U now uses Microsoft's 3D-controls to create a 3D-look. This is done by a file named CTL3DV2.DLL which should be in your WINDOWS\SYSTEM directory. Most likely you already have this file, and therefore it is not part of the Serv-U zip-file. But if the dialog boxes look flat then drop me a line and I will make the DLL available. Note that this is <u>only</u> meant for the 16-bit version of Serv-U. The 32-bit version will <u>always</u> look flat on NT 3.51! On Windows 95 and NT 4 the 32-bit version will use the built-in 3D-controls, and thus have a 3D-look.

1.6 Known Bugs & Problems

At the time this documentation is being written the only known bug left in the program is the lack of on-line help. Everything that was found in earlier versions has been fixed. If you have any problems, please first check out the 'Frequently Asked Questions' on my Web site (**HTTP://www.cat-soft.com**). I will do my best to keep an up-to-date list of everything worthwhile there.

Unfortunately there are still a number of WinSock socket stacks that are not 100% compliant with the WinSock standard. Since Serv-U uses the stack to its fullest, this can result in a number of symptoms. The following has been observed: directory listings and file transfers do not work; after the first client the server seems blocked; WWW browsers seem unable to work with Serv-U; the PC is blocked when Serv-U is transferring a file.

Problem WinSock socket stacks
Below is the current list of socket stacks that are known to cause problems. If you find any others with problems, or solutions to the ones mentioned below, please let me know so I can pass the word on to others.
- The **SuperTCP** socket stack results in crashes of Serv-U. Both versions 4.00 and 4.04 were tested with similar lack of results.
- Reports indicate that the WinSock stack which comes with an **AOL** account does not work. So far it is unclear if this is an incidental or general problem.
- **FTP Software Inc's** socket stack: Their earlier stacks cause problems. Version 2.2 with WINSOCK.DLL v1.15.1 is rumored to work OK. Check out ftp://ftp.ftp.com/support/ftpsoft/winsock for updates. Their latest version WINSOCK.DLL can be found in WINSOCK.EXE (currently v1.17.1).
- **Spry's Internet-in-a-box** stack: Version 2.0 is rumored to work OK. Earlier versions had a multitude of problems. Check out http://www.spry.com:80/products/upgradingibox.html for updates.

- The **Air** socket stack is rumored to come from the same factory as Spry's. Won't work though.
- **Core Systems' Internet-Connect** stack does not work with Serv-U.
- **DEC's PathWorks** stack does not work either.
- Neither does **CISCO's** socket stack that comes with **Suite 100**.
- **Sun's PC-NFS** stack version 5.1A and later works, but earlier versions are a problem.
- Earlier versions of the **Microsoft Wolverine** stack had a tendency to bomb back to DOS when the network got busy. This has been fixed in version 3.11b, available from ftp://ftp.microsoft.com/bussys/clients/wfw/tcp32b.exe. A highly recommended stack!
- **Novell's LanWorkplace** stack seems to work for some people, but doesn't for others. In any event, you need version 4.2T3 or later. The 'T3' patch is available from http://netwire.novell.com/servsup/binhtml/1203536.htm.
- In case you have **Netmanage's Chameleon's Sampler** you need at least version 3.11. Reports are that it works fine on version 4.5.1 of Chameleon.
- **PC/TCP's OnNet** stack locks up the PC during file transfers, but it works fine otherwise.
- **CompuServe's** WinSock stack does not work with Serv-U (Wasn't that bought from Spry?!).
- The various versions of **Trumpet's** WinSock stack generally work fine. But, version 2.0b had a tendency to kill the socket Serv-U uses to listen for incoming clients, thus making the server unreachable. This has been fixed in version 2.1f, so better get that one from ftp://jazz.trumpet.com.au/pub/winsock/twsk21f.zip.
- **Quarterdeck's** socket stack does not work with Serv-U.
  - The **Windows 95 built-in 16-bit TCP/IP stack** usually works fine, and is <u>very</u> stable. But, it sometimes kills Serv-U's listening socket, resulting in 'Connection denied' messages from clients. This is a Win95 bug which MS acknowledges and claims to be the result of programs not freeing their sockets before terminating. The latter is probably BS, IMHO, but it lets them pass the buck. Most people will never see this while others seem to have the problem continuously. The advise is to use the 32-bit version of Serv-U on Win95.

'SLIP/PPP emulators - TIA, TWinsock, Pipeline'
If you are using a so called 'SLIP emulator' like TIA, TWinsock, or Pipeline, there are also a number of things to keep in mind. These emulators are not real Internet connections, and because of the way they have to do their work they have a number of peculiarities. Since the PC has to share the IP number with the host system you will generally not be able to use the default port 21 for Serv-U. Instead, you either have to use a high port number (generally above 8000) or set up 'port redirection'. Ask you Internet provider for details about the latter. A related problem is that the IP number displayed by Serv-U is not the IP number that clients should use to contact your server. It is merely a dummy to keep the socket stack of the PC happy. For the outside world the IP number of your server is the same as that of the emulator host system, i.e. that of your Internet provider. Another special effect is that clients will not be able to use 'passive' mode for data transfers. This means that regular FTP clients will work OK, but WWW browsers like Netscape and Mosaic will not work when Serv-U runs via a SLIP emulator. For the same reason it won't be possible to use clients that are also connected via a SLIP emulator.

Dynamic IP

If you use a modem and telephone to connect to the Internet via a service provider, then chances are that you will get a different IP number each time you connect to the network. You can see what your current IP number is by looking at the startup messages of Serv-U: one of the lines is the IP number. Serv-U does not have a problem with dynamic IP. Just make sure you start the server <u>after</u> connecting to your service provider, so Serv-U will report the current IP number. However, if you want to run a steady FTP site, then dynamic IP is a pain in the lower rear end! You users will have to know which IP number to contact and if yours is changing all the time it does not help…

Your IP number is <u>entirely</u> up to your provider and there is absolutely nothing Serv-U can do about it! It merely uses the number available. So, if you want a fixed ('static') IP number the only person to talk to is your Internet service provider. Sometimes it is possible to get a fixed symbolic IP name, even though the number varies each time (An IP name is something like 'www.cat-soft.com', you can use it instead of an IP number). Again, talk to your provider.

1.7 A Word to the Wise (but paranoid)

Many a word has been spent on the alt.winsock newsgroup on the try-out enforcement practice of Serv-U, and in a broader sense on the security of network programs. If you choose for the 'fully functional try-out version' then this program will communicate with a remote system to determine if it should be allowed to work or not. That is the way the 30 days of try-out are enforced. To that effect, information is exchanged between your PC and a server over here, and as people asked: How do I know that my password file is not being sent over? Another, but similar question is: How can I be sure there are no 'back doors' in the server, allowing access to the author at any time? The short and hard answer is: You don't know and you can never be sure!

I know this is not much of a deal, so I'll at least give you one option to establish my good intentions. To anyone who is interested, I offer to personally go over the source code with you (all 22000+ lines of it) and we'll compile it on the spot. You will understand that I am not going to hand out any source code. If you want to take it home you'd better bring along a whole lot of $$$'s!

It is worthwhile to realize that security is a problem with any type of network program. It is fairly easy for a programmer to put code in, for example, a WWW browser that will wait for 5 months, until the full moon and Venus coincide, then monitor PC activity and if a user hasn't been present for a few hours, or if it's 4 in the morning, starts sending over the whole hard disk to unknown destinations. This kind of thing is <u>not</u> easy to detect, don't let anyone tell you otherwise, and I still have to meet the system manager that keeps a network monitor running 24 hours a day, year after year and actually reads the log files. The bottom line is that you will have to trust the author of a program at some point, there is no other choice!

Now don't write to me that conversely I should trust you that "the check is in the mail and please send me the registration key in advance"… ☺

That is all I have to say about installing Serv-U. Next let's look into how we can use it to get some real work done. The next chapter will take you through all its features.

2. Using Serv-U - How To …

Rather than describing every menu and dialog box to its last boring detail, I though it would be more useful to present a task-driven approach. This chapter is therefore divided into sections that each describe how a specific task can be done, in the form of 'How to' questions and answers. There is a bit of overlap between the sections, so you might come across the same thing more than once. If you read this manual cover-to-cover then you will also have learned about every menu and dialog box detail in the process. Makes great bed-time reading!

2.1 How to Setup Your Server

You shouldn't believe all I say: This first section is going to go over the 'Setup - FTP Server' menu choice in every grueling detail! No, seriously, this dialog box stands so much at the basis of Serv-U's operation and the items in there do not fit very well in other sections, that it is a good idea to cover it first, and I will only discuss those items that are not covered by other sections. So, let's pull up the 'Setup FTP-Server' dialog box.

The first item is 'FTP port number'. This is the (you guessed it!) port number that the server will listen on for incoming FTP clients. The default is number 21, but you're free to fill in anything you want, provided it does not conflict with other network programs. Filling in a number that is already in use will make Serv-U complain bitterly, so that's easy to catch. Of course, the rest of the world expects a FTP server to listen on port 21, but changing it to another number is one great way of insuring that only you and some selected friends will know about your server.

The next item is the 'Maximum no. of users'. With this you set the maximum number of simultaneous users at any given moment. Setting it to 0 will not allow anyone to enter, leaving it blank will allow an unlimited number, or, more precisely, until the PC runs out of network sockets or other resources like memory. If you need your PC for regular work as well as being a FTP server, it is probably wise to set a maximum so normal operations will not be slowed down too much by clients. Likewise, 'Maximum no. of anonymous' limits the number of 'Anonymous' users at any given moment. If 'Maximum no. of users' is specified, then this will limit the total number of users, both regular and anonymous, even if 'Maximum number of anonymous' is set to a larger number.

Some people have asked what would be reasonable settings for the maximum number of users that would still keep the system workable. The answer is: It depends. It depends very much on whether these users are local and thus capable of using up a large bandwidth in data transfer, or alternatively, if these users are further away on the Internet and cannot get transfer speeds of more than about 10 Kbytes per second. Another concern is the socket stack. Some WinSock stacks start behaving erratically when they have to service large numbers of sockets. Most notoriously is or was the first version of the Microsoft 32-bit 'Wolverine' stack, which had the tendency to kick the system back to MS-DOS without any warning when heavily loaded. One way to find out how many users your system can handle, in case you run Windows 95 or NT, is to start the system monitor and keep an eye on the CPU usage percentage. This will give you a pretty good idea very fast. My experience is that even a measly 486 can handle 20 or so concurrent users with easy.

This brings me to another topic which is related to the above: Server stability is going to depend very much on the operating system you are using. MS-Windows 3.1 and WFW3.11 are inherently so unstable that you should not expect Serv-U to run for more than a few days at a time before a reboot is needed. Very much better is Windows 95, especially when used with the build-in socket stack. Having used it since the early betas it seems to work fine for several weeks at a time. Definitely recommended for serious server work! At the top of the stability list stands NT. The Cat Soft ftp site (ftp.cat-soft.com) has been running on NT for quite a while now, and it seems very stable!

Another often asked question concerns maximum transfer speeds that can be obtained with Serv-U: Let's just say that even on a measly 486 you don't have to worry about this unless you have at least a T3 network connection, Serv-U can easily saturate a T1 line (= about 1.5 Mbits per second). You can try out raw speed by yourself: Just start a FTP client, like WS_FTP on the same PC as where the server is running and transfer some files through it. This bypasses the network, so what you are seeing is Serv-U plus hard disk access and part of the socket stack. The conclusion from this will probably be that your biggest performance bottle neck is going to be your network connection.

Let's get on with the list. To make sure that users cannot log onto your server and keep connections open until eternity it is possible to set 'Time-out users' and 'Time-out anonymous'. If a connection has been idle for more than the number of minutes you specify here it will be automatically closed. Filling in 0 or leaving it blank switches off the time-out. It is a good idea to fill in some values here, since otherwise the system would slowly fill up with sockets that for some reason got stuck, not to mention users that connect and start a transfer just before they leave the office in the evening and then go home. Default values are 15 minutes for anonymous users and 10 hours (=600 minutes) for regular users.

If you would like to leave your PC wide open for the rest of the world you can uncheck the 'Enable security' checkbox. But beware: DISABLING SECURITY WILL ALLOW ANYBODY ON THE NETWORK TO DELETE/CHANGE/COPY EVERYTHING ON YOUR PC!!! The only reason I put this option in is to make it easy for people that have their own local network and don't want to mess with users and passwords. By default this option is, of course, checked. **DO <u>NOT EVER</u> LEAVE THE 'Enable Security' OPTION UNCHECKED IF YOUR PC IS CONNECTED TO THE INTERNET!!!**

The rest of the items in the server setup dialog box are covered in other sections of this chapter. So, just read on and you'll eventually get to know them all.

2.2 How to Add a User

One of the most fundamental tasks you are going to have to do is adding new users. Unless you switched off all security, you are going to have to deal with this.

Upon choosing the 'Setup - Users' menu option a dialog box is presented to you. It contains a list of all known users on the left and by clicking on one of those users you can see the settings for that user on the right side. Now, if you just started this server for the first time there will be no names in the list, short of Divine Intervention. If you are at this point and are about to setup your first user, just go ahead and type in the various fields. If there already are users defined you will see the settings of the first one and to get rid of this and create an empty sheet just press 'New'. Doing the latter will pop up a little box asking you for the new user's name, which I think speaks for itself.

The next thing can come in handy, so pay attention: You can fill in or change any name in the 'User name' field. If this new name does not exist it will be added to the list of users. If this name exists, the settings for this user will be changed to the ones in the dialog box. In short, this is another way to add new users. So, if you clicked on user 'James' and then go on to set his password to 'lightbulb' <u>and</u> you change the user name to something that does not exist yet, let's say 'Tanya', then you just created a new user Tanya with all the settings of James except for his password. This way of dealing with users might strike you as somewhat strange. The advantage of it is that you can take an existing user and, by making only the few needed changes, turn it into a new user.

Now let's take a closer look at the various fields in the 'Setup Users' dialog box. I've dealt with the 'User name' field, so this brings us to 'Group name'. Every user can be part of a group. The convenience in making users part of a group is that you can leave common settings for all users of a particular group blank and just fill them out in the 'Setup Group' dialog box, about which you can find more information later on in the 'How to use groups' section. This goes for <u>all</u> settings, including password, home directory and path access rules. To overrule a certain group setting, simply provide one for the user. Only exception for groups is the user with special name 'Anonymous'. For 'Anonymous' the system ignores any group setting you might have. In fact, if you try to enter a group for a user with that name it won't even get stored and is gone the next time you look at the settings for 'Anonymous'. This was done for security reasons.

We did get ahead of ourselves in the discussion of the various fields, so let me back up a bit. The 'Password' field is of course for adding a password for a user. The passwords are stored encrypted using UNIX 'crypt'. As is normal for the 'crypt' function, only the first 8 characters of the password are used. Don't loose any sleep over this though, it is still very safe. This algorithm works like a sausage machine: you put in a pig on one end and turn the crank, out comes the sausage. But, pushing in the sausage while turning the crank backwards will not get you a pig! It is quite secure, I wouldn't know of a way to get the plain text password back (the NSA might though). Once a password is stored you will only see the word '<<Encrypted>>' in the password field. This is to indicate there is a password. You cannot edit existing passwords, since the original is truly lost and only the encrypted form is kept. The only way to change it is by typing in a completely new one.

If you leave the password field empty this does not mean that the user has no password. Serv-U assumes that all users need a password and will try if there is a group setting and look at that for the password. If no password can be found then access will be denied to the user. There is a way to setup a user without a password, but it needs direct editing of the SERV-U.INI file. You can find the details in the chapter on Serv-U internals.

There is one more exception with passwords that you'd probably already know: The user name 'Anonymous' never has a password. Instead, Serv-U will ask for a E-mail address.

The 'Home directory' field is for the user's home directory (To kick in an open door). This is the place where he or she is put immediately after logging in. Each user <u>needs</u> a home directory, without one the server will not permit logging in! Of course, if a user is part of a group, and this group has a home directory you don't have to specify one here. You might want to, if this user needs a different one from the rest of the group. Home directories <u>always</u> need to be full path names, including a drive letter!

A large portion of the 'Setup Users' dialog box is devoted to 'access rules'. These determine to which files and directories a user has access. In general, you will have to make sure that the user has at least read access to his or her home directory. The quick-and-dirty way to setup access for a user is therefore to press 'Add' and enter the path of the user's home directory. Then check the appropriate access rights, which in this case should be at least 'read', 'list', and 'inherit', and you have a functional account. Although their use is pretty straight forward in most cases there are still some ins and outs to access rules. So please take a look at the section 'How to use access rules' which discusses them in more detail.

Obviously, clicking the 'Store' button will store the newly created user settings. But this is not the only way: You can also leave the setup screen by pressing 'OK', any unsaved changes will be stored as well. Also, if you already have other users set up, clicking the mouse on another user name will save changes.

We already came across the special user name 'Anonymous' to allow access without a password. The user name 'FTP' is understood by Serv-U as a synonym for 'Anonymous'. When a user enters 'FTP' it will be translated to 'Anonymous', so the latter's settings apply to 'FTP'. There is one more user name with special meaning to Serv-U: 'ALL'. Now where does this tie in? Well, every action requiring security clearance (checking a password during login, reading, writing, etc.) is first checked against the settings for the particular user. If no appropriate setting is found there, and the user belongs to some group, the group settings are checked. If still no corresponding setting is found, the user 'ALL' is consulted (if it exists). So 'ALL' works as a blanket for all users, providing the most common settings. Of course, this also provides a potentially big security hole, so be careful!

Most of the other buttons and items are explained in other sections, but let's go over the details of the 'Miscellaneous' dialog box here. If all you want is to set up simple FTP access you probably can ignore anything in there. The miscellaneous items let you fine-tune some details that might be useful. Just press the 'Misc' button to get into it.

If you want to disable the account of a user so he cannot log in without deleting it, this is done by checking the 'Disable account' checkbox. A tip: This also works in a group, so you can disable an entire group of users with a single mouse click!

The next item, 'Show path relative to homedir' can be very useful and powerful, but there are some snags to it so pay close attention: When this is enabled Serv-U will try to show all the directories a user deals with as if they are relative to the user's home directory (Hence the name). For example, if a user has home directory c:\users\anonymous, this would show up as '/' to the user. If he changed to c:\users\anonymous\uploads all he'd see is '/uploads'. It completely hides the real path with drive, and makes the FTP client believe it is in the root of the file system. This can be great in many cases, but there are also potential problems to it. Serv-U can only do this trick for directories that are either at or below the home directory of the user. For example, if the homedir is c:\users\anonymous again, there is no way that Serv-U can represent e:\pictures\furry relative to that. In short, with 'relative paths' enabled the user will not be able to go outside of his or her home directory (and the directories below the homedir).

By checking 'Hide hidden files' Serv-U will not show any files that have the file attribute 'hidden' in directory listings. This is useful for hiding directory change message files and link files.

The final checkbox lets you control if a user should be allowed to log in even after the maximum number of users has been reached. This is good for you yourself, as the system administrator. It would let you log into your own server at all times.

This brings us to the 'Message file' field. You can put a file name of a text file in there. The contents of this file will be displayed to the user just after logging in. More about this in 'How to setup user specific login messages'.

This should have you started in creating user account. The next section gives you a few details on how to change existing user accounts.

2.3 How to Change a User

For the most part the things mentioned in the previous section, 'How to add a user', also apply to changing user settings. However, there are a few peculiarities that deserve extra attention.

First of all, keep in mind that you can change a user's settings at all times in the 'Setup Users' dialog box. This includes the user's name, but care should be taken in doing this, since there might be side effects: If you change the name to a non-existent one and store the settings this will in effect create a new user with that name. Also, if you change the name to one that already exists then you are changing the settings for the user with that name! To drag our favorite example once more out of the closet: Let's say you want to change the password of user 'James'. So you clicked on 'James' and then go on to set his password to 'lightbulb' <u>and</u> you change the user name to that of another user, let's say 'Tanya'. Then Tanya is going to be mighty upset when she tries to enter your FTP server! James will of course have to remember his old password, since nothing changed for him.

Another peculiarity that can come in handy is when changes to user settings get stored. There are several ways to do this. Obviously clicking on 'Store' will do the trick. Another way is to press 'OK' and leave the user setup. This will save any changes too. The third and last way to store changes is by simply selecting another user from the list on the left of the 'Setup Users' dialog box. Up until you store changes you can recover the original settings by the pressing the 'Restore' button. This does not hold for the 'Delete' button. Pressing this will erase the selected user completely and there is no way to get it back (I'm a nice guy, so you'll get warned first when attempting this)!

If you're editing an existing user who has a password, the word '<<Encrypted>>' will be shown in the password field. This indicates there is a password, as opposed to not having a password in which case this can be supplied by the group settings. There is no way to recover the original password, once entered it is lost! So, there is no possibility to edit the existing password, you can only enter a completely new one. To do this, erase the word '<<Encrypted>>' <u>completely</u> and type in the new password.

Something that might be of interest to you if you edit users by directly changing the SERV-U.INI file using your favorite text editor (for example 'notepad'): Any changes you make to users take immediate effect. No restarting of the server is needed. This means that you could keep a local copy of your INI file and use a local version of Serv-U to edit it. Then just use Serv-U itself for uploading the changed SERV-U.INI file to the production site and you are all set.

That concludes all you need to know for keeping your user's settings up-to-date, but it is time to fill in a big blank that is of the essence to using Serv-U: The access rules.

2.4 How to Use Access Rules

The 'Access rules' part in the 'Setup Users' dialog box forms the corner stone of Serv-U's security. This is also the part that makes Serv-U unique: It allows very fine control over what a user can or cannot do. It allows you to specify access <u>per directory and even per file</u> for each user!

How does all this work? Take a look at the 'File/Directory access rules' list in the 'Setup Users' dialog box. This list contains a number of paths with access information coupled to each path. Access to the PC is only allowed according to these paths and their access information. <u>No access path, no access</u>! So, there is one path you might always want to put in the list: The user's home directory. The check boxes at the right of the path list show what kind of access the user has to the path in the list. Again, unless the type of access is specified in these check boxes the user won't get in.

Let's take a look at the different type of access we can specify per directory or file. There are eight different types of access information that can be set, four that apply to files, three for directories, and the final one is a special case. To start with file access:
- '**Read**' access, allows files to be copied from the PC using the FTP 'get' command.
- '**Write**' access, allows files to be copied to the PC using 'put', but not changed, deleted, or renamed.
- '**Delete**' access, allows the user to change files, rename, or delete them. Having 'Delete' access automatically includes write access (although it does no harm to specify both).
- '**Execute**' access, is meant for executing files through FTP, i.e. for running DOS and Windows programs remotely.

Then there are three items that deal with directories:
- '**List**' access, allows the user to retrieve a directory listing using the FTP 'dir' command.
- '**Make**' access lets the user create new directories at this path, i.e. the user can make subdirectories.
- '**Remove**' allows the user to delete directories.

The final item is somewhat special:
- '**Inherit**' means that the access rule automatically applies to all subdirectories of the path, i.e. the rule is inherited by subdirectories.

Most of the above should be no surprise to you, but let's take a closer look at a few of the less clear items. First, to allow a user to use the FTP command 'cd' to get into a directory any of the rights is sufficient. So, a user that has 'read', 'write', 'delete', 'execute', 'list', 'make', or 'remove' can change to the directory in the access path. Conversely, if a path is specified without any access rights (except maybe for 'Inherit') then the user has <u>no access</u> what-so-ever to this path.

A user has 'write' access to a file or path, but no 'delete' access, can upload files to the server as long as they do not exist already. This is good for an upload directory, because it allows uploads without the chance of changing previously uploaded files.

'Execute' access is meant for remotely starting programs and usually applies to specific files. Be careful in granting this right: For example, allowing a user to start COMMAND.COM also means that this user can delete <u>anything</u> on your hard disk! More on using this in the 'How to execute programs via FTP with Serv-U' section.

If you want a user to be able to see a directory listing you need to specify 'list' access. This is often used in the opposite way: For example for an upload directory you would want a user to be able to upload files, but not see and certainly not download anything that is already there. This could be done by only specifying the 'write' right, and leaving 'list' access unchecked.

A somewhat strange attribute in the list is 'inherit'. This does not so much say anything about access for a path, but how that access rule works for subdirectories. When 'inherit' is checked the rule will automatically be valid for all subdirectories of the path in the rule, as if these subdirectories inherit the access rule. This is usually convenient, it means that with a single access rule you can control access to a whole branch in the directory tree. Also, if a user is allowed to create subdirectories, they will have the same access rights as the parent if 'inherit' is checked. In short, you are going to want to leave 'inherit' checked in most cases, but sometimes it can be convenient to specify access for a single directory in a tree without affecting subdirectories and that's when unchecking it can come in handy.

When a user executes an FTP command concerning files or directories, the user's access path list is checked to see if the command should be allowed to proceed. Now pay very close attention, this is the part of Serv-U that seems to be the least understood: The list is evaluated from top to bottom, and evaluation stops as soon as an applicable rule is found. 'Applicable' means that if it is a file the user wants to do something with, then the file name must appear in the access path list, or the directory of the file must be in the list, or finally, a parent directory of the file must be in the path list with 'inherit' checked. Directory access uses a similar mechanism. It follows from this that **THE ORDER OF THE PATH ACCESS RULES IS IMPORTANT!!!** Unless the access rules are in the right order you will not get the result you want!

Since one example can say more than a thousand words, or something along that line, let's work through a typical situation. Assume you want to setup an 'Anonymous' FTP site. This needs a directory tree with all the goodies the users might want to download, for which they need read access. You also need an upload directory where users can upload new goodies, but you don't want others to be able to immediately get their greedy fingers on it, since you want to check for viruses first. So, this upload directory needs write but no read access. We decide to put everything on the big network drive, 'Y:', under the 'ANONFTP' directory. We also create the 'UPLOAD' directory here for uploads. In Serv-U we would create the user 'Anonymous' with the following access path rules (and in this order):

        Y:\ANONFTP\UPLOAD   - write, inherit rights
        Y:\ANONFTP   - read, list, inherit rights

Reversing the rules will not work: If a user tries to write to the upload directory, the security mechanism will check against Y:\ANONFTP and conclude that UPLOAD is a subdirectory, so the rule applies, and the rule grants only read and list access. Please take note that write access does not allow a user to get a directory listing of the UPLOAD directory, so not only won't a user be able to download anything from there, the user cannot even see what files are uploaded.

If the drive letter is left out of a path, it applies to all drives. So, a fast way to get full access to all files on all drives is:

    \\   - read, write, delete, list, make, remove, and inherit rights

Now, the same mechanism that determines access to directories also applies to files. It is possible to grant access to specific files on a per-file basis. Let's take the previous example about the anonymous FTP server. We want to put a file 'SECRET.TXT' in the ANONFTP directory, but nobody is allowed to read it of course. So, our access paths list would look like this:

    Y:\ANONFTP\SECRET.TXT   - no rights
    Y:\ANONFTP\UPLOAD   - write, inherit rights
    Y:\ANONFTP   - read, list, inherit rights

Again, the order of the paths is important! The directory access rights do not have any meaning for files. Alternatively, if SECRET.TXT was a directory instead of a file, the above settings would keep users completely out of this directory.

2.5 How to Setup Anonymous Access

Since setting up anonymous access is such a common task when running a FTP server, I want to present a separate section on this subject. It is not much different from setting up any other user, so this is going to be a 'follow the recipe' kind of instruction, with the specifics for anonymous thrown in.

First, a tip: If you are running Windows 95 it might be worthwhile to create a new 'DriveSpace 3' volume with all the files you want to make available for anonymous FTP access. This not only provides you with a lot more space due to the compression (which at regular FTP speeds has no impact on performance), but also effectively isolates the users from more sensitive areas like your C: drive. So, for the rest of this story I will assume you just created a DriveSpace volume named 'F:'.

Let's say all the good stuff for our users is going to be in subdirectories of F:\ANONFTP, and we want to allow uploads of new files to the F:\ANONFTP\UPLOAD area.

The first task in setting up anonymous access is creating a user with this name. Thus, go ahead, pull up the 'Setup - Users' menu and its dialog box. Fill in the 'Username' field with 'Anonymous'. The 'Group name' and 'Password' fields stay blank, since they are ignored for anonymous anyway. Now, the 'Home directory' field needs to be set to F:\ANONFTP since that is the point we want the users to start when they log in.

We're almost there, but the users still need access to those directories. So click the 'Add' button and enter the path F:\ANONFTP. When this is done the access rights are already set, since the default of 'read', 'list', and 'inherit' are exactly what we want. Now for the upload directory: Click 'Add' again and fill in F:\ANONFTP\UPLOAD. Serv-U will place this above the previously entered access rule, which is good. However, the access rights are not exactly what we want, so uncheck 'read', check 'write' and uncheck 'list'. You should now have the following two access rules in the list (In this order!):

F:\ANONFTP\UPLOAD	- write, inherit rights
F:\ANONFTP	- read, list, inherit rights

This is all we have to do in the 'Setup Users' dialog box itself, but there are some things in the miscellaneous section that are useful, so press the 'Misc' button. Once in there, check the 'Show path relative to homedir' button'. It causes anonymous users to see all directories and path names relative to their home directory. So, if your anonymous users have 'F:\ANONFTP' as their home directory, they will receive back '/' when they inquire with the PWD command. Similarly, every reference they make to a file or a change of directory is taken relative to this path. The main reason it is in there is because most WWW browsers need 'change directory' access to the '/' (=root) directory to make them work. This way they believe they have access to the root and are happy, while on a PC you don't always want to give users access to your real root directory. The disadvantage of having this mechanism is that anonymous users are restricted to their home directory and below, nor do they have access to other drives. Sometimes you want them to be able to have access outside their home directory tree and unchecking this option will allow that. Anonymous users will then be treated like any other user as far as path names are concerned and they will be able to change to parallel paths and other drives if their access rights permit this. The price you pay is that unless you provide some form of access to the root of their home directory, i.e. 'F:\' in our example, WWW browsers won't be able to log in.

When they log in we want to let them know the especially good things and give them instructions on how to upload, and to do that we need a login message file. Netscape will show such a message in a very nice way, at the top of the directory listing. So, let's create a file named LOGANON.TXT with the following contents:

```
Welcome %name from %IP!
For the latest in nature's feline beauty see directory IMAGES
You can hear them in the directory MEOW
Please keep uploads smaller than 100 Kb and place them in UPLOADS
```

Since I can imagine that not everyone envisions cats when hearing about 'pretty pictures' please feel free to change the text according to your wishes! The words starting with '%' are directives to Serv-U which will be replaced by real text when the user logs in. '%name' becomes the user's name, and '%IP' either the IP number or the IP name of the user (whichever is available). There are many more '%' directives and you can find more information about these in the 'How to use sign-on and/or sign-off messages' section.

Now let's place the file LOGANON.TXT in F:\ and thus enter F:\LOGANON.TXT in the 'Message file' field.

To store our settings press 'OK', and in the user setup dialog box press 'Store'.

There are a few more things that are important for anonymous access. They can all be found in the 'Setup - FTP Server' menu choice. The first field of interest to us is 'Max no. anonymous', this determines how many anonymous users can be logged in at the same time. If you leave this blank then Serv-U will allow any number. Use your own judgment what you think your machine can handle. Next is 'Time-out anonymous'. This determines how long Serv-U will wait when anonymous users are not doing anything before kicking them off the server. The default of 15 minutes is a good value, still allowing slow transfers and bad connections to work while keeping

the server reasonably clean of all those Netscape users that stay logged in after their transfer is long done. Setting the time-out to 0 or leaving it blank means that Serv-U won't check how long a user has been idle.

The last item we need to concern ourselves with for anonymous users is the 'Check anon. passwords' option. By default this is unchecked, which means that anonymous users can type anything as a 'password' to get into the server. If this option is checked, then Serv-U makes sure that the 'password' entered by anonymous users is something that looks like an E-mail address. Netiquette prescribes that anonymous users send their E-mail address as a password, alas this good custom is disappearing fast and version 2.0 of Netscape does not even have the ability to do this anymore. So, unless you want to severely restrict access and teach people good manners it is probably best to leave this unchecked.

Just so you know: As is customary for FTP servers, Serv-U also looks for a '-' (hyphen) as the first character of the anonymous 'password'. If present, all login and directory change messages are suppressed.

Well, you made it through and now know just about everything there is to know about setting up anonymous access to your server. You can enhance things further with sign-on and sign-off messages, directory change messages and links, which are covered in the following sections.

2.6 How to Use Groups

To ease the burden of maintaining a large site somewhat, Serv-U uses the concept of 'groups'. This allows you to group together users with similar settings. All the similarities are stored in the group information, leaving only the differences per user for you to set up.

To edit groups select the 'Setup - Groups' menu choice. The dialog box you see should look familiar by now: It's the same one as for the user setup. So, if you know how to set up users, you also know how to set up groups! The only item that is not applicable for groups is the 'Username' field, so this is grayed out, instead you use the 'Group name' field to specify the name of the group you are editing.

To show the power of groups a quick example: Let's say you are the Pentagon system administrator and want to create FTP access for everybody in case they are on field trips. So, you hook up this old PC to the net, install Serv-U and register it (Hypothetical situation). Then you proceed to create a group 'StarWars'. Now you go on to set the password for this group to 'RonaldR', and their home directory (all their files are shared anyway) to 'y:\super\secret\starwars'. You fill in some access path rules as well, and you're all set: The only thing left is entering the user names, you don't have to provide any other information per user. A 10 minute job.

Now there's this occasional guest user, 'BillyC'. You don't want him to get into certain directories, so you make him a member of the group but specify those secret directories in his access path rules with 'no access', and you're all done.

As you already saw from the previous example the user settings have precedence over the group settings. This allows you to selectively override certain parts of the group settings, making groups a very flexible mechanism for user maintenance. To summarize: Serv-U first checks the user's settings, then looks if the user is part of a group and if so uses this group's settings, and finally, if

nothing is found yet the settings of a user with name 'ALL' are used if they exist. The only user that cannot be used together with a group is 'Anonymous'. For security reasons only the user settings themselves are used here.

Well, by now you know most of what there is to know to run a smooth Serv-U FTP server. The next sections are mostly about goodies that make life easier or more colorful. Next you'll learn how to find out what users are doing on your server and if they are nasty, how you can boot them off.

2.7 How to See Who is Logged In

Even though Big Brother might not be watching you, you can certainly watch your users! Select the 'File - User Info' menu choice and up pops the user information window. The upper part shows a list of all users currently connected to your server. By clicking with the mouse on one of the users you can get extra information about that user in the lower part of the window.

There are a few things to keep in mind concerning the displayed statistics: The IP name is found by initiating a reverse DNS lookup of the IP number. It can take time before an answer comes back from the DNS server, and there are quite a few IP numbers that do not have an IP name. So, it is not at all unusual for a user not to have an IP name.
Then, the transfer speed displayed should not be taken too absolute. Since the socket stack buffers any send activity (Serv-U can write lots of bytes to the stack in a very short time, after which the stack sends it off at its leisure) the indicated speed can be much higher than it actually is. Also, if there are local transfers going on with a very high speed then Serv-U and the socket stack can get into a tight loop of sending or receiving, and the user information might not get updated for some time. This is normal ("Not a bug, but a feature!").

This brings us to the 'Show Commands' button. Pressing this will pop up another window which will show you all the FTP commands the user sends and the replies from Serv-U. It tracks the commands of the in 'User Info' selected user and by clicking on another user you can switch.

The last button of interest in this window has the prosaic name 'Kill User'. This is the place where, if so inclined, you can alleviate all your frustration after a particularly bad day at the job. For that reason the entire next section is devoted to it.

2.8 How to Kick a User Off the Server

If someone is doing things you don't like, or you just want to be mean, Serv-U offers you the option of kicking that user off your server and keeping him or her off.

This is done by selecting the 'File - User Info' menu choice. In the window that pops up as a result you select the user-to-be-killed, and proceed by pressing the 'Kill User' button. This will pop up a little dialog box asking you if you're really sure about this, and offering several options to keep the user in question off on a more permanent basis.

You will see two checkboxes. The first is 'Refuse IP access in future'. If you check this and then press 'OK' the user's IP address will be added to the list of IP numbers that are refused access. This will prevent that user from logging in again, or at least it not from the same IP number. We will talk about access and IP numbers in greater detail in the 'How to limit access by IP number section'.

The second checkbox is 'Disable account' and checking this does exactly that: The 'Enable' checkbox in the 'Setup Users' dialog box will be unchecked. Nobody will be able to use that login name until the account has been enabled again.

2.9 How to Setup Logging

When you select the 'Setup - Logging' menu choice you will see a dialog box that lets you specify what events you want to log and where to log them to: either to screen only or to both the screen and a logfile.

The first two items deal with logging to a logfile. The first one, 'Enable logging to file' switches writing to a logfile on or off. The second item, 'Logfile' is meant for entering the path and file name into which all the messages are written. Of course, logging will only work when a valid logfile name is entered, it has to be a full path name including a drive letter.

Serv-U gives you a wide choice in what should be logged or not. In all there are seven different categories that can be individually enabled or disabled. The items are self descriptive except maybe for three: 'Log FTP commands' logs every command as it is received by the server and 'Log FTP replies' logs the command replies that are sent back by the server to the client. These two options are mainly intended for diagnostic purposes and are switched off by default. The last option is 'Log IP names' and when checked, Serv-U will try to find the IP name of a user by doing a reverse DNS lookup on the IP number. Keep in mind that DNS lookups can take just about any amount of time, cost a fair amount of overhead, and not all IP numbers have a IP name associated with it. When this option is checked the IP name of a user will be logged as soon as it becomes available.

Messages are always logged to the Serv-U window, regardless of the logfile settings. There is no difference between the messages on screen and the ones in the logfile, although some things are only shown on screen. The latter are server and program related matters, like version number, server going on/off-line etc.

Log messages always have the same layout. The reason for using such a strict format is to make it easier to search for specific messages or certain types of messages. This also makes it possible to do automatic accounting by machine reading the logfile if you need it. The logfile can be read or copied at any moment, even when Serv-U is running (It is also possible to get the file via FTP using Serv-U itself!). The format is as shown below, in stylized form:

    [n] DATE TIME - (xxxx) MESSAGE

The first number, 'n', indicates the type of message that is being logged. Currently there are seven different categories:

1	- system messages (problems etc.)
2	- FTP commands (from client to server)
3	- GET file transfers
4	- PUT file transfers
5	- security related events (users logging in etc.)
6	- FTP replies (from server to client)
7	- WinSock socket stack activity (only used for diagnostics)

The second number, 'xxxx', is a unique ID assigned to a client the moment the connection is made. All further messages concerning that client will use the same number. Again, this was done to make it easy to do automated accounting, or, to find back events using the 'search' facilities of every editor.

You can rename, move or delete a logfile at any time while the server is running, there is no need to stop the server. You can even copy a logfile to a remote system using Serv-U itself! If you want to temporarily stop logging to file and see it on screen only, you can uncheck the 'File - Logging' option in the main menu. Needless to say this has to be checked to enable logging to file.

One more note: If you log GETs and PUTs then Serv-U will also tell you the average transfer speed. For GETs the value Serv-U shows can be a bit optimistic, especially for small files, as a result of the socket stack buffering the bytes that are sent over the network. Serv-U can very quickly transfer 10-20 Kbytes into the socket stack, which then proceeds to send it over the network at its own leisure. However, Serv-U has no way of finding out when the actual transfer is complete, and assumes that when it has sent the block to the socket stack it is done.

Enough boring talk, let's get back to the fun part! Next will be explained how you can pour your heart out to your users through all kinds of messages.

2.10 How to Use Sign-on and/or Sign-off Messages

Your FTP-server can display a welcome message every time a user connects to it. This can be very useful to provide users with information about your FTP server, like where to find games, or 'Serious Software'. Likewise, you might want to say good-bye to them when they leave, or remind them to send that check . . . The way to do this is by entering a text in the 'Sign-on/Sign-off' dialog box which pops up when you choose the 'Setup - Signon/off' menu choice.

There are several special words that you can enter in your sign-on and sign-off text which get expanded while being sent to a client. They all begin with '%'. Here is the list:

%Time	- displays the current time on your PC
%Date	- displays the current date on your PC
%UNow	- displays the current number of Serv-U users connected
%UAll	- displays the number of users since the server was started
%U24h	- displays the number of users in the last 24 hours
%MaxUsers	- displays the maximum no. of users, as set in 'Setup - Server'
%MaxAnonymous	- the maximum no. of anonymous users, as set in 'Setup - Server'
%Name	- displays the user's login name
%IP	- displays the user's IP number or name if available
%Dir	- displays the user's current directory
%Disk	- displays the user's current disk drive
%DFree	- displays the amount of free disk space on the user's current disk in

Kb

%FUp	- displays the number of files uploaded by the current user
%FDown	- displays the number of files downloaded
%FTot	- displays the total number of files transferred
%BUp	- displays the number of Kbytes uploaded by the user
%BDown	- displays the number of Kbytes downloaded by the user

%BTot	- displays the total number of Kbytes transferred
%TConM	- displays the total connect time in minutes
%TConS	- displays the connect time in seconds - to be used with '%tconm'
%RatioUp	- displays the 'upload' ratio part for UL/DL ratios
%RatioDown	- displays the 'download' ratio part for UL/DL ratios
%RatioCredit	- displays the current download credit for UL/DL ratios (Kb or 'files')
%QuotaUsed	- displays how much disk quota is used in Kb
%QuotaLeft	- displays how much disk quota is left in Kb
%QuotaMax	- displays the maximum amount of disk space that can be used in Kb

Not all of these directives make sense for both the signon and signoff text: There is little point in putting '%disk' in the signon message, since the user is not yet logged in. Likewise for some other directives. However, these same directives can also be used in directory change messages and login messages, which are covered in the following sections. There they can be very useful.

You could make the following signon text:

```
Welcome, it is %time on %date, and you are user number %unow
Over the last 24 hours, %u24h people have visited this site
```

I'm sure you'll figure out by yourself what this will look like to the user . . .
Please keep in mind that the average client has only 80 characters per line, and the first four are taken up by the reply code. So keep your lines short if you want the users to actually see your prose.

2.11 How to Setup User Specific Login Messages

After astonishing the user with your signon message you can go for the kill by also providing a user specific login message. These messages should be put in one or more text files and the file name can be filled out in the users or groups setup dialog box, in the 'Message file' field of the miscellaneous part, i.e. press the 'Misc' button in the user or group box to get there.

The file name you enter in the 'Message file' field can have several forms: You can use an absolute path with a drive, for example C:\FTP\LOGMES.TXT. This will of course make Serv-U use this file. You can also specify only the name, and no path, for example LOGMES.TXT. This will cause Serv-U to look in the user's home directory for this file and if it is there it will be displayed to the user. Finally, you can specify a path but no drive letter, for example \FTP\LOGMES.TXT. This means that Serv-U will add the drive of the user's home directory to make a full path. The latter two ways of specifying the file can be very useful for groups, i.e. with just a single file name set up you can provide all the users that are part of that group with a separate login message.

Of course, you can use all the '%' directives mentioned in the previous section in login messages. Try for example:

```
Hello %name from %IP!
```

With a little luck Serv-U will have found the IP name of the user by the time he or she logs in. So you can impress them by showing you know where they are (Big Brother IS Watching!). By the

way, Netscape displays the login message in a nice looking way at the top of the screen (The sign-on message does not get displayed by Netscape).

2.12 How to Use Directory Change Messages

It can be very useful to tell a user some specifics when he enters a directory. For this purpose Serv-U has directory change messages. There can be a different message for each directory.

As with the login messages, the directory change message text should be stored in a file. In the 'Setup - FTP Server' menu choice you can specify which file name Serv-U should look for when the user changes directory. You can specify the file name in two different ways: First is an absolute file name, for example C:\FTP\CDMES.TXT. As you'd expect this will display that file for each directory change. Then it is possible to use a file name only without the path, for example CDMES.TXT. This will make Serv-U look for and display the file with that name, in the directory into which the user is changing. The latter one is the usual way to use directory change message files, since this enables you to specify a different message for each directory.

The server setup dialog box has room for a 'primary' and 'secondary' directory change message file. When looking for a message file, Serv-U will first try the primary file name. If there is no file name specified in there, or the file cannot be found, the server will try to find the secondary file name. This is very useful in case you have specific dir change message files for some directories, using a relative file name as the primary directory change message file. You can then set up an absolute file name as the secondary directory change message file. This will show the users the primary file message when available, and the secondary one if there is no primary (For example because it's a CD-ROM so you cannot place directory change message files in its directories).

Like sign-on and sign-off messages you can use any of the '%' directives in directory change messages. Take a look at the previous sections for a list of all directives.

Netscape will display those directory change messages at the top of the page. One more trick: If you enabled the 'Hide hidden files' feature in the miscellaneous user setup, you can hide the directory change message file itself from the user by making it 'hidden'. Hidden files are files with the DOS/Windows 'hidden' attribute set, something you can do using the DOS program 'Attrib' or by changing the 'properties' of a file in Win95 (using the right mouse button). So, if you make your directory change message files hidden they won't show up in the directory listings while users still get to see the messages.

2.13 How to Use Links à la UNIX

Now for the *pièce de resistance* (pardon my French), which sets Serv-U miles apart from any other FTP server that I know of for MS-Windows: Links!

If you know UNIX then you know what I am talking about when links are mentioned. If not, then try the following: Links are names of files and/or directories that appear in the directory listing but they are not in the directory that the user is listing. Rather, they point to the actual directory or file and can be used by the user as if they were the real thing itself. A little like the Windows 95 'shortcuts', but more real: When you delete a link through Serv-U the actual file or directory is gone! The great value of links is in showing users that you have all kinds of goodies available on

other disk drives. The user can then simply 'cd' to a link and thus change to a completely different drive and directory.

The way Serv-U works with links is by reading them from a text file. You can have different files and thus different links for each directory, much in the same way as the directory change messages work in the previous section. The file name is entered via the 'Setup - FTP Server' menu selection (Guess where!). The server setup will let you specify a primary and secondary link file. During use the server will first try to find the primary link file, and if that one doesn't exit it'll try the secondary file name. As with message files there are several ways to specify which file Serv-U should use for links: You can enter a file name by itself, for example LINKS.TXT. This makes the server look in the user's current directory for a file with this name and if found it is merged into the directory listing. This is the way to set up different link files for each directory. The second method is by specifying a full path name, for example C:\FTP\LINKS.TXT. As you'd expect the result will be that the same links are shown in each directory. Finally, you can leave out the drive letter, for example \FTP\LINKS.TXT, which means Serv-U adds the current drive to the name and then looks for the file. Useful for setting up links that change per drive, but not per directory. If you make the link file's attribute 'hidden' and enable the 'Hide hidden files' feature in the miscellaneous section of the user or group setup then users won't see the file itself appear in their directory listings while they still get to see the links.

In a typical situation you'd specify a relative path name for the primary link file, and an absolute path name for the secondary link file. For example, if you enter the file name LINKS.TXT as the primary file name Serv-U will look for this file in the users current directory. This lets you use different LINKS.TXT files for different directory, typically with links specific to that directory. You'd then put all the links you want to always show the user, even when they are on a CD-ROM, in a secondary link file with an absolute path name, let's say C:\SERV-U\ABSLINKS.TXT. A good candidate for the secondary link file would be a link to the user's home directory.

Now, what should a link file look like? Each line of the file describes a different link. First comes the link name, which is what the user is going to see in the directory listing, then a '|' which acts as a separator, and finally the file or path that the link should point to. Here is an example link file that should make all this a little clearer:

```
D-Drive           | d:\
CD-ROM            | f:\
Home              | ~
Fun Stuff         | g:\public\games
One up            | ..
Pointless         | \junk\..\more\..
Poem              | c:\culture\poem.txt
```

As far as the user is concerned links behave like the real thing: They can be used for 'cd ' if it's a directory, 'get' for a file etc. Only difference is that if a link gets deleted the file or directory it points to will get deleted but the link will continue to show up in the directory listing. Needless to say that it is your responsibility that the links in a link file point to something that makes sense.

So much for the fun part, back to more serious matter with more about how to select whom you want to let on your server and whom to bounce…

IMPORTANT NOTE: A point that seems to cause much confusion and is the source of much of the E-mail I'm receiving is the effect of links and using the 'Show path relative to homedir' feature in the miscellaneous dialog box of the user and group setup. People will make all these beautiful links to other drives, CD-ROMs and the like, then switch on 'relative paths' and act surprised their links don't work. Once again, while using 'relative paths' Serv-U will try to show the users path relative to the home directory for that user. This only works for paths at or below the user's home directory. For example, if the homedir is C:\USERS\JIM, then a link to C:\USERS\JIM\UPLOAD will work fine, but a link to E:\PICTURES will not!

2.14 How to Set Up Upload/Download Ratios

Sometimes you want to force your users to upload a file first, before they can download anything. Serv-U can do this for you, and more!

The setup for UL/DL ratios can be found in the user and group setup dialog boxes, by pressing the 'U/D Ratios' button. In general, UL/DL ratios will let you specify how much a user can download for each upload. For example, if for a user the ratio is set to 1 / 3 this means that for each upload the user can download 3. So, after uploading 2 the user has 'credit' for 6.

Serv-U can handle two types of ratios: Either it will keep track of the uploads and downloads for each time the user logs in separately (the 'per session' settings), or it can keep track of file transfers over multiple sessions, even while several clients are logged in using the same user name (the 'overall' settings). For each of the 'per session' and 'overall' choices Serv-U can count files (each file uploaded or downloaded counts as '1'), or it can count the actual file sizes in bytes. Depending on which of the two is used the ratio and credit setting will either mean 'files' or 'bytes'.

For the UL/DL ratio setting the actual numbers are not important. What counts is of course their ratio. For example, setting a upload/download ratio of 1 / 2 means (for 'files') that each uploaded file gives the user two files 'credit' for downloading. Likewise, if bytes are counted each uploaded byte gives two bytes 'credit' for downloads.

Every upload is converted into 'credit', and every download is taken from that credit. In a 'per session' setting the credit entry in the dialog box is intended for entering an initial credit value that the user gets when logging in. For an 'overall' setting this is the actual current credit for that particular user, which can be edited in the ratio setup.

There is an important item which is closely related to UL/DL ratios, which is named "**free files**". These are files that will not be counted for the upload/download ratio count, something very useful for README.TXT files and the like.

"Free files" can be set up in the 'Setup Server' dialog box, by pressing the 'U/D Ratios' menu item. Just fill out the file name in the text edit box and press the 'Add' key. This will put it in the list which shows all the current "free files". The file name can be a name only, i.e. without path. That way any file with that name in any directory will be free. For example, adding INDEX.TXT will make any file on the system with that name freely downloadable. You can also specify a full path in case only a specific file should be free. For example C:\FTP\INDEX.TXT will make INDEX.TXT free only if it's in the C:\FTP directory. You can also leave out the drive letter, which is taken to mean 'for all disk drives'.

The free file names can contain wildcard characters: '*' and '?'. For example adding *.TXT will make all files ending on .TXT free on your system, or A???.TXT will make all files starting with A followed by 3 other characters and ending on .TXT free. Wildcard characters can also be used in a path. For example, **.TXT would make all files on all drives ending on .TXT free if they are in a sub-directory one level deep (like C:\FTP\INDEX.TXT would be).

2.15 How to Limit Disk Usage of Users

Unless you have deep pockets and consequently large hard disks you are going to want to limit the amount of disk space that your FTP users can occupy. Again Serv-U comes to the rescue by letting you specify exactly how much a user can use.

For this purpose the user and group setup boxes have the 'Quota' button which brings you to the disk quota dialog box. In there you have two items: One to specify the maximum amount of disk space a user can use, the other to specify what the current amount is. Even though the setup is in Kbytes, internally Serv-U is accurate up to the byte.

The current amount lets you set up an initial value, in case you are filling out the user's details for the first time. It also keeps track of the disk usage as time goes by, and you can thus change it whenever you feel like it.

2.16 How to Limit Access by IP Number

The 'Setup - IP Access' menu choice will pop up a dialog box which provides the means to restrict access to your FTP server to certain IP-numbers. If for example, you work at a university and only want your faculty members to be able to access the server, then this is a great way to do it. In the upper left corner of the dialog box you can choose which type of rules you want to specify: 'Deny' or 'Allow' rules. The deny rules decide who should be kept out, the allow rules indicate who should be welcomed. **THE ORDER OF THE RULES IS IMPORTANT!** When a client contacts the server, the rules are looked at FROM TOP TO BOTTOM. The first matching rule applies, and evaluation is stopped. If there are no IP-access rules everybody can enter the FTP server. As soon as there is one rule, only those clients that pass the rule check are allowed to enter.

You can select which type of rules you want to have Serv-U evaluate first, in case both deny and allow rules are present. Just select the right one in the 'Evaluation Order' part. This can be useful in case you need to specify a few exceptions to a general rule.

You can type in a new rule in the 'Rule' edit and then use the 'Add' button to add the rule to the list. The 'Remove' button will remove the currently selected rule from the list. To change the order of the rules you have to select one by clicking the mouse on it, and then use the 'Up' and 'Down' buttons to move it around.

Rules are nothing more than IP-numbers or ranges of IP-numbers. There are two special characters: the star '*' and the hyphen '-'. A star functions as a wildcard for checking the number. Any number will match that section of the rule if it is a star. The hyphen is used to denote a range of numbers. Simply separate the starting and ending values by a hyphen. For example, say all IP-numbers in your company look like 134.56.34.xxx with 'xxx' being any number. Now, you want to restrict access to your FTP server to other members of your company only. The way to do it is to create an 'Allow' rule that looks like this:

134.56.34.*

That's simple, isn't it!? Likewise, if you know that the competition has IP-numbers in the range 168.76.xxx.xxx, you can keep them out of your server with the 'Deny' rule:

168.76.*.*

Now, you need to share some of your files with a few colleagues, and management in your company is too cheap to install a local network. You find out that their PC's have IP-numbers 134.56.34.128, 134.56.34.129 and 134.56.34.130. You could of course make three 'Allow' rules, each with one of these numbers. A faster way to do this is to make a single rule like this:

134.56.34.128-130

The special characters '*' and '-' don't need to be at the end of the IP-numbers, any place will do. The rule 221.*.76-154.89 is perfectly OK. I wouldn't know when you'd need this, but, hey, the world is a strange place! Remember, order is important and deny rules are always evaluated before the allow rules. Experiment a bit, and you'll get the hang of it.

Since people keep asking me this: A few words about the security of IP-access rules. The check on IP number stands pretty much at the top of the Serv-U security mechanism and all connections have to pass through this rather narrow bottle neck. The code that does the checking is quite simple (In contrast to the code that does the access rule checking for paths), so it is unlikely that there are any loopholes or bugs left in there. So, in my humble and honest opinion this is one feature that makes for a <u>very</u> secure FTP server if that is what you need!

2.17 How to Print via FTP

Serv-U also allows access to all PC ports: PRN:, LPT1:, LPT2:, LPT3:, LPT4:, AUX:, COM1:, COM2:, COM3:, and COM4:. This can be a convenient way of setting up a 'network' printer by transferring files directly to PRN: or LPT1: using FTP. These ports are treated like any regular path name, so a user needs access rights to use them. Thus, to make a printer on PRN: accessible and a modem on COM2: the user needs the following access rights in the 'Setup Users/Groups' dialog box:

PRN: - write and delete rights
COM2:   - read, write and delete rights

There seems to be much confusion on how to actually use this feature, therefore a short explanation. These ports behave very much like files that always exist and can be found in every directory although they don't show up in directory listings. So, no 'cd' commands are needed, just transfer your file-to-print to the port with the printer. For a command line FTP client this would look like 'put FILE.TXT PRN:' to print file FILE.TXT on a printer attached to port PRN:. For point-and-click type FTP clients you have to specify that it should ask you for the destination file name. Then, when prompted for the destination file name you enter 'PRN:' or another port of your choice. For example, for the popular client WS_FTP you have to check the 'Prompt for destinations file names' option in the 'Session Options' section. After doing that you can simply upload the file-to-print and it will ask you for the destination, for which you fill in the printer port name.

The above procedure should work well to print ASCII text and text-based protocols like PostScript (If your printer supports it). However, with binary files your mileage may vary: It seems that the port closes when a ^D is encountered in the stream that is being printed.

2.18 How to Execute Programs via FTP with Serv-U

Serv-U allows you to start DOS or Windows programs remotely using a FTP client. Keep in mind though that Serv-U is not a Telnet server, and any program output or windows will be displayed at the PC where Serv-U is running. However, for many applications the ability to start a program is enough, or you might be able to redirect input and output, i.e. like 'MYPROG.EXE < INPUT.TXT > OUTPUT.TXT'.

The first requirement to using remote program execution is that the user needs sufficient access rights to start the desired programs. This is controlled by the 'execute' access right in the 'Setup Users/Groups' dialog box. You can either give a user execute access to the directory where the program can be found, or you can specify the exact program name. For example, if we want to be able to execute COMMAND.COM remotely, and this program can be found in the C:\DOS directory, then the following access rule would do the trick:

       C:\DOS\COMMAND.COM   - execute access

Now, to actually start a program remotely you need to use the FTP command 'SITE EXEC' from the FTP client. Usually you cannot do that directly, since the client doesn't understand this command, and you need to tell the client to send it as a literal string to the server. for example the UNIX FTP client (and others like it, as the one that comes with Windows 95 and NT) uses the 'QUOTE' command for this purpose. So the full command line to let COMMAND.COM copy a file from A.TXT to B.TXT would become:

       QUOTE SITE EXEC C:/DOS/COMMAND.COM COPY /B A.TXT B.TXT

You might notice the use of '/' instead of '\' in the path. This is because most command line FTP clients do not take kindly to backslashes. For Serv-U it doesn't matter and you can use '/' and '\' alike. You can also see that it is possible to specify arguments to a program as you would usually do, Serv-U passes them on to the program. The example above will likely only work for command line FTP clients that are based on the UNIX client. For point-and-click type clients, like WS_FTP you are on your own. They may or may not have the ability to send literal commands to the server but the way to get that done will vary from client to client.

Starting programs remotely may not be as straight forward as it seems, and can have varies side effects. Also, some programs need to be started via the command interpreter, COMMAND.COM, others not. In short: Some experimentation is recommended!

One more important thing about 'execute' access: Although Serv-U checks the program's path for access, the program you start this way is free to access and delete any file in any directory on your computer! Also, if you give 'execute' access to a directory then any program in the DOS path can be started! In short: **Be very careful with 'execute' access, as this is a very big potential security hole!**

2.19 How to Use Serv-U with 'SLIP/PPP Emulators'

More and more Internet Access Providers these days are offering Internet connections that are not 'real'. Instead of having their own IP address these connections run on top of the IP address of a UNIX host system. The general name for this category of connections is 'SLIP/PPP emulators' and a few of the popular programs that will do that are '**TIA**' (=The Internet Adapter), '**TWinsock**', and '**Pipeline**'. The reason ISP's use them is because they offer a cheap way to get many people connected to the net. They usually work fine for programs like Netscape, mail, and FTP clients. However, for servers they come with a number of build-in disadvantages.

Because of the way they have to do their work they have a number of peculiarities. Since the PC has to share the IP number with the host system you will generally not be able to use the default port 21 for Serv-U. Instead, you either have to use a high port number (generally above 8000) or set up 'port redirection'. Ask your Internet provider for details about the latter. A related problem is that the IP number displayed by Serv-U is, in this case, not the IP number that clients should use to contact your server. It is merely a dummy to keep the socket stack of the PC happy. For the outside world the IP number of your server is the same as that of the emulator host system, i.e. that of your Internet provider. Another special effect is that clients will not be able to use 'passive' mode for data transfers. This means that regular FTP clients will work OK, but WWW browsers like Netscape and Mosaic will not work when Serv-U runs via a SLIP emulator. For the same reason it won't be possible to use clients that are also connected via a SLIP emulator either.

Just in case you are interested I'll give you the technical details behind all this. Since Serv-U has to share the IP number with the Internet connection host system (Usually a UNIX system), the usual FTP port (number 21) will most likely be already in use by the host system. Because at a single IP address there is only a single port 21 this means either Serv-U or the host has to move to another port. The second side effect has to do with the socket stack: They usually need to know their own IP address to do the work, and to keep the stack happy it is installed with a non-existent IP number like 192.0.2.1 (Seems to be a popular choice). The FTP protocol does not use the server side IP number much, the only exception is the FTP 'PASV' command, which is used for passive mode data transfers (Which all WWW browsers use). In essence, what the PASV command does is ask the server for an IP address and port number where it will listen for the data connection. Since the FTP server does not know that its address is bogus, it will trustfully report this back. The client will then try to connect to this dummy address, which of course will never work.

2.20 How to use Netscape to access Serv-U

You might not be aware of it, but Netscape and other WWW browsers can do a whole lot more than just anonymous FTP access. The general syntax for FTP is the following:

>   FTP://<user>:<password>@<IP address>/<path/file>

Very little of this is needed and in its simplest form you only use the <IP address> which results in the usual anonymous access. If you add a <user> section (leaving out the password) then Netscape will prompt you for a password if needed. Of course, you can also specify the password by yourself by adding the <password> section. So far we did not use the <path/file> section and this means the user will end up in his or her home directory. If you want to go somewhere else you can tell Netscape to do this with the <path/file> section.

In the <path/file> section you should use '/' instead of '\', and you can add a drive letter as part of the path. So, to finish up with an example that uses all of the above:

> FTP://john:secret@ftp.cat-soft.com/f:/john

This would log John into the Cat Soft server.

By the way, did you know that the version 2.0 or later of Netscape can also upload files to a FTP server? To do so, simply drag the file-to-upload from 'Explorer' (the Win95 file manager), or from the 'File Manager' (in NT or Windows 3.1) to the Netscape window and drop it there. Experiment & Enjoy!

2.21 How to Let the Whole World into Your Server (and Delete All Your Files)

Even for those with a PC-oriented exhibitionist inclination Serv-U has something to offer. You can let the whole world into your server and have them delete all your files. To top it off, if set up correctly they will reformat your hard disk along the way too!

The easiest way to all that fun is to uncheck the 'Enable security' checkbox in the 'Setup Server' dialog box. This utterly and completely disables all security in the server and thus offers the most comfort to your FTP users. Not only can they freely copy and delete each and every file, they can also use the unique Serv-U feature that lets them start programs remotely. Of course, the fun is over once they remotely start FORMAT.COM, but what the heck, it was nice while it lasted!

Now, for those that are a bit 'sarcasm inhibited' I'll spell it out: **NEVER, NEVER, NEVER LEAVE THE 'ENABLE SECURITY' OPTION SWITCHED OFF IF YOUR SERVER IS CONNECTED TO THE INTERNET!!!** This might all seem a bit overkill, but believe it or not, I quite regularly get E-mail from people that tell me they are running Serv-U exactly like that on the Net. Now don't tell me I didn't warn you…

2.22 How to use multi-homed IP support

If your operating system supports it, and you have multiple IP addresses installed at your computer (Typically this is done in NT), you can let Serv-U respond differently for different IP addresses that the users connects to. To make it a little clearer a small example: Say you have two IP addresses assigned to your PC, 152.3.110.167 and 152.3.110.168, which correspond to IP names "ftp1.cat-soft.com" and "ftp2.cat-soft.com" respectively. Now you want anonymous users logging into the "ftp1" IP address to have access to different files and directories from those logging into "ftp2". This is called multi-homed IP, and supported by Serv-U.

The first step to using this is in the 'Setup Server' dialog box. Press the button labeled 'IP Homes' in there. Then use the screen that pops up to enter all the IP numbers of your PC. When this is done you can go to the 'Setup Users' dialog box, which will now show a new selection list with all your IP numbers. Just set up the users as you did before, and select the appropriate IP number that should be associated with that user name. This will allow you, for example, to make several users named 'anonymous', each responding to different IP addresses. That's all there is to using multi-homed IP, it could not be easier!

3. The Inner Workings

Before I go on to describe the settings of the SERV-U.INI file I want to spend a few words describing how Serv-U was made and how it goes about its job.

3.1 Serv-U Internals

The program was written using Borland C++ version 4.52 (for the 16-bit version) and version 5.0 (for the 32-bit version). To check for shaky pointers and catch all those resource leaks the program Bounds Checker version 3.01 from Nu-Mega was used. I think no serious Windows programmer should be without the latter, very highly recommended!
This whole project started a little over two years ago after much disappointment with the existing FTP servers for WinSock. In its current version it consists of a bit over 22000 lines of C++ code, divided into 41 C++ classes. The whole program was constructed from scratch, not using any existing FTP server code, and is tailored to MS-Windows and WinSock.

Internally, everything is very much compartmentalized, using a different class for different partial tasks. There is a WinSock class library, providing hi-level access to Windows Sockets and hiding all the nasty parts of dealing with them. It uses 100% asynchronous WinSock functions (which unfortunately do not always work correctly on a number of WinSock stacks) thus avoiding problems with multiple active sockets for a single task and re-entry. There is a FTP-manager class, taking care of listening for clients, and setting up instances of the FTP-command interpreter class when this happens. The latter does the actual interpretation of the FTP commands, talking to the security class for clearance and the WinSock class for communications. Then there are some utility-like classes, like those dealing with setup and logging. By having all these compartments that handle very well defined tasks, I hope to be able to easily extend this FTP server and fix those (hopefully few) remaining bugs quickly!

The 32-bit version of Serv-U is <u>not</u> multi-threaded, a single thread is used for all clients. 'Multi-threading' is one of those buzz words that are poorly understood by most and abused by many. 'Threads' are light-weight processes, that can be used to run concurrent tasks without all the overhead of creating real processes. In the mainframe world this has traditionally been used for servers: Each client would start a separate process or thread. However, unless you are running NT on a multi-processor machine (and I wager that now we are talking about <u>much</u> less than 1% of all NT installations), multi-threading is not generally going to gain you anything because thread maintenance and switching brings overhead with it. There are still good reasons for using multiple threads on a single processor machine: For independent background tasks, like a printer spooler, it makes perfect sense. Also, if a process needs to perform multiple concurrent tasks that each would take a long time (and 'long' in computer terms could mean anything over a few tens of milliseconds), using multi-threading is a good practice. However, for servers the practice of starting a separate thread for each client is usually to provide an easy way out of programming the thing. That way the programmer can use 'blocking' sockets and simply block (i.e. 'do nothing') until something happens on the socket stack. Makes for much simpler programs and is used for all the UNIX servers, so anything ported from there will be coded in this fashion. The better way to do things in Windows is to make the program event-driven and let it respond to messages from the socket stack instead of blocking the task. This means 'asynchronous' sockets have to be used, and the whole program is much harder to make: Instead of a single easy to follow flow the program

now becomes a number of message handler routines that can be called in any order. Serv-U only uses asynchronous sockets, and the response to each socket stack message normally does not take much time. Thus, a single thread can handle this perfectly well, and cutting it up in multiple threads would only slow things down.

3.2 The SERV-U.INI File

All the settings for the server, users, and groups are stored in a single file in text format. This file is always named SERV-U.INI. When the program is started it looks for this file in a number of different places: First the directory with SERV-U.EXE is checked. If no .INI file is found an environment variable SERV-U is checked. If this variable exists it should be set to the path where SERV-U.INI is found. Finally, if this variable does not exist, the whole DOS path is scanned, including the Windows directories. The first SERV-U.INI file found on the way is used. If after all the above no .INI file has been found, the program will create one in the directory of the Serv-U program. Use of the SERV-U environment variable and the DOS path makes it easy to set things up for network users where there is a single copy of the program but all the users needs their own settings. Serv-U uses the Windows built-in functions to read and write from/to the .INI file. A consequence of this is that the total size of SERV-U.INI can never exceed 64 Kbytes on Windows 3.1, 3.11 and Win95, limiting the number of users that can be set up. On NT there seems to be no limit to the .INI file.

We'll now go over all the items that can appear in SERV-U.INI. I will show you an invented setup file:

```
[GLOBAL]
Security=TRUE
PortNr=21
MaxNrUsers=15
MaxNrAnonymous=10
Invisible=TRUE
Logfile=c:\serv-u\logfile.txt
Logging=YES
TimeoutUser=600
TimeoutAnonymous=15
TryOut=Crippled
LogGETs=ON
LogPUTs=ON
LogSystemMes=ON
LogSecurityMes=ON
LogFTPCommands=OFF
LogFTPReplies=OFF
LogIPNames=ON
RegistrationKey=S%FgdfsdEvG,Rob Beckers,Cat Soft,1
Window=100,100,400,300
UserInfoWin=409,300
```

```
DirChangeMesFile=index.txt
DirChangeMesFile2=c:\serv-u\index.txt
LinkFile=link.txt
LinkFile2=c:\serv-u\link.txt
CheckAnonPass=OFF
Version=2.0.3.11
StartIconic=No

[SIGNONOFF]
SignOn1="Welcome to Robby's FTP-Server!"
SignOn2="It is %time on %date and you are user no. %unow"
SignOff1="Thanks for logging in!"
SignOff2="Hope to see you again soon . . ."

[IP-ACCESS]
DenyRulesFirst=YES
Bounce1=132.68.175.201
Bounce2=223.*.*.*
Allow1=132.68.176.53
Allow2=132.68.175.*
Allow3=101.43.23.30-40

[USER=Anonymous@2]
HomeDir=d:\anonftp2
Access1=d:\anonftp2\upload,WP
Access2=d:\anonftp2,RLP
LoginMesFile=logmes.txt
HideHidden=YES
RelPaths=YES
RatiosEnable=YES
Ratios=BytesPerSession
RatioUp=1
RatioDown=7
RatiosCredit=19968.000000

[USER=Anonymous@1]
HomeDir=d:\anonftp1
Access1=d:\anonftp1\upload,WP
Access2=d:\anonftp1,RLP
LoginMesFile=logmes.txt

[USER=Rob]
Group=system
Password=WdRx.Jlk0kemm%
HomeDir=c:\
AlwaysAllowLogin=YES
QuotaEnable=YES
QuotaMaxCurrent=10240000,89739
Access1=\,RWMCDLPE
Access2=lpt1:,WM
Access3=prn:,WM
Access4=aux:,WM
Access5=lpt2:,WM
Access6=lpt3:,WM
Access7=lpt4:,WM
```

```
        Access8=com1:,RWM
        Access9=com2:,RWM
        Enable=YES

        [USER=ALL]
        HomeDir=y:\
        Access1=y:\,RL
        Enable=NO

        [GROUP=SYSTEM]
        Access1=c:\system,RWDCMLEP
        Access2=d:\,RWDCMLEP
        Access3=y:\novell,RWDLP

        [RATIOS]
        Free1=*.txt
        Free2=\*\*.bat
        Free3=d:\gifs\sample.gif

        [IP-HOMES]
        IP1=152.3.110.168
        IP2=152.3.110.169

        [EXTERNAL]
        ClientCheckDLL1=CHKNOVELL.DLL
        ClientCheckDLL2=CHKNT.DLL
```

All but five of these settings can be changed and set interactively through the 'Setup' menus. The exceptions are the entries for 'Invisible', 'RegistrationKey', 'Window', 'UserInfoWin', and 'StartIconic', and if you really desire a user to have no password, that user's 'Password=' entry has to be set manually (i.e. nothing after the '=' sign to indicate 'no password').

The following paragraphs will describe each section and entry in more detail.

[GLOBAL]
All the settings related to the Serv-U program itself, i.e. the functioning of the FTP server and system functions, are found in the '[Global]' section. For the file formats of directory change message files and link files please see the appropriate 'How to …' section.

If security should not be enforced, the '**Security**' entry can be set to FALSE or 0. Doing so will leave the FTP server wide open to everybody!!! Default value for 'Security' is TRUE.

The '**PortNr**' entry determines the IP port that the server will listen on. Default value is 21.

To limit the maximum number of simultaneous users the '**MaxNrUsers**' entry should be set to the desired number. No entry or a negative number results in no maximum, only the number of available sockets will limit the number of users in that case. Similarly, the '**MaxNrAnonymous**' entry limits the maximum number of 'Anonymous' users. The value put here is only meaningful if it is smaller than that of the 'MaxNrUsers' entry.

For system managers that don't want their users to mess around with the server settings, it is possible to make Serv-U invisible by setting the '**Invisible**' entry to TRUE, or YES and put the Serv-U program in the 'startup' group. When this is done the server will not show up in the task manager list. One consequence is that there is no way to stop the program short of exiting Windows. Default is NO for this entry.

The '**LogFile**' entry should specify a full path and name for a logfile if logging is desired. There is no default logfile. To actually switch logging on and off the '**Logging**' entry can be set to ON or TRUE, or OFF or FALSE. Switching logging ON will only work if a logfile is specified. By default 'Logging' is set to ON.

The entries '**TimeOutUser**' and '**TimeOutAnonymous**' specify a time-out in minutes for respectively regular users and anonymous users. If a FTP connection is left idle for the indicated amount of time it is automatically closed. Filling in 0 results in no time-out. Default values are 15 minutes for anonymous and 10 hours for regular users.

The next one deals with the way the program can be tried out (when there is no registration code). '**TryOut**' can be 'Crippled' or 'Full'. The first allows a user to do a maximum of 5 GETs and 5 PUTs per session, switches the server off-line after 1 hour, and notifies clients that they are looking at an unregistered try-out version. However, it does <u>not</u> contact my permission server. The 'Full' option gives you no limitations while trying the program, it is exactly equal to the registered version. The downside is that it <u>does</u> access my permission server to ask for permission to run each time Serv-U is started. This is how the 30 days of try-out are enforced.

All the '**LogXXXX**' entries switch logging options on or off. Their names say it all, so I'll let you figure them out.

The '**RegistrationKey**' entry is used for entering the registration code. You get this code after registration. By default it has no value and for evaluation of the program it should be left blank or out of the .INI file.

The next entry is '**Window**' and this is set by Serv-U every time the program is stopped. It contains the last position and size of the program window, in the format 'top,left,width,height'. Likewise, '**UserInfoWin**' saves the window position of the user information window, in the format 'top,left'.

Related to this is **'StartIconic'** which stores if the program should start as an icon. This happens when the server was stopped while being iconized.

To enable directory change messages you have to set the '**DirChangeMesFile**' entry to a valid file name. This can be a complete path name, in which case the same file will be displayed for all path changes, or it can be a file name only, which means Serv-U will look for that file in the path the user is changing to. Likewise, '**DirChangeMesFile2**' is used in case the file specified in 'DirChangeMesFile' cannot be found.

If links should be available and displayed in the user's directory listings then '**LinkFile**' should be set to the file name to use. The file name format is the same as for 'DirChangeMesFile'. In case 'LinkFile' cannot be found the server looks for '**LinkFile2**'.

To check if anonymous 'passwords' resemble an E-mail address you can set '**CheckAnonPass**' to YES, ON or TRUE. Likewise, you can switch checks off by setting it to FALSE, NO or OFF, and this is also the default. In that case Serv-U will allow anything as an anonymous password.

Finally, '**Version**' keeps track of the Serv-U version number. Format is 'major,minor,revision,build no.' and it is used for automatic .ini file conversion.

[SIGNONOFF]
This section contains the messages that are displayed after a user contacts the FTP server and just before he disconnects. Every line has a separate entry with a number at the end, denoting the order. The sign-on message is put in '**SignOnxx**' entries (with xx the line number), and the sign-off message is put in '**SignOffxx**'.

There are several special character combinations recognized by Serv-U and they are expanded to their actual values when a user logs on or off. They all begin with '%' and you can find a complete list in the 'How to use sign-on and/or sign-off messages' section.

A tip: Keep the number of lines and the their length limited. Most FTP clients will mess up lines over 80 characters, and since a FTP reply code is tagged to the beginning of these lines before they are sent, it is wise to keep them to less then 75 characters.

[IP-ACCESS]
This section determines which client IP-numbers will be allowed access to Serv-U. There are two kinds of rules: Those that refuse access in the form of '**Bounce**' entries, and those that grant access using '**Allow**' entries. If this section doesn't exist, or no entries are found, then all clients are allowed to contact the server. The reverse is also true, if there is even a single entry ('Bounce' or 'Allow') then only those clients will be allowed to contact the server that pass the rule. All entries are numbered sequentially ('Allow1', 'Allow2' etc.) and they are evaluated according to their number from first to last. Numbers should be consecutive. The 'Bounce' rules are evaluated before the 'Allow' rules.

The IP-number of the client is matched section by section to each rule until a match is found. If the matching rule is one of the 'Bounce' rules, the client is disconnected. If the IP number matches an 'Allow' rule then he can proceed. The rules can be exact IP-numbers, or contain special characters. There are two of those:

    *   = wildcard, match any number
    -        = denotes a range

A quick example: The rule '132.*.76.48-89' will allow entry to clients with an IP-address starting with 132, the second section can be anything (0..255), the third <u>must</u> be 76 and the last section should be between 48 and 89 (limits included).

The **'DenyRulesFirst'** flag is used to specify in which order the access rules should be evaluated: The 'deny' rules first and then the 'bounce' rules, or vice versa. Default this is set to YES.

[USER='Name']
The information about a user is stored in this section, 'Name' stands for the user's name. Each user has a separate section. It contains information needed to authenticate a user during login, and rules determining what this user is allowed to access. The Serv-U program will first check this section for a regular user. If no applicable information is found and the user is a member of a group, the group is addressed for the same information. If the result of this is still undetermined, the special user name 'ALL' is searched.

Now to the entries that can be found in this section. The identity of a user is verified by comparing his password, after encryption, with the one in the '**Password**' entry. The UNIX 'crypt()' command is used to encode the passwords. This makes it possible to extract users with their password from the PASSWD file of a UNIX system, the same passwords should work on both systems. Unfortunately, there is not a single standard for password encryption on UNIX systems these days. Serv-U uses the most common scheme, but this might not work for your system.

If the password matches, the home directory of the user is taken from the '**HomeDir**' entry. This should always be a full path name, including drive letter!

To make a user a member of a certain group, the '**Group**' entry can be used. All information needed and not found in the user's section; password, home dir and file/directory access, are then looked for in the group's section.

Information about file and directory access for a user is stored in the '**Access**' entries. Each of these is numbered, and access information is checked in order: first comparing it to the first rule, then the second, etc. The numbering must be consecutive. Access rules start with a path or file name. These paths are usually full names, including drive letter. If the drive letter is missing, they apply to all drives. If different settings are needed for a subdirectory, then a rule with that directory should appear before its parent, i.e. with a lesser number (As a consequence of the order in which Serv-U evaluates access rules). The path in an access rule is followed by a comma and the access information itself. This can be a combination of up to eight different characters:

> R   = read access to files
> W   = write access to files
> M   = modify access to files (implies write access)
> E   = execute access, for remotely starting programs
> C   = right to create subdirectories
> D   = right to delete subdirectories
> L   = directory list access
> P   = the rule 'propagates' to sub directories as well

It is entirely possible to have no access information at all (only a path). This means that the user will not have any access to that path. One thing to realize is that write access to a file does <u>not</u> imply read access! As you can see it is also possible to specify access rights for the parallel and serial ports. They are part of the regular security scheme and to transfer to or from a port a user needs access rights. Then finally, the path in an access rule does not have to point to a directory. It is also possible to specify a filename. Of course, the 'C', 'D', 'L', and 'P' rights will not have any meaning then.

There are two special user names: 'Anonymous' and 'ALL'. If there is an user 'Anonymous', it will be possible to log into the server without a password. Instead, Serv-U will ask for the user's E-mail address and log this. Most of the regular entries apply for 'Anonymous' as well, except 'Password' and 'Group', these are ignored. In fact, for anonymous users the sections for groups and 'ALL' are never searched.

The user 'ALL' is searched if no appropriate rule is found in a user's or the user's group's entry. It can contain any of the regular entries.

The user account can be disabled by setting **'Enable'** to NO. By default accounts are enabled.

'**RelPaths**' determines if users should be treated with all path names relative to their home directory. This is desirable for use with WWW browsers that insist on having access to the root directory ('/'). Switching this on limits users to the sub-tree of their home directory, they cannot switch to other drives. If you switch it off then users will see their full path. Default is switched off.

The '**LoginMesFile**' entry is used for pointing to a login message file that should be displayed to the user upon login. The same file name rules apply as for the 'DirChangeMesFile' entry, and you can find more details about its usage in the 'How to setup user specific long messages' section.

The user heading can contain an added '@' followed by a number, like '[USER=Anonymous1@1]'. This indicates that the user name is associated with a particular IP number for multi-homed IP support. A list with IP numbers and their index can be found in the '[IP-HOMES]' section.

In case **'HideHidden'** is YES or TRUE all the files with the 'hidden' attribute will not show up in directory listings. Default this is NO. In case a user should always be allowed to log in, even when the user limit is reached, the **'AlwaysAllowLogin'** setting should be TRUE or YES. Default for this is NO.

The UL/DL ratio settings are also part of the user setup info. To enable the use of ratios for a user the **'RatioEnable'** setting should be set. The type of ratios Serv-U will use is controlled by 'Ratios': The four options are **'FilesPerSession'** which counts files for each session, **'BytesPerSession'** which counts bytes for each session, **'FilesOverall'** to count files over all sessions, and **'BytesOverall'** which is for counting bytes over all sessions. The **'RatioUp'** setting specifies the upload part of the upload/download equation, and **'RatioDown'** is for the download part. The credit is stored in **'RatiosCredit'**, depending on the type of ratios used this can be in 'files' or 'bytes', and it can be a preset value for each session or the current credit in case ratios are counted over all sessions.

In case the user's disk usage should be limited, disk quota limitations can be switched on by enabling **'QuotaEnable'**. The **'QuotaMaxCurrent'** value keeps track of the maximum and current disk usage values in bytes.

 [GROUP='Name']
These sections contain the group info. The entries here are exactly the same as those for a user,
except that the 'Group' entry has no meaning of course.

[IP-HOMES]
This section lists all the IP numbers that Serv-U should use for multi-homed IP support. Each line
couples an index number with a corresponding IP number of the form '**IPx=yyy.yyy.yyy.yyy**'. In
this 'x' is the index number (any positive number between 1 and 30000), and 'yyy…' is the IP
number that should be associated with it.

When a user name should be coupled to a particular IP home, the index number for that IP home is
appended to the user name, preceded by a '@' character.

[RATIOS]
This is an auxiliary section for the use of UL/DL ratios. It allows for specifying a list of files which
should not be counted by the ratio system. Each line starts with **'Free'** followed by a sequential
number. The information after the equal sign is the path or file name that should be discounted.

[EXTERNAL]
This heading denotes a list of dynamic link libraries which Serv-U should use to very user access.
In case the Serv-U user database does not provide an answer to access requests the inquiry is passed
on to the first DLL in the list, if this DLL does not have an answer it is send to the second DLL, and
so on.

Each line in this section has entries of the form **'ClientCheckDLL'**. These are followed by an
order number and specify the file name of the DLL to use. The DLL file should be either in the
Serv-U program directory, somewhere in the DOS path, or in the Windows directory for Serv-U to
find it.

3.3 Using External User Access Verification DLLs

Serv-U can use an external DLL to verify client access and retrieve information like a client's home
directory etc. If one of more external DLLs are specified and Serv-U cannot find the appropriate
information internally it will question each of the external DLL's in turn. This can be used to create
an interface to external user databases, which can then control FTP access. If the appropriate DLL
exists, for example the NT build-in user database could be used, or a Novell user database.

Setup
To make Serv-U use external DLLs for client access verification you need to add the DLL names to
the SERV-U.INI file. At the moment there is no interactive user setup to do this, so the ini file has
to be edited directly. There can be more than one DLL, and Serv-U will query them in the order
specified until one of the DLLs signals that it had the required information.

The DLL names need to be added to a section named '[EXTERNAL]'. The format is as follows (for
example):

```
[EXTERNAL]
ClientCheckDLL1=CHKNOVELL.DLL
ClientCheckDLL2=CHKNT.DLL
        .
        .
```

The file names can be either full path names, or file names only. If the full path is specified then Serv-U will only look at that path. If only the file name is given Serv-U will first look in the program directory, then the current directory, the entire PATH, and finally in the Windows directories.

DLL Specifications
The DLL entry point for Serv-U needs to be the following function:

```
int HandleClientEvent(RClientEventStr* pEventStruc)
```

Please note that the function name is case sensitive!
The function should return TRUE (=1) if it handled the event and does not want it to be passed on to the next DLL. It should return FALSE (=0) in case it didn't handle the event.

The RClientEventStr structure is defined as follows:

```
struct RClientEventStr {
  int Event;              // event code
  int Flag;               // flag, meaning depends on event
  char User[40];          // user name
  char Aux[512];          // auxiliary area, usage depends on event
  char HostIP[16];        // server IP home
}
```

The 'Event' code determines the nature of the request and can have the following values:

```
#define SRVU_LoginMesFile    1      // get login message file
#define SRVU_HomeDir         2      // get home dir
#define SRVU_Password        3      // verify password
#define SRVU_IPAccess        4      // verify IP access
#define SRVU_WriteFile       5      // verify write access
#define SRVU_ReadFile        6      // verify read access
#define SRVU_ModifyFile      7      // verify mod./del. file access
#define SRVU_ExecProg        8      // verify execute access
#define SRVU_ListDir         9      // verify dir listing access
#define SRVU_ChangeDir       10     // verify dir change access
#define SRVU_DeleteDir       11     // verify dir delete access
#define SRVU_CreateDir       12     // verify dir create access
#define SRVU_HideHidden      13     // get setting for 'hide hidden files'
#define SRVU_RelPaths        14     // get setting for 'relative paths'
#define SRVU_RatioType       15     // get setting for type of ratios
#define SRVU_RatioDown       16     // get setting for download ratio
#define SRVU_RatioUp         17     // get setting for upload ratio
#define SRVU_RatioCredit     18     // get/ajust ratio credit setting
#define SRVU_RatioFree       19     // verify if file is free for ratios
#define SRVU_QuotaEnable     20     // verify if disk quota is enabled
#define SRVU_QuotaChange     21     // change in disk quota
```

```
            #define SRVU_QuotaMax           22     // maximum disk quota
            #define SRVU_AlwaysLogin        23     // allways allow login
```

Event Details

SRVU_LoginMesFile
    On entry: User = user name
    On return: Flag = TRUE (=1) if file name was found, FALSE (=0) otherwise
           Aux = file name of login message file, if one was found

SRVU_HomeDir
    On entry: User = user name
    On return: Flag = TRUE if home dir was found, FALSE otherwise
           Aux = home dir (full path), if one was found

    Note: In case a user account is disabled 'no home dir' should be returned.

SRVU_Password
    On entry: User = user name
           Aux = password the user entered
    On return: Flag = TRUE if password was correct, FALSE otherwise

SRVU_IPAccess
    On entry: User = user name if available
           Aux = IP address of client, in text format
    On return: Flag = TRUE if access is allowed, FALSE otherwise

    Note: If no user name is present it means the client just   connected and a check should be made for server-wide IP access. If a user name is present access should be checked for that particular user only.

SRVU_WriteFile
    On entry: User = user name
           Aux = full path of file
    On return: Flag = TRUE if access is allowed, FALSE otherwise

SRVU_ReadFile
    On entry: User = user name
           Aux = full path of file
    On return: Flag = TRUE if access is allowed, FALSE otherwise

SRVU_ModifyFile
    On entry: User = user name
           Aux = full path of file
    On return: Flag = TRUE if access is allowed, FALSE otherwise

SRVU_ExecProg
    On entry: User = user name
           Aux = full path of program name to be executed

On return: Flag = TRUE if access is allowed, FALSE otherwise

SRVU_ListDir
    On entry: User = user name
                Aux = full path of directory
    On return: Flag = TRUE if access is allowed, FALSE otherwise

SRVU_ChangeDir
    On entry: User = user name
                Aux = full path of directory
    On return: Flag = TRUE if access is allowed, FALSE otherwise

SRVU_DeleteDir
    On entry: User = user name
                Aux = full path of directory
    On return: Flag = TRUE if access is allowed, FALSE otherwise

SRVU_CreateDir
    On entry: User = user name
                Aux = full path of directory
    On return: Flag = TRUE if access is allowed, FALSE otherwise

SRVU_HideHidden
    On entry: User = user name
    On return: Flag = TRUE if request was handled by DLL, FALSE otherwise
                Aux =   Aux[0]=1 if server should hide 'hidden' files
                        Aux[0]=0 if server should show 'hidden' files

SRVU_RelPaths
    On entry: User = user name
    On return: Flag = TRUE if request was handled by DLL, FALSE otherwise
                Aux =   Aux[0]=1 if server should use 'relative paths'
                        Aux[0]=0 for using full path names

SRVU_RatioType
    On entry: User = user name
    On return: Flag = TRUE if request was handled by DLL, FALSE otherwise
                        if U/D ratios should be disabled FALSE is also returned
                        Aux = set to the textual ratio type, as used in the INI file ie.
                                "FilesPerSession" = count files for each session
                        "BytesPerSession" = count bytes for each session
                                "BytesOverall"   = count bytes over all sessions
                                "FilesOverall"   = count files over all sessions

SRVU_RatioDown
    On entry: User = user name
    On return: Flag = TRUE if request was handled by DLL, FALSE otherwise
                Aux = set to the textual representation of the download ratio
                        (an unsigned 32-bit number)

```
SRVU_RatioUp
```
On entry: User = user name
On return: Flag = TRUE if request was handled by DLL, FALSE otherwise
            Aux = set to the textual representation of the upload ratio
                    (an unsigned 32-bit number)

```
SRVU_RatioCredit
```
On entry: User = user name
            Aux = set to the textual representation of the value to add
                    to the credit (a floating point number, can be negative)
                    set to "0" if the request is to obtain the current value
On return: Flag = TRUE if request was handled by DLL, FALSE otherwise
            Aux = set to the textual representation of the (new) credit
                    value (a floating point number which can be negative)

   Note: All 'credit' calculations should be done with sufficient precision. In C a double is recommended.

```
SRVU_RatioFree
```
On entry: User = user name
            Aux = set to the path/file name of the file to check if
                    it should be downloadable for free
On return: Flag = TRUE if file is free, otherwise FALSE

```
SRVU_QuotaEnable
```
On entry: User = user name
On return: Flag = TRUE if request was handled by DLL, FALSE otherwise
            Aux =   Aux[0]=1 if disk quota limitations are enabled
                    Aux[0]=0 if no disk quota limitations

```
SRVU_QuotaChange
```
On entry: User = user name
            Aux = set to the textual representation of the value to change
                    the current quota with (can be negative, decreasing available quota)
                    set to "0" if the request is to obtain the current value
On return: Flag = TRUE if request was handled by DLL, FALSE otherwise
            Aux = set to the textual representation of the (new) credit
                    value (a number which should always be positive).

```
SRVU_QuotaMax
```
On entry: User = user name
On return: Flag = TRUE if request was handled by DLL, FALSE otherwise
            Aux = set to the textual representation of the maximum disk quota amount.

```
SRVU_AlwaysLogin
```
On entry: User = user name
On return: Flag = TRUE if request was handled by DLL, FALSE otherwise
            Aux =   Aux[0]=1 if user should always be allowed to log in
                    Aux[0]=0 if user should bounce if max. no. of is reached

4. Getting In Touch - Bugs & Registration

If you have questions or want to offer suggestions you are most welcome to drop me a line. However, please check out this manual and the Cat Soft Web site first to see if the answer to your question is already in there. Although I truly love to hear from you and will do my best to help, the problem is that each day between 30 and 50 E-mail messages arrive here, and you are talking to a one-man, spare-time company! As it is I'm spending 2-3 hours a day on answering E-mail and that is about as much as I have to spare. The easiest way to get in touch with me is via E-mail. The address is:

RB5@acpub.duke.edu

The Cat Soft Web site can be found at:

http://www.cat-soft.com

Regular mail should work as well, but might take a bit longer. My address for this is:

Rob Beckers
210 Alexander Ave., Apt. D
Durham, NC 27705
USA

4.1 Serv-U Mailing List

There is also a WWW mailing list for Serv-U where you can ask questions and discuss Serv-U related topics. To subscribe to the Serv-U mailing list, send an E-mail message to:

list@corridor.com

with only a single line in the message body, which should read:

subscribe Serv-U

This will return a message from the list manager within a minute or so, explaining in detail how to post messages to the list and related matters. I am also subscribed to this list, so each message you post will reach my E-mail box as well. When needed I'll respond to your message, but my hope is that others will reply to, thus cutting down a bit on the flood of messages I have to answer each day.

If you want to look through previous postings to the mailing list (an invaluable source of information for solving problems), please take a look at:

http://www.pacemail.com/hyper/

4.2 Reporting Bugs

Nothing in this world is perfect, least of all me! Alas, chances are that despite careful testing you'll still find a bug. Please don't think others will report it, let me know! There are a few things I need to know in order to improve chances of fixing the beastie, so take note of the following:

- Most important: Can you get the same bug to appear by repeating certain actions? Please try hard; without a recipe for repeating a bug, it's going to be very hard to track it down.
- What TCP/IP and WinSock stack are you using? Brand/type and version number please. What operating system (DOS version and Windows or Windows-For-Workgroups, Windows 95, NT version x.xx)? Any memory manager (QEMM etc.), what version?
- Please indicate also if this bug is merely cosmetic or of vital importance for using Serv-U. Somewhere in between is possible as well of course. By the way, I consider security related bugs <u>very</u> important!
- Finally, please give me a chance to fix a bug, before you start to shout all over the Internet how bad this program is . . .

4.3 Registering Serv-U

If you're happy with the performance of Serv-U, then please make me happy and register this program! Just a few words for those who are in doubt: Making this program took me (very literally) many months of work, spread out over the last three years. Your registration fee is going to motivate me to continue improving Serv-U. In general, registration is important for shareware programs: It makes it possible for you to use professional quality software for peanuts. Lastly, being a biomedical engineering graduate student, I'm not exactly making lots of $$'s (to put it mildly). So, those 25 bucks for registration mean a lot to me!

What do you get if you register? As soon as I get your registration I'll send you a registration code plus instructions on how to add it to the program. This will enable you to use the program after the 30 day try-out period. Please let me know your E-mail address, since Serv-U is tailored to pick the registration key out of the E-mail message I will send you. In case I have your E-mail address you'll also get notified when there are updates. Once registered you'll get those updates for free. That is, you can use the same registration code on the updates, but you'll have to get them yourself. Apart from all this you'll also get the nice, warm feeling of having contributed to improving my earthly wealth!

The registration code is tied to the user/company name you specify on the registration form. You can see if your server is registered by looking at the 'About' dialog box: If registered it will tell you to whom. Another thing to keep in mind is that the registration information is sent back to any FTP client who uses the FTP command 'HELP'. This is not a much used command but in principle it allows the whole world to find out who paid for your copy of Serv-U. If you're the lawful owner of the server this shouldn't bother you, if not…

The registration fee is $25 for each copy. If you want to make me even happier and need several copies, the following license prices apply:

The FTP Serv-U license prices:

1-9 $25 per copy

License for:
20      $250
50      $500
100     $700
100+ $500 per block of 100 copies

Licenses for more than 500 copies are negotiable.
Educational discount: For licenses of 50 or more copies: 30% discount.

The 'number of copies' in the above refers to the number of concurrently running Serv-U FTP servers in your organization (i.e. it is independent of the number of concurrent FTP clients). The last page of this manual is the registration form. Please use this form for all registrations! That way I can keep my administration manageable.

4.4 Registration in the US

To register from within the US, please fill out the registration form and send it along with payment to the address on the form. A (personal) check made out to Rob Beckers is the preferred form of payment, but a Postal Money Order, or Purchase Order is welcome too. If you need a receipt, then please mention this on the registration form and I will send you one by paper mail. It is also possible to register via CompuServe's SWREG service. The registration ID for that is 7743 and this costs $30 (Since CS takes $5). Credit cards are welcome too! Accepted are Visa, Master Card, Novus/Discovery, and American Express. I'm working on on-line registration, but meanwhile you can either send me the card number and expiration date by E-mail (please include the registration form), or drop me a line asking for my fax number, so you can call the details in.

IMPORTANT NOTE:
Make sure to include the registration form with whatever you send me!! I'm receiving way too many checks and purchase orders without a form, often even without a name nor a phone number. If your order goes through a purchasing department then please make sure they include the registration form with the PO or check!

4.5 Registration from Abroad

While borders are disappearing in many parts of the world, getting money across from one country to another is still not easy, at least not when you want to keep the cost down. Below is a list of payment options that I accept (in order of preference). Please make sure to include the registration form with anything you send me.

By far the easiest way to get money across the borders is by **credit card**. Accepted are Visa, Master Card, Novus/Discovery and American Express. On-line registration is in the works, but meanwhile you can either send me the card number and expiration date by E-mail (please include the registration form), or drop me a line asking for my fax number, so you can call the details in.
Other payment options for registration from abroad are:

- Using **CompuServe's SWREG** service. The registration ID for Serv-U is 7743 and registration costs $30 this way (CS takes $5, so I end up with $25).
- By **check**, drawn at an American bank and in US dollars. The check should be made out to Rob Beckers.
- By **American Express Travelers Checks** for the correct amount in US dollars. These are cheap and safe, but there might be a minimum commission charged by your bank. The checks should be made out to Rob Beckers and don't forget to sign them twice!
- By **Postal Money Order**. As I understand it, you can buy these international money orders in most countries for very little money ($3 here in the US). Payment is in your own currency, but the money order should be made out for US $25 and to Rob Beckers.
- By **cash**, but only in US dollars and I give no guarantees about safe arrival! Please do not send me other currencies, it would probably cost me much more to convert them to US dollars than it will cost you. A trick I found useful for sending cash in envelopes: put the money in a folded sheet of paper so it doesn't shine through the envelope. This might improves chances of arrival considerably.
- **For the Dutch**: Het is ook mogelijk te betalen door fl. 40 over te maken op girorekening 53.95.461 t.n.v. Rob Beckers te Bunde. Stuur dan wel even het registratie formulier per E-mail naar mij toe. S.v.p. geen geld vanuit Belgie of Duitsland rechtstreeks overmaken, want daarvan blijft niet veel over!
- For **Canadians** only: By any type of (personal) bank check of Canada. Just make the check out in US dollars. Doing so is easy: Write "US" in front of the dollar sign and add "US Funds" below the sum. This way neither of us pays anything extra (I don't, and as far as I know you don't either, but check with your bank if you want to make sure). The check should be made out to Rob Beckers.
- By **direct money transfer** in US dollars to my American bank account. Please add $13 extra, since that is the fee my bank charges me to receive money this way. Please send me the registration form by E-mail in this case. The details for a wire transfer are:

    Bank:
    Wachovia
    Durham - North Carolina
    ABA/Routing no:
    053100494
    Account:
    No. 2373193064
    Rob Beckers
    210 Alexander Ave., Apt. D
    Durham, NC 27705
    USA

Rob Beckers
210 Alexander Ave., Apt. D
Durham, NC 27705
USA

Registration Form Serv-U

Name: ....

Company Name: ....

Address: ....

....

....

Phone Number: ....

E-Mail Address: ....

(Important! Reg. Code is sent by E-mail)

No. of copies: ....

Additional comments, suggestions, complaints, praise, etc …

....

....

....

Registration fee is **$25 per copy**. Send this order form along with your payment to: Rob Beckers, 210 Alexander Ave., Apt. D, Durham NC 27705, USA.

If you have any questions, comments or suggestions please contact Rob Beckers at the above address or via E-mail to RB5@acpub.duke.edu. Check out the site license prices in the manual if you need multiple copies.

As this software is shareware it comes 'as is', there is no warranty implied or otherwise, nor is support provided. However, if you discover any bugs or problems please contact the developer at the above e-mail address.