Finger Daemon Help Index

File Menu

The File menu includes commands that enable you to customize the functioning of the finger daemon.

For more information, select the File menu command name.

Setup
Exit

File Setup Command

This command allows the user to customize the functioning of the finger daemon using the setup dialog box.

For more information, select the dialog control name.

Default File
Finger Dir

Normal Logging
Debug Logging

Default Only

File Exit Command

If you attempt to terminate the finger daemon while there are active client connections, then you will be warned of this by a dialog box. If you select OK, then the client connections will be broken, and you will be able to terminate the finger daemon.

In the normal situation, a dialog box verifying the exit command will appear allowing the user to abort the exit command.

Setup "Default" File

This topic explains how to select the defaul file. Pressing the Default file button will cause a browse dialog to appear. Use this dialog to select the file that will be sent in response to a "default" finger request.

"Default" Request

A "default" request is a finger request without a user name. An example would be:

```
finger host.domain
```

Setup Finger User Directory

This dialog box edit field is used to tell the finger daemon where to look for the user information that is to be sent in response to a request for a finger on a specific userid. If the directory entered does not exist, then a warning dialog box will be generated. The directory must *not* end with a backslash.

For example, if the home user directory was `\USERS`, and a finger client requests a finger of `root@host.domain`, then the finger daemon would look for the file

```
\USERS\ROOT\USERINFO.INI
```

If the user has specified a plan file, then it must be present below the \USERS\ROOT directory. The finger daemon will verify that the plan file name is a valid, and does not contain ".." or any other similar characters.

See also

Only Allow "Default"
User Information

User Information

Inorder to provide the specific information on a user, the user should create a USERINFO.INI file in their home directory. The location of the user home directories should already be set accordingly.

The contents of the USERINFO.INI file should be similar to the following:

```
[FingerInfo]
RealName=Foo Bar
Office=Room 227, x555
Directory=c:\users\foobar
Shell=progman.exe
Plan=plan.txt
```

This file will use the information provided in this file to generate the response to any finger request for the user foo. The directory entry is ***NOT*** used to specify where the plan file is. It is present to allow the system administrotor to speicfy the true home directory for this user, since the finger directory field does not allow multiple search entries.

The USERINFO.INI file above will generate the following:

```
Login name: foobar                      In real life: Foo Bar
Office:     Room 227, x555
Directory:  c:\users\foobar              Shell: progman.exe
On since: Mon Aug 01 19:37:53 1994 EDT  Terminal: console
Plan:
This is a plan file for fwbent.
```

Assuming the file PLAN.TXT is in the home diretory of the user foobar.

The On since field is generated by returning the time that the current Windows session on the server was started. If the user is

See Also:

Finger Service

Normal Activity Logging

This topic explains what normal logging is.

See also:

Debug Logging

Debug Activity Logging

This topic explains what debug logging is.

See also:

Normal Logging

Only Allow "Default"

Since MS-Window 3.x is not a multi-user environment, you can configure the finger daemon to simply return the same response to *any* finger request, including a "default" request.

Installing Finger Daemon

In order to use the Finger daemon, you must have already installed a Windows Sockets WINSOCK.DLL which conforms to version 1.1 of the Windows Sockets specification. A Windows Sockets stack is *NOT* provided, in this package and if the user does not have one, then this program will not work. The finger daemon will listen on a socket and respond to finger client requests when it has been properly installed.

To install the finger daemon follow these steps:

1. Place the executable and support files in their own seperate directory, \FINGERD for example. Copy the CTL3D.DLL to your \WINDOWS\SYSTEM directory.

2. Start the File Manager and select the FINGERD.EXE executable, and drag it to the program group that you wish to place it in.

3. If you wish to have the finger daemon automatically execute when you start Windows, then also place a second copy in the Windows Startup program group.

4. Now execute the program by double clicking on it's icon. Use the Setup command in the File Menu to configure the daemon. You should specify where the default file and user's home directories are going to be located. This information will be written to FINGERD.INI which will be located in your Windows directory.

5. After you have configured the executable, modify the program properties so that the daemon will run minimized. Use the Properties command in the program managers File Menu.

6. Now double click on the finger daemon icon to execute it to verify that it is working properly.

7. If you are going to have multiple users, place a copy of the provided USERINFO.INI file in each of the users home directory. Edit these files are needed.

Your computer should now be providing finger services.

You should also enter a TZ environment variable, by adding a line to your AUTOEXEC.BAT file, otherwise Fingerd will assume you are in the Eastern timezone (i.e. TZ=EST5EDT).

Finger Socket

The socket to be used by the finger service must connect to a well-known port address. The port addresses are normally specified in the `services` file. However, if the finger service is not specified in the `services` file, then the well-known port value of 79 decimal, as defined in RFC 1060, will be used.

Daemon

A daemon is a process that executes "in the background" (i.e. without an associated terminal or login shell) either waiting for some event to occur, or waiting to perform some specified task on a periodic basis. This term comes from Unix.

Daemon processes typically have the following characteristics:

1) They are started once, usually when the system is initialized.
2) Their "lifetime" is the entire time that the system is operating.
3) They spend most of their time waiting for some event to occur.

Glossary

daemon
default request
Winsock
Client
RFC

Windows Sockets

The Windows Sockets specification defines a network programming interface for Microsoft Windows which is based on the "socket" paradigm popularized in the Berkeley Software Distribution (BSD) from the University of California at Berkeley. It encompasses both familiar Berkeley socket style routines and a set of Windows-specific extensions designed to allow the programmer to take advantage of the message-driven nature of Windows.

Operating Finger Daemon

The finger daemon allows your computer to provide finger services to clients. The finger daemon listens on a socket for client connections.

When a connection is established, the hostname of the client and time are placed into the log listbox. The userid that the client requested to finger is also logged. In addition, the information sent to the client will be displayed in the main window. The main window will only hold a maximum of 200 lines of text before it drops the oldest lines.

Since there is no real user interface, the finger daemon should typically be run in a minimized state unless the user is setting up the daemon.

Finger

The finger service is used to provide textual information about a user on a computer. The format and content of the information returned by the finger daemon in response to a client's request is host dependent.

Finger Service

For example, a typical Unix finger daemon might return the following in response to a finger service request:

```
Login name: user                        In real life: Joe User
Directory:  /u/user                     Shell: /bin/csh
On since:   Tue Jul 12 21:29:43 1994    Terminal: console
New Mail:   Tue Jul 12 21:30:54 1994    Last read:
Plan:
My plan file.
```

However, anything that you desire could be returned by thisinger daemon. You simply have to create a USERINFO.INI file to generate the above user information.

RFC

All the Internet protocols are defined by *Request for Comments (RFCs)*, which are their formal specifications. The RFC whish specifes the Finger protocol is RFC 742.

Client-Server Model

The client/server model is the most commonly used paradigm in constructing distributed applications is . In this scheme client applications request services from a server application.

The client and server require a well-known set of conventions before service may be rendered (and accepted). This set of conventions comprises a protocol which must be implemented at both ends of a connection. Depending on the situation, the protocol may be symmetric or asymmetric. In a symmetric protocol, either side may play the master or slave role. In an asymmetric protocol, one side is immutably recognized as the master, with the other as the slave. An example of a symmetric protocol is the TELNET protocol used in the Internet for remote terminal emulation. An example of an asymmetric protocol is the Internet file transfer protocol, FTP. No matter whether the specific protocol used in obtaining a service is symmetric or asymmetric, when accessing a service there is always a "client process" and a "server process".

A server process normally listens at a well-known address for service requests. That is, the server process remains dormant until a client's connection to the server's address is made. This connection causes the server process to "wakes up" and services the client, performing whatever appropriate actions the client requests of it.

Future Enhancements

1) Allow multiple directories in the finger directory path. They would be seperated using a semi-colon, and searched in a manner similar to the environment PATH.variable.

2) Allow the logged information to be transfered to the clipboard, or saved to logging file as well. The information would be transfered when the listbox contents are cleared. Provide an entry in the FINGERD.INI file to specify a logging file, where the daemon should write the logging information to.

3) Provide an entry in the FINGERD.INI file to specify how large the logging listbox should be.

4) Create a dialog for selecting a directory.

5) Have the logging level do something more useful.

6) Make sure that Windows 3.1 is running before activating CTL3D.DLL.

Edit Menu

The edit menu allows you to manipulate the main window.

Copy Copies the entire contents of the window to the Windows clipboard.
Clear Clears the contents of the window.

The main window will only hold 200 lines of text before discrading the oldest line in the window.

Help Menu

The help menu allows the user to access information about the Finger daemon.

Index Allows you to access the index of the FINGERD.HLP file using WinHelp.
About Displays the version and copyright information.

Copyright Notice

Finger daemon version 1.0.0, a Windows Sockets Finger Protocol Daemon

Sofrware License Agreement:

Revision History

This project was started because of the large number of people on WinSock Usenet newsgroups asking for a "good" finger daemon. Currently (Aug 3, 1994), TXTSRV is the only finder daemon available that I know of but it has several shortcomings. Most obvious one is that TXTSRV will send *any* file on the server.

I started out using TXTSRV and FINGER (Lee Murach) and WSMTPD (Ian Blenke) as WinSock programming guides. However, there is practically no unaltered code from either of these programs in this finger daemon. All source code for the Finger daemon is included, all I ask is that if you use this program, please drop me a email message (`vorlon@hookup.net`) with some comments on the program. If you wish, and really feel the need you can even send me $, since dial-up Internet SLIP access is not cheap.

However, I do not want the headache of supporting this program, which is why I created this help file. But what I hope for is that MS-Windows programmers who wish to create *public domain* WinSock software can use this code as a guide for creating other public domain WinSock programs. You can create commercial/shareware programs, but you MUST abide by the distribution agreement in this software's copyright.

- Fred Bent
vorlon@hookup.net

Mailing address:

Frederick W. Bent
R.R. # 7 Stn Main
Guelph, Ontario N1H 6J4
CANADA.

Bibliography/References

Harrenstien, K., "Finger/Name Protocol", RFC 742, 7 pages, Dec. 1977.

Stevens, W. Richard, "UNIX network programming", Prentice-Hall Software Series, Englewood Cliffs, N.J., 1990. ISBN 0-13-949876-1.

Trademarks

Icon Warning

My only concern is that some people might try to stick their finger in an electrical socket after seeing the program icon. Please, do ***NOT*** place anything in an electrical socket other than a undamaged electrical cord! I will not be held legally responsible if someone electrocutes themselves.

If you have small children please purchase some of those electrical socket covers!

Lee Murach

This is the agreement that was present in the source code of TXTSRV:

Ian Blenke

This is the agreement that was present in the source code of WSMPTD:

Author: Ian Blenke

Ian Blenke cannot be held responsible for damages, expressed or implied, for the use of this software. No commercial use can be made of this product without the consent of the author. No profit of any kind can be made on the sale or distribution of this program. If you wish to distribute this program with other samples of WinSock programming, you must first contact the author so that he can keep accurate records of its usage. If you write any programs based on this source code, you may not sell them for any profit without the written consent of the author. If you incorporate this source code into a public domain program, all the author requires is a notification that "part of the code was written by Ian Blenke" and some form of notification that his name was used in the public domain software distribution. This does not represent a contract on the part of the author. If any issues cannot clearly be resolved by reading this text, immediately contact the author.

I don't like such agreements, but in today's world of lawyers and lawbreakers I have little other choice. Enjoy!