

FTP Serv-U

FTP-Server Daemon for WinSock

Version 1.1

© 1995 Rob Beckers

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, unfortunately it is 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

µIntroduction.....	1
1. Making It Work - Installation.....	3
1.1 Installation.....	3
1.2 De-installation.....	4
1.3 Upgrading.....	4
1.4 Known Bugs & Problems.....	4
1.5 Changes Since Version 1.00.....	5
1.6 A Word to the Wise (but paranoid).....	6
2. The Grand Tour - Menus.....	7
2.1 The File Menu.....	7
2.2 The Edit Menu.....	7
2.3 The Setup Menu.....	7
2.4 The Help Menu.....	14
2.5 The Logfile and Screen.....	14
3. The Inner Workings.....	15
3.1 Serv-U Internals.....	15
3.2 The SERV-U.INI File.....	15
4. Getting In Touch - Bugs & Registration.....	21
4.1 Reporting Bugs.....	21
4.2 Registering Serv-U.....	21
Registration Form Serv-U.....	24

Introduction

Thank you for giving this program a try!

With FTP Serv-U your PC will be turned into a FTP server. This means that others on the computer network that you are connected to 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 communicating between computers, so that many different types of computers, using different operating systems and file formats, can exchange information.

FTP Serv-U is a 'server' program and/or daemon. The term daemon comes from the 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 about a year 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 a year and over 11000 lines of C++ later you're looking at the result!

So what has this FTP server to offer?

- 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).
- 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, and very strict adherence to the FTP standard (found in document RFC 959). 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.
- 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).
- A very cute icon!
- Compared to the commercial implementations, Serv-U is dirt cheap!

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 it 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!

Before I forget it: Thanks, Lars, Kyle, Arend, Michael, Ryan, Yiannis, Jozsef, Rick, and Brad for beta testing and turning my attempt at English into the real thing. And, Alun, I hope I didn't shock you too much by bringing this program out. You can't say I didn't warn you though...

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.

Serv-U comes with the following files:

SERV-U.EXE	- The FTP-server executable itself
SERV-U.DOC	- The documentation in MS-Word format
SERV-U.TXT	- The documentation in ASCII format
README.TXT	- Something you have to read
BWCC.DLL	- The Borland Custom Control library that creates the 3D-look
REGISTER.TXT	- A registration form in ASCII format
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 only fine point you might want to pay attention to is whether or not you already have the file BWCC.DLL in your Windows directory. If so, you can delete the one in the Serv-U directory, but make sure it is the same version (compare the sizes)!

Serv-U offers two very distinct ways to try it out. Both with their 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 (my PC in fact) 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 this time, 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, 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 not contact another system over the network.

Just to make sure this is clear: Once Serv-U is registered it will NEVER send anything over the network other than what is needed for regular FTP traffic!

When you run Serv-U for the first time there will be no users and access will be restricted. To change this, go through the 'Setup' menu items and put in your heart's desires. I advise you to take a look at the next chapter, explaining the various menus, since the security setup is simple but not totally intuitive (sorry for that, but . . .).

For network use with a single executable shared between many users that need their own settings, 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 this file. 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 searched for and created if necessary in the default program directory.

1.2 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 not change any system files and does not place any files in other directories.

1.3 Upgrading

Upgrading from an earlier version of Serv-U is not a big deal either: Just copy the new .EXE file over the old one and you're all set. Your old SERV-U.INI file will be used by the new program so you don't have to re-install anything.

1.4 Known Bugs & Problems

At the time this documentation is being written there are no known bugs left in the program. Everything that was found in the 1.0x versions has been fixed. There are however still a number of known 'problems', even if they are not bugs as far as I can tell. Let's go over them:

- If you are using FTP Software Inc.'s TCP/IP software and WinSock stack, make sure you have version 2.3 of their software and v1.15.1 of their WINSOCK.DLL (or anything more recent). Older versions of their socket stack will give you a truck load of problems, and not just with Serv-U. Their latest WINSOCK.DLL is available via anonymous FTP from ftp.ftp.com.
- According to my information, Serv-U does not work well with Sun's PC-NFS WinSock stack, nor with Chameleon's stack. As far as I could trace the problems it seems that in both cases the socket stack is to blame, it is simply not WinSock v1.1 compliant. If you do get it to work (a new version may come out that solves the problems) then please let me know so I can pass the advice on to others.
- Netscape works well with Serv-U, but it seems never to close the connection when getting a directory listing, i.e. the big blue 'N' keeps 'pumping'. This is definitely a Netscape bug. I've contacted the company about it. Just ignore this behavior, it does not affect the actual operations.

1.5 Changes Since Version 1.00

Many little things were changed since v1.00. Most of you will not notice the majority of the changes, but some of the them were needed to fix pretty Big Bugs. Below is a list of what was done:

Version 1.1, released 19 March 1995:

- Fixed some spelling errors in messages. Fixed logging to screen for time-out messages. Added log message in case limit of nr. of users is reached. Added log message when server is (re)started.
- Added lots of logging.
- The SYST command now replies with the code for a UNIX system. This is because some clients use it to determine the format of directory listings.
- Time-out values for idle/hung connections are now part of server setup.
- Drastically increased packet time-out for data transfer, now set at 5 minutes (was 30 seconds). Should be sufficient to allow transfer even on bad connections.
- Log messages for failed data transfers now have a specification showing why.
- Fixed bug that caused path for anonymous users with root as home directory to be reported without a '\' at beginning. The same bug caused absolute paths in CWD to be processed incorrectly.
- Changed the HELP response to make WS_FTP work properly with Serv-U.
- Added support for transfer to/from ports (PRN: AUX: LPTx: and COMx:).
- Made a work-around for FTP Inc.'s WinSock stack. This stack does not handle SO_LINGER properly on closing a socket, causing 'data channel in use' errors.
- Fixed bug that caused random truncation of PUT files in combination with some clients.
- Fixed bug that allowed users to get 'dir' listings for paths with explicitly no access set to them.
- Fixed bug causing 'dir' with absolute path name to go wrong.
- Changed response messages to file transfers, only the filename is shown now, not the path name.
- Added a retry period for the server to come online. This should solve problems with socket stacks that do not allow to re-use a port immediately after closing it.
- Changed the timing of the '150-' response message for PASV transfers. It is now sent after the data connection is established instead of at the time of a transfer command.
- The listening socket will now automatically be restarted when killed by the socket stack. Some stacks kill listening sockets without reason (Trumpet for one).
- Fixed a bug that made RMD (=remove directory) fail if the directory was on a drive other than the active drive.
- Username 'FTP' is now synonymous to 'ANONYMOUS'.
- Fixed bug in very long directory listings (>64K data).
- Clients that connect but never log in are now kicked off the system after 5 minutes.
- User can now select the try-out method: Fully functional with contacts to my permission server, or, crippled but no permission server contacts.
- Installed selectable path mechanism for anonymous: Either absolute paths (like a regular user) allowing for drive changes, or paths relative to the home directory (needed for WWW browsers).
- Changed registration key to work with user/company name instead of IP number. Every time Serv-U is started it tries to read the key from a file KEY.TXT. Registered version displays the key in the "About" box and in reply to the FTP HELP command.
- Changed the RETR and STOR replies (used for GET and PUT). They are now conform the average UNIX system. This makes WS_FTP more happy, so it shows a progress bar while downloading.

Version 1.00:

- Initial release 7 February 1995

1.6 A Word to the Wise (but paranoid)

Many a word has been spent on the alt.winsock newsgroup in recent weeks 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 (my PC in fact) to determine if it should be allowed to run or not. That is the way the 30 days of try-out are enforced. To that effect, information is exchanged between your PC and mine, and as people remarked: 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 him/her (all 11000+ 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 full moon and Venus coincide, then monitors PC activity and if a user hasn't been present for a few hours, or if it's 4 in the morning, will start sending over the whole hard disk to unknown destinations. This kind of thing is not 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!

2. The Grand Tour - Menus

The menus and associated dialog boxes have been designed to be as simple as I could make them while still providing the needed features and flexibility. The next paragraphs take you on a grand tour through all of them. I highly recommend you to take a closer look at the part about setting up users and security under 'Setup - Users/Groups'. The default settings have been chosen to be both secure and reasonable. Make sure you know what they are about before changing them.

2.1 The File Menu

The 'Logging' option can be checked to enable logging of FTP events to a file. If unchecked, logging will be to screen only. This option is only available if a logfile has been specified under the 'Setup - Logging' menu choice. Changing logging through the 'File - Logging' menu will only affect the current session. The changes are not saved and the next time you start Serv-U logging will again be as specified in 'Setup - Logging'. This can be convenient if, for example, you want to temporarily switch logging on/off for testing purposes.

The other option is 'Exit', guess what that does . . .

2.2 The Edit Menu

You'll find only two options here: 'Copy' and 'Clear'. The first one copies selected text from the Serv-U logscreen to the clipboard and 'Clear' wipes the logscreen clean.

2.3 The Setup Menu

This is where the fun starts. Almost everything concerning the functioning of Serv-U can be set through the 'Setup' menu. There are two exceptions: First, if you insist on allowing access for a user without a password, and, second, if you want to make the Serv-U program invisible; meaning it will not show up in the list of the task manager. In both cases you'll have to edit the SERV-U.INI file directly. For more information on how to do this, take a look at the 'Internals' chapter.

The dialog boxes where you can fill in your settings have not been made totally bullet proof. It is entirely possible to enter nonsense in certain items and the program will accept it (like path names that don't exist, etc.). Of course, things might not work exactly as you're expecting them to, but it should not cause the program to crash. The bottom line is that it is your responsibility to provide settings that make sense.

Now, let's continue with the 'Setup' menu choices and associated dialog boxes.

The 'Setup - FTP-Server' option

This menu choice will lead you to a dialog box containing matters directly related to the workings of the FTP server itself.

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. Of course, the rest of the

world expects a FTP server to listen on port 21, but changing to another number is one great way of insuring that only you and some selected friends will know about your server. One fun choice is to set this to port number 23 and then use a telnet program to 'telnet' to your own PC.

The next item is the 'Maximum number 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. 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 number of anonymous' limits the number of 'Anonymous' users at any given moment. If 'Maximum number 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 the Microsoft 32-bit stack, which has the tendency to kick the system back to MS-DOS without any warning when heavily loaded. For my own server I have set the limit to 10 simultaneous users. Since I mostly get relatively slow long distance traffic this means in practice that I have never noticed any slowdown in system response while using it for other work and the server has remained stable. If you use a dedicated PC as a FTP server it is probably safe to go to about 20 simultaneous users.

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 NOT EVER LEAVE THIS OPTION UNCHECKED IF YOUR PC IS CONNECTED TO THE INTERNET!!!**

The last item is the 'Relative paths for anonymous' checkbox. By default this is checked and it causes anonymous users to see all directories and path names relative to their home directory. So, if your anonymous users have '\USERS\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 WWW browsers need read 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 do this 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 'Setup - IP-Access' option

This dialog box 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 deny rules are looked at **FROM TOP TO BOTTOM**. If no matching rule is found, the allow rules are evaluated, again 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 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.

The 'Logging' option

This dialog box 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. If you are interested in the exact format of the logfile, it is described in detail further on in this manual.

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 of a file to write all the log messages to. 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. Log messages will always be shown on screen, regardless of these settings.

The next items are a list of events for which logging can be enabled or disabled. The items are self descriptive except maybe for the last two. '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 default value of the other options is 'on'.

The 'Setup - Signon/Signoff' option

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 'Signon/Signoff' dialog box.

There are a few special characters that you can enter in your signon/signoff text which get expanded while being sent to a client. These are:

- %t - displays the current time on your PC
- %d - displays the current date on your PC
- %u - displays the current number of Serv-U users logged in

So, you could use the following signon text:

Welcome, it is %t on %d, and you are user number %u

I'm sure you'll figure out by yourself what this will look like to the user . . . If you have ideas for other useful special characters, let me know about it!

The 'Setup - Users' option

Unless you switched off all security, you are going to have to set up users. And, you guessed it, this is the place to do so!

Upon choosing this option a dialog box is presented to you. It contains a list of all known users. To change the settings for a certain user, just click on the name and click 'Edit' (or just double-click on the name). Now, if you just started this server for the first time there will be no names, short of Divine Intervention. Never mind, just go ahead and click 'Edit'. You'll be presented with an empty dialog box containing entries for everything you always wanted to set for a user.

The next thing is important, so pay attention: You can fill in or change any name in the 'User name' field. If this 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. So, if you double-clicked on user 'James' and then go on to set his password to 'lightbulb' and you change the user name to another of your users, '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. 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 'Edit User' 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 'Edit Group' dialog box. This goes for all settings, including password, home directory and path access rules. To overrule a certain group setting, simply provide one for the user.

For example, you're 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, 'RonaldR', and their home directory (all their files are shared anyway), '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 directories in his access path rules with 'no access', and you're all done.

We did get ahead of ourselves in the discussion of the various fields, so let me back up a bit. The 'Password' field shows stars ('*****') when entering a password. Don't worry, this is only to protect you from prying eyes. Also if you're editing an existing user who has a password, nothing will be shown here, but the password is still there. To keep the existing password for a user: don't edit this field! The passwords are stored encrypted using UNIX 'crypt'. This algorithm works like a sausage machine: you put in a pig on one side 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).

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 needs 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 always need to be full path names, including a drive letter!

This brings us to the last part of this dialog box, the 'File/Directory access' rule list. 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. No access paths, no access! So, there is one path you might always want to be in the list: The user's home directory.

When a user executes a FTP command concerning files or directories, the user's path list is

checked to see if the command should be allowed to proceed. The list is evaluated FROM TOP TO BOTTOM! SO THE ORDER OF THE PATH ACCESS RULES IS IMPORTANT!!!

There are five different types of access information that can be set for each path:

- ‘Read’ access, this allows files to be copied from the PC using the FTP ‘get’ command.
- ‘Write’ access, allowing files to be copied to the PC using ‘put’, but not changed, deleted, or renamed.
- ‘Delete’ access, allowing the user to change files, rename, or delete them. Having ‘Delete’ access automatically includes write access.

The last two items deal with directories:

- ‘Create’ access lets the user create new directories at this path.
- ‘Delete’ lets the user delete directories.

To get a directory listing any one of these rights is sufficient for a path. If none of the rights have been granted for a certain path, then the user has no access what-so-ever to this path.

The rights of a certain path are valid not only for the path itself, but also for all subdirectories of it. If you don’t want this to happen for a certain subdirectory you have to specify this directory with the desired rights before its parent in the list of paths.

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, 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 fingers on it, since you want to check for viruses first. So, this 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 rights
Y:\ANONFTP              - read rights
```

Reversing the rules will not work: If a user would 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 access. Please take note that write access still allows a user to get a directory listing of the UPLOAD directory, but he won’t be able to download anything from there.

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

```
\                          - read, write, delete, create dir and delete dir 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. Lets take the previous example about the anonymous FTP server. We want to put a file ‘SECRET’ 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	- no rights
Y:\ANONFTP\UPLOAD	- write rights
Y:\ANONFTP	- read rights

Again, the order of the paths is important! The directory access rights do not have any meaning for files. Alternatively, if SECRET was a directory instead of a file, the above settings would keep users out of this directory. In fact, they would not even be able to get a directory listing.

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: using FTP. These ports are treated like any regular path name, a user needs access rights to use them. So, to make a printer on PRN: accessible and a modem on COM2: the user needs the following rights:

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

The buttons speak for themselves, so I'll not waste any bytes on them.

There are two special user names, although in setting them up they are dealt with exactly the same as any other user. These are the user names 'Anonymous' and 'ALL'. (Actually there are three special names: User name 'FTP' is taken to be synonymous to 'Anonymous'.)

We already came across 'Anonymous', it allows users to log in without a password. The FTP server asks for their E-mail address instead and logs this. The 'Password' section in the 'Setup Users' dialog box is ignored for this user name. If an anonymous user logs in, he will normally not see the whole file structure. To him everything will appear to be relative to the home directory. So, to abuse our previous example yet another time: After logging in he will be put in Y:\ANONFTP, but if he asks for the current path the server will answer '\'. All actions concerning files will, also, be relative to his home directory. The reason for making it this way is that certain World Wide Web browsers that log into an anonymous FTP server will execute a 'change directory' to '\' immediately thereafter. They get mighty confused if this is refused, so by making their home directory appear to be '\' this is avoided.

The other special user name is '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!

The 'Setup - Groups' option

Choosing this presents you with exactly the same dialog box as the 'Setup - Users' section. The only difference is that you cannot fill in a user name. Everything works the same way too, so I'll let you figure it out. Of course, you're not dealing with users here but with group names.

2.4 The Help Menu

This menu choice is still a bit underdeveloped, it has only the 'About' item. This does however

present you with a very beautiful 'about' box, thus more than making up for the lack in other areas. This is also the place where you will see your name or that of your company appear once the program is registered.

2.5 The Logfile and Screen

Although they are strictly speaking not part of the menus, this is a convenient place to discuss the format of the logfile and screen messages.

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 (It is also possible to get the file via FTP using Serv-U itself!), even when Serv-U is running. 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 six 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)

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.

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 3.1. To check for shaky pointers and catch all those resource leaks the program Bounds Checker version 2.1 from Nu-Mega was used. I think no serious Windows programmer should be without the latter, very recommended!

This whole project started about a year ago after much disappointment with the existing FTP servers for WinSock. In its current version it consists of just a little over 11000 lines of C++ code, divided into 16 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 WinSockets and hiding all the nasty parts of dealing with them. It uses 100% asynchronous WinSock functions (also called 'non-blocking' functions) thus avoiding problems with multiple active sockets for a single task and re-entry (let me know if you're interested in this class library, I'm thinking of selling it in source code format). 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!

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, an environment variable SERV-U is checked. If this variable exists it should be set to the path where SERV-U.INI is found. Next, 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. This makes it easy to set things up for network users where there is a single copy of the program but each user needs its own settings. If after this no .INI file has been found yet the directory of the Serv-U program is searched. If SERV-U.INI is found here it will be used, if not, the program will create one, in the program directory.

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=23
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
RegistrationKey=S%FgdfsdEvG,Rob Beckers,Cat Soft
AnonRelPaths=YES
Window=100,100,400,300

[SIGNONOFF]
SignOn1="Welcome to Robby's FTP-Server!"
SignOn2="It is %t local time on %d and you are user nr. %u"
SignOff1="Thanks for logging in!"
SignOff2="Hope to see you again soon . . ."

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

[USER=Anonymous]
HomeDir=d:\anonftp
Access1=d:\anonftp\upload,W
Access2=d:\anonftp,R

[USER=Rob]
Group=system
Password=WdRx.Jlk0kemm%
HomeDir=c:\
Access1=\.RWMCD
Access2=lpt1:,WM
Access3=prn:,WM
Access4=aux:,WM
Access5=lpt2:,WM
Access6=lpt3:,WM
Access7=lpt4:,WM
Access8=com1:,RWM
Access9=com2:,RWM
```

```
[USER=ALL]
HomeDir=y:\
Access1=y:\,R
```

```
[GROUP=SYSTEM]
Access1=c:\system,RWDCM
Access2=d:\,RWDCM
Access3=y:\novell,RWD
```

All but three of these settings can be changed and set interactively through the 'Setup' menus. The exceptions are the entries for 'Invisible', 'RegistrationKey', and 'Window', and if you desire a user to really have no password the entry 'Password=' has to be set manually for that user.

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.

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, 1 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 5 GETs and 5 PUTs per session, puts the server off-line after 1 hour, and notifies clients that they are looking at an unregistered try-out version. However, it does not 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 does 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.

'AnonRelPath' determines if anonymous 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 anonymous users to the subtree of their home directory, they cannot switch to other drives. If you switch it off then anonymous users will be treated like all others. Default it is switched on.

The last 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'

[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 signon message is put in 'SignOnxx' entries (with xx the line number), and the signoff message is put in 'SignOffxx'.

There are three special character combinations recognized by Serv-U and they are expanded to their actual values when a user logs on or off. These are:

%t = current time
%d = current date
%u = current number of users that are logged in

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 ('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' ones, 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 must be 76 and the last section should be between 48 and 89 (limits included).

[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. Also, access rules apply not only to the exact path, but to all subdirectories as well. If different settings are needed for a subdirectory, than a rule with that directory should appear before its parent, i.e. with a lesser number. The path in an access rule is followed by a comma and the access information itself. This can be a combination of up to five different characters:

- R = read access to files
- W = write access to files
- M = modify access to files (implies write access)
- C = right to create subdirectories
- D = right to delete subdirectories

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. For a user to be able to list the files in a directory he needs

at least one of these five rights, any will do. Another thing to realize is that write access to a file does not 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' and 'D' 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 his group's entry. It can contain any of the regular entries.

[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.

4. Getting In Touch - Bugs & Registration

I'd love to hear from you! Not only for bugs, but also if you have ideas, questions, or remarks. Please drop me a line! The fastest and easiest way to do so is by E-mail. My address is:

RJB@eel-mail.mc.Duke.edu

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

Rob Beckers
1911 Erwin Road, Apt. I
Durham, NC 27705
U.S.A.

4.1 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 beastly, 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. Also, what operating system (DOS version and Windows or Windows-For-Workgroups 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 very 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 . . .

4.2 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) months of work, spread out over the last year. 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 20 bucks for registration mean a lot to me!

To register, please fill out the registration form below (There is a separate one in ASCII format in the file REGISTER.TXT.) and send it to me. Payment should in principle be included with the registration form, although there are exceptions possible for foreign (non-USA) customers, to which I'll get in a moment. If you are inside the US a personal check will do fine. The

registration fee is \$20 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 \$20 each

License for:

20 \$200

50 \$400

100 \$600

100+ \$500 per block of 100 users

Licenses for more than 500 users are negotiable.

Educational discount: For licenses of 50 or more users: 30% discount.

Payment from outside the USA is a problem. Despite all the international networks the easy solutions are costly and the cheap solutions risky. The exceptions are if you are inside the Netherlands, then payment is easy (see further on), or if you're in a European country and can use Euro-checks, see below for details. For non-US customers the following forms of payment are accepted, in order of preference:

- By Euro-Cheque, for Europeans only. The check must be made out in NLG (=NederLandse Guldens), and the price per copy is NLG 35. Please don't forget to sign the check both on the front and the back and to fill in the 4 digit security number on the back. Make it payable to R. Beckers - Bunde. Mail the check to: Rob Beckers; St. Agnesstraat 16; 6241CB Bunde; The Netherlands. Please mail the filled out registration form to the US address mentioned on the form.
- By check, drawn at an American bank. The check should be made out to Rob Beckers (Alas, Cat Soft only exists in the mind for now).
- 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 USD \$20 and to Rob Beckers.
- By cash, but only in US dollars and I give no guaranties about safe arrival! Please DO NOT send me other currencies, it would probably cost me much more to convert them to \$\$'s than it costs 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 improves chances of arrival considerably.

For companies buying a site license there are a number of less cheap but more secure options, again in order of preference:

- By direct money transfer in US dollars to my American bank account. This costs you \$35 extra, since that is the atrocious rate my bank charges me to receive money this way. If you are interested, ask me for details.
- By sending me a (foreign) check from your own bank. The check should be made out in USD and payable to Rob Beckers. Using this option costs you \$18 extra, which is how much my bank charges me for cashing such a check.
- By direct money transfer to my Dutch bank account. Only \$9 extra for this one, which is what they charge in the Netherlands. Ask me for details if you're interested.

Now for the **Dutch**:

Daar ik nog steeds een Nederlandse girorekening heb is het mogelijk op die wijze te betalen. De prijs bedraagt f. 35,- per kopie. Dit dient overgemaakt te worden op girorekening 53.95.461 ten name van Rob Beckers, te Bunde. Vermeld s.v.p. "Registratie Serv-U" zodat duidelijk is waarvoor betaald wordt. A.u.b. geen geld vanuit het buitenland overmaken! Van die 35 piek zou dan heel weinig overblijven. Het registratieformulier gewoon naar de VS sturen (post of E-mail).

Next, 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, even after the 30 day try-out period. Please let me know your E-mail address, so I can notify you fast. 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 financial status!

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' dialogbox: 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 next page is the registration form. Please use this form for all registrations! That way I can keep my administration manageable.

Rob Beckers

1911 Erwin Road, Apt I
Durham, NC 27705
U.S.A.

Registration Form Serv-U

Name:

Company Name:

Address:

.....

.....

Phone Number:

E-Mail Address:

(Internet only please)

Additional comments, suggestions, complaints, praise, etc ...

.....
.....
.....

Registration fee is \$20 per copy. Send this order form along with your payment to: Rob Beckers, 1911 Erwin Road Apt. I, 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 RJB@eel-mail.mc.duke.edu. Check out the site license prices 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.