

ZanNet Manual (Beta II)

by Zan Software

Table of Contents

1	Copyright/License/Warranty Information.....	2	
	License Agreement.....	2	
	Evaluation and Registration.....	2	
	Distribution.....	2	
	Registered Version.....	2	
	Governing Law.....	2	
	Disclaimer of Warranty.....		3
2	About ZanNet - What is it?.....	3	
3	Operating System and Software Requirements.....	4	
4	Installation, Setup, and Removal.....	4	
	The Windows 95 Network Client.....	4	
	The Unix Server.....	5	
	Configuring ZanNet.....	6	
	Logging In.....	6	
	Removing ZanNet.....	7	
5	Unix Porting and Compatibility.....		7
6	Known Limitations, Problems, and Other Notes.....	8	
7	Performance Considerations.....	10	
	The Status Dialog and Streaming Disabled.....	11	
	Killing I/O.....	11	
	Killing I/O with the Hot-key.....	12	
	Killing I/O with the ZanNet Status Dialog Buttons..	12	
8	Ordering Information.....	12	
	Entering Your Registration Key.....	13	
9	Product Support	13	
10	Zan Software Contact Information.....	13	

1. Copyright/License/Warranty

ZanNet, Copyright 1996-1997 by Zan Software
All Rights Reserved.

License Agreement

Please read the following terms and conditions before using this software. Your use of this software indicates your acceptance of this license agreement and warranty.

Evaluation and Registration

This is not free software. You are hereby licensed to use this software for evaluation purposes without charge for a period of 30 days. If you use this software after the 30 day evaluation period a registration fee of \$49 is required. Payments must be in US dollars drawn on a US bank, and should be sent to Zan Software, 332 SE 7th Ave, Delray Beach, FL, 33483. Credit card ordering through e-mail and fax, in addition to quantity discounts are available, as described in the included file: order.doc. All orders must include that form.

Unregistered use of ZanNet after the 30-day evaluation period is in violation of U.S. and international copyright laws.

Distribution

You are hereby licensed to make as many copies of the trial version of this software and documentation as you wish; give exact copies of the original trial version to anyone; and distribute the trial version of the software and documentation in its unmodified form via electronic means. There is no charge for any of the above.

You are specifically prohibited from charging, or requesting donations, for any such copies, however made; and from distributing the software and/or documentation with other products (commercial or otherwise) without prior written permission.

Registered Version

One registered copy of ZanNet may either be used by a single person who uses the software personally on one or more computers, or installed on a single workstation used non-simultaneously by multiple people, but not both.

Governing Law

This agreement shall be governed by the laws of the State of Florida.

Disclaimer of Warranty

THIS SOFTWARE AND THE ACCOMPANYING FILES ARE SOLD "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED.

Good data processing procedure dictates that any program be thoroughly tested with non-critical data before relying on it. The user must assume the entire risk of using the program. **ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE.**

2. About ZanNet - What is it?

ZanNet is a Windows 95 network client and Unix server that provide you with a Windows 95 network drive to access your server files. The product includes network provider and redirector for Windows 95 in addition to a Unix server. The server portion ships with both POSIX compliant source code and binary support for select Unix platforms. ZanNet is intended to replace both File Transfer Protocol (FTP) and Telnet programs currently used to access web page and other files through an Internet Service Provider (ISP).

Windows to Unix networking is not new. There are a number of solutions including the Network File System (NFS) and Samba. Unfortunately, with today's fledgling Internet service market, very few service providers are willing or able to give their customers access to these servers. In most cases, Windows users that want to manage a web page must do so using both FTP and Telnet. With ZanNet, you can get the functionality of a Windows network drive without requiring any configuration on the part of your service provider. Additionally, the ZanNet client is designed to handle slow or dead servers by allowing you to stop network I/O at any time through both hot-key sequences and control buttons. This feature is extremely important for applications that have the potential to lock-up Windows while waiting for network I/O (for example the Windows Explorer).

To use ZanNet you must have Windows 95, Microsoft's TCP/IP Protocol, WinSock, a Unix style shell account, and an Internet connection. The ZanNet client is installed in the same fashion as any other network client (see installation instructions later for details). Once the client is installed, you must install the server portion by uploading either the source or binary files depending on whether binaries for your version of Unix are included. When both client and server are ready, you need to add a new ZanNet server so the ZanNet client can login to the Unix machine and execute the ZanNet server. Once the server is started, it will open a connection back to your Windows machine through a TCP/IP port that you can define in the client configuration. The ZanNet client uses WinSock to establish a control connection through Telnet to the server while regular file system requests are handled at the Windows driver level directly through the Microsoft TCP/IP stack.

Once ZaNNet is set up, you can configure it so that Windows automatically maps a network drive to your Unix shell account whenever you establish a connection to the Internet. Then, manage your web files through whatever Windows utility or application you like. Be sure to read sections 6: *Known limitations and problems* and 7: *Performance Considerations* for notes on using ZaNNet.

3. Operating System and Software Requirements

The ZaNNet software package consists of a Windows 95 client and Unix server. Source code for the server portion is also included, allowing ports to various Unix platforms. Because that source code is POSIX compliant, other operating systems supporting the POSIX standard may be compatible with ZaNNet source code.

For the client side, you will need:

- Windows 95.
- WinSock running on Microsoft's TCP/IP Protocol.
- An established Internet connection.

In most cases, ZaNNet will NOT work with other TCP/IP stacks from vendors other than Microsoft. The ZaNNet client interfaces with both WinSock and Microsoft's TCP/IP VxD Interface. Confused yet? As long as you are running the TCP/IP protocol that came with Windows 95, you should be fine.

For the server side, you will need:

- A Unix style shell account accessible over the Internet.
- File Transfer Protocol (FTP) access to that account.
- A ZaNServer binary for your shell account's operating system and hardware platform OR C compiler access.

4. Installation, Setup and Removal

The Windows 95 Network Client

Before you begin installation of ZaNNet, we assume you have Windows 95 with a working Internet connection. Additionally, you should be running Microsoft's TCP/IP Protocol and Microsoft's WinSock (both are shipped with Windows 95). You can verify the protocol by checking the *Control Panel* | *Network* dialog where you should see TCP/IP listed as an existing component.

To install the ZaNNet client, click on the *Add...* button within that same dialog and select *Client* as the type of network component you want to install. Click on that dialog's *Add...* button to get to the *Select Network Client* dialog. Here, you need to click on the

Have Disk... button where you can click **Browse...** to get to where the ZanNet files are located. Once you are in the directory where the uncompressed ZanNet files are, you should see **zannet.inf** highlighted. Click on **OK**. Click on **OK** again in the following **Install From Disk** dialog. Two more **OKs** are required on the next two dialogs (Windows may require the installation disks after the final **OK**).

After rebooting your computer, look on the lower right hand corner of your screen. If the ZanNet client was installed correctly, you should see a blue “**Z**”. Seeing this tray icon indicates that both the ZanNet Network Provider and Network Redirector were installed correctly and have been loaded. This is also an indication that the redirector has found the proper version of TCP/IP. If you get a load error such as “**Can't load redirector**”, you may not have the required version of TCP/IP.

The Unix Server

You can install the server once the client has been successfully installed. The first thing you should do is verify your Unix operating system brand (Linux, SunOS, etc.), version, and hardware platform (Intel, Sparkstation, etc). ZanNet Beta II currently ships with the following Unix binaries:

zanserv.lnx	--	Linux for Intel compiled with version 2.0.11
zanserv.sos	--	SunOS for Sparkstation compiled with version 4.1.4
zanserv.bsd	--	FreeBSD for Intel compiled with version 2.1-STABLE
zanserv.bdi	--	BSDI for Intel compiled with version 2.1
zanserv.aix	--	AIX for IBM RISC System/6000 compiled with version 4.1.4
zanserv.irx	--	IRIX for Silicon Graphics compiled with version 5.7

By the time you are reading this, there may be more binary support. Just check the included **readme.txt** file.

If you don't know what type of Unix your provider runs, login to your shell account and type:

```
uname -a
```

Where you will get the Unix brand, machine name, operating system version, and machine type. Check the man page on that command for your Unix implementation to see how this information is ordered (**man uname**).

If ZanNet provides binary support for your Unix, simply upload the appropriate binary to your shell account using FTP or similar tool. Once the file is in your home directory, you will have to make it executable. Something like this:

```
chmod 755 zanserv.sos
```

Then, just type in the name of the binary to verify that it will load on that machine. You should get output similar to this:

unx% zanserv.sos

Usage: zanserv.sos <IP Address> <Port>

This machine's host name is: unx

This machine's IP address is: 204.91.67.111

If the binary does not load, you will get output like this:

zanserv.sos: Exec format error

That output indicates that the binary is not compatible with your Unix platform. Make sure you have uploaded the correct binary. If you still can't get it to load read the section on *Unix porting and compatibility*.

(**Tip:** Some Unix implementations—most notably Linux—don't have the current directory in the path and require you to preface the ZanNet server executable with ./, or ./zanserv.lnx)

Configuring ZanNet

Once you have the server installed correctly, you are ready to add a new ZanNet server with the add server wizard. First, double click on the blue Z tray icon. That will bring up an agreement dialog for the trial version. Read the agreement and click on **I Agree...** The ZanNet property page will be followed with the ZanNet add server wizard. Follow the instructions for each page and click on Finish at the end. Your new server, named with a combination of Unix host name and user name, will be added to the server list box.

If, at any time, you wish to change the above server setup information, simply highlight the server, and click on **Properties...**, or just double click on the server name itself. You can add and remove servers with the appropriate **Add...** And **Remove** buttons.

The ZanNet property page also contains tabs to configure global properties. Be sure the flip through these pages and read the associated on-line documentation.

Logging In

After you have configured a new ZanNet server, initiate a login by mapping a new Windows Network Drive. You can map a drive by right clicking on the **My Computer** desktop icon and selecting **Map Network Drive**. Or, simply highlight the server in ZanNet properties and click on **Map Network Drive...** Windows will bring up its Map Network Drive dialog with the next available drive letter selected. In the Path edit field, you need to enter the Universal Naming Convention (UNC) path that corresponds to your

server's Telnet host name and the user name for your shell account. The Telnet name should correspond to whatever you specified for your *Hostname or IP Address* in the server properties. For example, if your Telnet host name is:

login.whatever.com

You would specify the following in the server properties:

Hostname or IP Address: login.whatever.com

And then map the new network drive to the following UNC path:

\\login.whatever.com\myusername

Where, *myusername* is the user name for your shell account and *login.whatever.com* is the Telnet hostname. Make sure you put double back slashes before the host name and a single back slash before the user name. The ZaNNet client network provider will parse the UNC path and connect to the host using WinSock. If you did not previously enter a password, you will have an opportunity to enter a password and may elect to cache that password. Once the connection has been established, the tray icon will turn yellow. If the server successfully connects back to your Windows machine, the tray icon will turn red.

Removing ZaNNet

The best way to remove ZaNNet is to bring up the Windows | Control Panel | Network configuration, select *Zan Software*, *ZaNNet Client*, and click on remove. You can also remove ZaNNet from the opening agreement screen by clicking on the *Remove ZaNNet* button, or in the Registration property page with the same button.

5. Unix Porting and Compatibility

Can't get a ZaNNet server binary for your Unix implementation? Don't worry, it's a snap to compile the server. First thing to do is upload the files: *zanserv.c* and *zanserv.h* to your shell account. This is extremely vanilla C code written under the POSIX.1 standard. Once those files are uploaded type in the following:

```
cc -o zanserv zanserv.c
```

The command prompt should return to you with not errors. If you got something like this:

```
cc: Command not found.
```

That is bad news. Your service provider does not provide you with C compiler access. In this case you must either beg your provider to compile it for you, or get a ZaNNet server

binary from somewhere on the net. Some compilers (most notably the *gcc* compiler) allow you to specify multiple target machines. Perhaps you can find someone that will compile the source for your target operating system and architecture.

Verify that your binary has compiled successfully by typing *zanserv* at the command prompt see section 4 for the output you should get. Don't forget to modify your login script to match the name of the server binary!

(Note to Unix Gurus: In order to keep the compilation simple and clean we chose to keep all server code within a single C source file. Also, there is no makefile either. Though some may consider this poor project management, our design requires server simplicity both in source management and object code.)

6. Known Limitations, Problems and Other Notes

The following section is a list a potential pitfalls you may encounter while using ZanNet beta II. Some of these will be fixed in future releases, while others are design limitations. We compiled this list during our development cycle for Beta I and Beta II. Many of the listed file system requests are not supported for obvious reasons (for example direct disk I/O on a network drive). We have listed all unsupported file system requests for completeness.

- No attempt is made to support older applications that use the 8.3 naming standard. Applications that cannot handle long file names will see a truncated version of any name over the 8.3 limitation and will fail if they try to manipulate that truncated file name. If you MUST use an 8.3 applications, be sure to keep file names under the 8.3 limitation. The MS-DOS Prompt included with Windows 95 works fine, but some DOS commands fail, for example: DIR works, but DIR /S fails. We recommend using the Windows Explorer.
- The file system request: ***FS_SearchFile*** is not supported. This request handles all older DOS find first and find next operations - the INT 21h functions: 11h, 12h, 4Eh, and 4Fh. In some cases this may break older DOS applications.
- The file system request: ***FS_EnumerateHandle*** is not supported. This request handles the Win32 function: *GetFileInformationByHandle()* and the DOS INT 21h functions: 440Dh, subfunction 086Dh. This function is documented by Microsoft as an optional file system function. Any application that depends on this function, though in error, may fail.
- The file system request: ***FS_LockFile*** is not supported. This request handles the Win32 functions: *LockFile()* and *UnLockFile()* and the DOS INT 21h function: 5Ch. All file region lock requests currently return successfully. This may break some applications (notably: Microsoft Access).

- Overlapped I/O used by the Win32 functions: WriteFileEx() and ReadFileEx() is not supported by the ZanNet file system requests: *FS_ReadFile* and *FS_WriteFile*. Not many applications are using overlapped I/O, and none should be dependent on it.
- The file system request: *FS_MountVolume* is not supported. Network drives don't use this function.
- The file system request: *FS_Ioctl16Drive* is not supported. This request supports the DOS INT 21h function 44h. This will break older DOS applications that try to perform direct disk I/O. Network drives don't support this request.
- The file system request: *FS_DirectDiskIO* is not supported. This request supports the DOS INT 21h functions: 25h and 26h. This will break older DOS applications that try to perform direct disk I/O. Network drives don't support this request.
- The file system request: *FS_GetDiskParms* is not supported. The ZanNet file system driver does not maintain DPBs. This should not effect applications.
- The Windows attributes: *Archive*, *Hidden*, and *System* are not supported. While these attributes can be set without error, they will never be reported. This could possibly break some applications.
- The FindFirst must match attributes are not yet supported. This may break some applications. Also, case sensitive file searches are not supported.
- The directory entries "." and ".." are never reported during a directory listing. This may break some applications.
- All files are opened fully read/write sharable (OF_SHARE_DENY_NONE). Sharing flags will be ignored, but will return successfully. **The user must manually guard against problems with file sharing. I.e. don't open the same file twice over ZanNet.**
- If you elect to disable read and write streaming through the property page **Performance**, some applications will run poorly. Most notably are file compression software packages which tend to make extremely small file system reads. Read section 7 *Performance Considerations* for more on this.
- File size is limited to 4 gigabytes.
- The disk usage report is approximate and may not exactly match statistics reported from the Unix command *du*. Disk space remaining is calculated by recursively traversing down the file tree starting from your home directory. If your tree is very large you should disable this feature by setting *Max. Server readdirs for Diskspace* to 0.

- You cannot browse a ZaNNet network through the Windows Network Neighborhood. ZaNNet does not really fit into this method of browsing. This feature may be added in the future.
- Command line *NET* commands such as *NET USE* are not supported and may cause unpredictable results.
- You cannot map multiple drive letters to a single shell account. You can, however, map multiple drive letters to multiple, different shell accounts.
- Some service providers do not allow multiple logins to a single shell account through Telnet. With a provider such as this, a mapped ZaNNet drive will make it impossible for you to login with a regular Telnet program. Simply disconnect the ZaNNet drive to resolve this contention. You may also break into the server program through the ZaNNet server debug window with a control-C and execute line mode commands. To restart the server, just log-out of the Telnet session and the ZaNNet client will re-establish the connection.
- If you use a Unix host name, the ZaNNet server must be running on the same IP address that you Telnet in to. In some rare cases, a provider may route your Telnet login to another machine that does not match what the name server returned to the ZaNNet client. You can avoid this problem by specifying a *Unix Server Final Address* in the Unix property page.
- Right mouse button click on .exe files results in many file system calls and will respond very slowly on slow connections.
- Basing your server to the Unix system root is very slow because all server paths must be mapped to Windows paths. We suggest that you use the default for best performance.

7. Performance Considerations

When you are using the ZaNNet Beta, there are a number of things to kind in mind. First of all, if you have never have experienced a network drive over a slow connection such as a with a modem, you may find it very frustrating to use this product. There is just no comparison between hard drive speeds and modem speeds. Additionally, many applications are developed on, and designed for hard drives, and or, high speed local area networks. Some such applications use a ridiculous number of file system requests. For example, we have seen a few that make sure the file is present (*FindFirstFile()*) for every mouse click! Even though there is very little data transferred in this scenario, just the ping time can cause a one second delay for that mouse click. We leave it up to you to weed out any such applications. For example, Windows Notepad works great for a quick update of HTML, while some HTML specific editors get a little carried away with backup files, etc.

The Status Dialog and Streaming Disabled

If you are operating without ZaNNet streaming enabled keep in mind buffer sizes. Try right mouse clicking on the ZaNNet tray icon. Select **Status**. That will bring up the ZaNNet Status dialog. Our network redirector posts status for file open, close, read, and write to this dialog for all volumes. Here is a rundown on the fields of the status dialog:

Server: The server name in UNC format (ex. \\login.whatever.com\myusername).
File: The name of the file (ex. index.html).
Operation: What kind of operation just completed (ex. Write).
Bytes Moved: The number of bytes moved since the last report (ex. 16384)
File Position: The current file position after that operation. (ex. 97792)
Speed: How much time in kilobytes per second that operation took (ex. 2.6k/sec).
Last Update: When the status was last updated (ex. 16:1:21)

For example, suppose you are extracting a file from a .ZIP archive and you notice it is taking an incredibly long time. After pulling up the status dialog, you see that the application is performing many tiny reads of less than 20 bytes. The application is *depending* on file caching for good performance. Those little buffers are adding a HUGE overhead because a read request must be sent to the server to fill each buffer.

Another Windows caveat to keep in mind has to do with the Windows Explorer. For whatever reason, the Explorer seems to be single-threaded, and tunes its buffer sizes to match the speed of your connection! That way, a very poor connection may have 512 byte file copy buffers, while a better connection may use 4096 byte buffers. Why do you care? When you operate ZaNNet without streaming enabled, each read (or write) must be requested by the client adding somewhere around the ping time for each read. For example, if it takes $\frac{1}{4}$ of a second to reach a given server, that time is added as overhead for each read buffer. Take a 100,000 byte file with Explorer using 4096 byte buffers. 100,000 divided by 4096 is around 24. Multiply that by $\frac{1}{4}$ seconds gives you an overhead of about 6 seconds. Not bad, but if you do the math for a 10MB file using 512 bytes buffers, you'll see why we think this is a concern.

Also, know that the Explorer does a connection speed calculation before it copies a file. Could be a problem if your connection is flaky during this time.

Our recommendation is to enable both read and write streaming whenever possible.

Killing I/O

We all are painfully aware by now that the Internet is far from perfect. Most popular WinSock applications have some kind of stop button or other mechanism to abort a file transfer. The ZaNNet design required the same functionality, but, because the ZaNNet client is a network redirector running as a VxD, a simple button would not suffice.

Though you may not be aware of it, there are many cases where Windows locks the user front-end waiting for I/O (Explorer does this). Makes it kind of hard to click on a button!

Killing I/O With the Hot-Key

We solved the I/O killing problem with a variety of techniques. The first, and probably most powerful, is a hot-key sequence. This hot-key goes directly to the ZaNNet redirector VxD. The actual hot-key can be configured in the ZaNNet Properties | I/O Control. On that page, you'll notice something called: *Time, in seconds, that ZaNNet considers an I/O request "dead"*. When you hit the hot-key, the redirector goes through its entire list of I/O requests for all ZaNNet volumes. Each request that has been waiting for a response for more than the dead I/O seconds value is flagged as an "Unexpected Network Error" and the application thread is released with that error code set. You can set this value to 0 to make the redirector abort everything.

When the thread that requested ZaNNet I/O returns, the application sees the error and hopefully aborts I/O. Many times the application will request a file close after the returned error in which case you'll have to hit the hot-key again. You may have to kill the application if it does not behave properly. Of course, it's always best to disconnect the network drive for problem servers if you can get to Windows.

Killing I/O with the ZaNNet Status Dialog Buttons

The ZaNNet status dialog contains two stop I/O buttons. The first called *Stop This File's I/O* will stop I/O (in the same way as the hot-key) for the file listed in the status dialog only. This method is much less destructive than the hot-key because it will only abort a given file, not every pending request. Please note that you should always try to use whatever the application provides for stopping a request first. For example, the explorer has a *Cancel* button for file copies.

The other button in the status dialog is *Stop All ZaNNet I/O Now!* This one produces the same effect as a hot-key sequence with a 0 second dead I/O value (see above). There is also a menu item on the tray icon called *Abort All Dead I/O* which has the exact effect as the hot-key.

8. Ordering Information

Current ordering methods are e-mail, regular mail, and fax. Payment options are check, credit card, and purchase order (\$250 minimum for purchase order). All orders must include a properly filled out order form. This form is contained in the enclosed file: *order.doc*. You may fill out the form on your computer with an editor and send it electronically, or print it out and hand write it for regular mail or fax. Multi-user pricing is included in the order form.

Entering Your Registration Key

Your order will be processed within a 24 hour period after which time you will receive a registration key. That key should be entered in the **Registration** page of the ZanNet properties. Within that page, enter your first name, last name, and the key supplied by Zan Software. Make sure you use the same first and last name as supplied with your key.

9. Product Support

Initial contact for product support must be made through e-mail, fax, or regular mail. In your problem description, please state your name, e-mail address, and phone number. If your problem cannot be easily resolved via e-mail, you will be given a support number to call.

10. Zan Software Contact Information

Zan Software
451 4th Ave. S. #106
Kirkland, WA 98033

E-mail: see below web page--tech support section

Web: www.zannet.com

Fax: (517)731-2642

Phone: Please e-mail us for a voice number.