European
Microsoft
Windows NT
Academic
Centre

Gopher Server Manual
Version 0.9

Copyright Notice

European
Microsoft
Windows NT
Academic
Centre

Gopher Server Manual

Chris Adie

European Microsoft
Windows NT Academic Centre
Computing Services
30-38 George Square
Edinburgh EH8 9LJ

Table of Contents

1. Introduction

This manual describes the Gopher Server for computers running the Windows NT operating system. You should read it if you plan to install, operate, manage or deinstall the Gopher Server software. This manual assumes you have a reasonable degree of competence in the use of Windows NT, and a reasonable knowledge of Gopher.

The Gopher Server for Windows NT implements the "Classic" Gopher protocol, as described in RFC 1436. It runs as a Windows NT "service", just like the FTP Server which comes with Windows NT. By analogy with the UNIX gopher server daemon which is called `gopherd`, the Windows NT Gopher server service is called `gophers`, which is pronounced Gopher-Ess.

This version of GOPHERS originates from the European Microsoft Windows NT Academic Centre (EMWAC). Located at Edinburgh University Computing Service, EMWAC has been set up to support and act as a focus for Windows NT within academia. It is sponsored by Datalink Computers, Digital, Microsoft, Research Machines, Sequent and the University of Edinburgh.

EMWAC will collect and periodically review bug reports and suggestions for improvement.

To report a bug in GOPHERS, send a mail message to `emwac@ed.ac.uk` with GOPHERS BUG in the Subject: line.

To suggest an enhancement, send a mail message to `emwac@ed.ac.uk` with GOPHERS SUGGESTION in the Subject: line.

Edinburgh University Computing Service will be producing a "Professional" version of GOPHERS, which will be marketed through third parties. The Professional version will be fully supported and will have enhanced functionality.

2. Installation

2.1. Requirements

To use the Windows NT Gopher Server, you need to have a computer with the following characteristics:

Intel, MIPS or Digital Alpha processor.
Windows NT 3.1 final release, with TCP/IP software installed.
At least 16Mb of memory.
Network connection - typically Ethernet.

2.2. Installing

1. Log into your Windows NT system as a user with administrative privileges.
2. The Gopher Server is distributed in three versions, for the Intel, MIPS and DEC Alpha architectures. Select the appropriate ZIP file for your processor.
3. Unzip the file. You should have the following files:

`GOPHERS.EXE`	The Gopher Server itself.
`GOPHERS.CPL`	The Control Panel applet.
`GOPHERS.HLP`	The Control Panel applet help file.
`GOPHERS.DOC`	This manual in Word for Windows format.
`GOPHERS.WRI`	This manual in Windows Write format.
`GOPHERS.PS`	This manual, in postscript ready for printing.
`COPYRITE.TXT`	The copyright statement for this product.
`READ.ME`	Summary of new features, etc.

4. Decide which directory you are going to put `GOPHERS.EXE` in, and move it there. A good choice is the `\WINNT\SYSTEM32` directory, which is where many other services live. Using the Security/Permissions menu option in the File Manager, ensure that the SYSTEM user has read permission for the file.
5. Move `GOPHERS.CPL` and `GOPHERS.HLP` to the `\WINNT\SYSTEM32` directory. Start the Control Panel from the Program Manager to verify that the Gopher Server applet is represented as an icon in the Control Panel.
6. Determine which version of `gophers` you have. To do this, at the Windows NT Command Prompt, type:
   `gophers -version`

and the version number will be displayed. This manual covers `gophers`. (If the program reports a later version number, you will find a corresponding later manual in the files you unpacked from the ZIP archive.) You should also check the IP address of your machine using the command:
   `gophers -ipaddress`

This will display the name of your machine (*eg* `emwac.ed.ac.uk`) and its IP address(es) as reported by the Windows Sockets API. If this information is incorrect, you need to reconfigure the TCP/IP software on your machine. The Gopher Server will not work if this address (or list of addresses if your machine has more than one network interface) is wrong.

7. If you have installed a previous version of the Gopher Server, you must remove it by typing:
   `gophers -remove`

See section 2.4 for further information. You can use either the old or the new version of `GOPHERS.EXE` to perform this remove operation. Note that this will delete your existing Gopher Server configuration information from the Registry. IF YOU ARE REPLACING VERSION 0.7 OR EARLIER WITH

VERSION 0.8 OR LATER, READ THE NOTE AT THE END OF THIS SECTION!

8. Install `gophers` into the table of Windows NT Services (and simultaneously register it with the Event Logger) by running the program from the Windows NT command line, specifying the `-install` flag. (NOTE - it is vital that you execute this command using the copy of `GOPHERS.EXE` which you placed in the `\WINNT\SYSTEM32` directory, and not using some other copy which you plan subsequently to delete.) For instance:

```
gophers -install
```

The program will register itself and its location with the Service Manager and with the Event Logger, and will report success or failure. In the case of failure, see the section on Installation Problems below.

9. To verify that the installation has succeeded, start the Windows NT Control Panel and double-click on the Gopher Server for Windows NT Services icon. The resulting dialog should list Gopher Server as one of the installed services. If so, see the Configuration section for further instructions.

10. If you plan to use the WAIS index searching capabilities of the Gopher Server, you should obtain and install the WAIS toolkit for Windows NT, available from the same place you obtained this software. Ensure that you place the WAISLOOK program in a directory where the Gopher Server can find it - `\WINNT\SYSTEM32` is a good choice.

NOTE - UPGRADING FROM VERSION 0.7 OR EARLIER TO VERSION 0.8 OR LATER. With version 0.8, the "short name" by which the Gopher Service is known to the operating system changed from "Gopher Server" to "GOPHERS". This means that the Windows NT registry stores information about the service in a different place from earlier versions. If you are upgrading from an earlier version of the Gopher Server, this has two consequences:

The information stored in the registry by the earlier version of GOPHERS must be deleted. This can be done by running the earlier version ofGOPHERS from the command line with the `-remove` flag. Alternatively, version 0.8 or later (when you run it with the `-remove` option) will detect whether information relating to version 0.7 or earlier is present in the registry, and if so it will delete it. However, both methods have occasionally been observed to cause the EventLog Service to terminate with an access violation. This is harmless - just restart the EventLog Service from the services dialog in the control panel.

When you replace version 0.7 or earlier with 0.8 or later, any events recorded in the Event Log by the earlier version will be unintelligible. This is because the Event Viewer program cannot find the information in the registry which tells it where the `GOPHERS.EXE` file is located.

2.3. Installation Problems

The system says that GOPHERS.EXE is not a Windows NT program

This is probably because you are trying to run an executable for the wrong sort of processor. Check you have unpacked the correct ZIP file for your processor type.

The system says that the Gopher Service won't install because of a "duplicate service name"

You must remove a previous version of the Gopher Service using the `gophers -remove` command before installing with

`gophers -install`.

GOPHERS waits for a while, then terminates with a "usage" message

You must not run `gophers` from the command line or from the File Manager, except with the `-install` or `-remove` or `-version` or `-ipaddress` options. The `gophers` program is a Windows NT "Service", and must be started through the Services dialog in the Control Panel.

"CreateService Failed with error: service already installed".

This error message occurs if you type `gophers -install` when the Gopher Server is already installed. You cannot install two Gopher Servers.

2.4. Deinstalling

This section describes what to do if you want to remove the Gopher Server from your computer, or if you want to move the program to a new location.

1. If necessary, stop the Gopher Server using the Stop button in the Services dialog in the Control Panel.
2. At the Windows NT command line, run `gophers` with the `-remove` option:
   `gophers -remove`

This will remove the Gopher Server from the Service Manager's list of services. NOTE - this also deletes the Gopher Server's configuration information from the Registry.

3. If you are deinstalling the Gopher Server, simply delete the `GOPHERS.EXE` program, the `GOPHERS.CPL` Control Panel applet, and the `GOPHERS.HLP` help file.
4. If you want to move `GOPHERS.EXE` to a new location, you must move the file, then type `gophers -install`. This informs the Service Manager and Event Logger of the new location of the program. You will need to configure the Gopher Server and start it running again from the Control Panel.

3. Configuration

The Gopher Server is configured using the Gopher Server applet in the Control Panel. The Gopher Server applet looks like this:

Note that the version number of the applet is displayed in the lower left-hand corner of the dialog. The version number reported by the command
`gophers -version`
must be the same as the version number of the applet. (If there is no version number in the lower left-hand corner, you are using version 0.2 of the applet.)
You can use this dialog to:

Set the root of the directory tree containing the files you wish to make available using Gopher. Use the **Data directory:** field for this. Full details of how Gopher treats the files and directories in this directory tree are given in Section 5 of this manual. Default: `D:\GOPHER`. There are special considerations if the directory tree is located across a network on a fileserver - see later in this chapter for details.

Specify the TCP/IP port on which the Gopher Server listens for incoming connections. Use the **TCP/IP port:** field for this. The value must be a positive integer representing a legal and otherwise unused port. Default: 70.

Specify the Gopher type which corresponds to a given filename extension. This is covered in more detail below.

Enable or disable whether caching is performed. Default: caching is disabled. See Section 5 of this manual for further information regarding caching.

Specify the timeout interval (in minutes) after which Gopher's cache files will be deemed to be out of date. Specify 0 to prevent cache files from being timed out. Default: 5 minutes.

Enable and disable the logging of Gopher transactions. If this box is checked, the Gopher Server will record each Gopher request it receives in a log file. See Section 4.3 of this manual for more information about logging. Logging is disabled by default.

Specify the directory in which log files are stored. The default is the Windows system directory (`\WINNT`).

Enable and disable a special "UNIX Compatibility Mode". If this is enabled, `gophers` pays attention to files in the Data Directory with names starting with a dot. It treats such files (and also the `.cap` directory, if any) in the same way as the UNIX `gopherd` program does. If you set the Data Directory to point to a FAT volume, you should not enable this option, since directory and file names cannot begin with a dot on FAT volumes. See Section 5 for further information. Default: disabled.

Restore the default values of all the configuration settings. Click on the **Defaults** button to do this.

When you have finished making changes to the configuration, click on the **OK** button. The configuration will take effect the next time you start the Gopher Server. If the Gopher Server is already running, a dialog box to remind you to stop and restart it (using the Services dialog in the Control Panel) will be displayed.

3.1 File Extension to Gopher Type Mapping

The Gopher protocol represents the type of each file in a single byte. Several different types are defined in the protocol. The Gopher Server infers the type of a file from the filename extension, using a mapping table. The mapping table is configurable, using the list in the Control Panel applet and the buttons to its right, labelled **New Mapping**, **Change Mapping** and **Delete Mapping**.

The default contents of the mapping table are as follows:

File extension
Gopher type
Meaning
TXT
0
Text file
ZIP
5
Binary archive
ARC
5
Binary archive
UUE
6
UUencoded
SRC
7
WAIS index
EXE
9
Binary
DLL
9
Binary
GIF
g
GIF image
BMP
I
Windows bitmap
AU
s
Sound
HTM
h
HTML
HTML
h
HTML

To change a mapping in the table, select the mapping in question from the list and click the **Change Mapping** button. The following dialog will be displayed:

You may alter the extension you wish to map using the Filename Extension field, and select the Gopher type to which you wish to map it from the li

st (or you can enter the Gopher type character yourself if it's not in the list). Then select OK to confirm your changes.
To create a new mapping, click on the **New Mapping** button. A dialog box, similar to the Change Mapping box shown above, will appear. You can use this box to specify the filename extension, and to select the corresponding Gopher type. Note that you cannot create a new mapping for a filename extension

already present in the mapping list - an extension may only occur once in the list.
To delete an existing mapping, select it from the list in the main Gopher Server dialog, and click on the **Delete Mapping** button.
If there is no entry in the mapping table for a particular extension, the Gopher Server will use the "default extension mapping". This mapping is shown in the list with (Default) in the File Extension column. You may change the default extension mapping in the same way as other mappings, but you may not delete it.

3.2 Putting the Gopher Directory on a Fileserver

If the directory tree which you wish to make available to Gopher clients is located on a fileserver instead of on the local Windows NT machine, you will need to take special action.
Normally, directories on the fileserver are mapped to a drive letter on the local system. You might expect that simply using the mapped drive letter in the Gopher Server configuration dialog would have the desired effect, and indeed it does - until you log off the local machine.
Drive mappings are established only when someone logs on to the Windows NT machine. They are specific to a user, not to the machine. The Gopher Server is normally kept running, independently of whether someone is logged onto the machine or not. Often, the Gopher Server will be set to start up automatically when the operating system loads, when no-one is logged in and there are therefore no drive mappings in effect.
To overcome this, you can specify the Gopher Data Directory to the Gopher Configuration dialog using a UNC form of directory name - for instance:
`\\CLYDE\INFOSERVER`
Here, CLYDE is the name of the server, and INFOSERVER is the sharename of the directory which is to be served using Gopher.

4. Operation

4.1. Using the Services Dialog

You use the Services dialog in the Windows NT Control Panel for managing `gophers` operation.

After you install `gophers`, you can start it running by selecting it from the list of services in the dialog and clicking the Start button. If all goes well, a message box containing a rotating timer will appear while the service starts up, and will then disappear. The Gopher Server will then appear in the list of services with status "Started", and will be able to respond to Gopher clients.

If the service fails to start, it can be for one of several reasons. See the chapter on troubleshooting later in this manual for further details.

You may want to arrange for the Gopher Server to start automatically when the system is started. You can do this using the **Startup** button in the Services dialog. You can also use this button to specify a different user ID for `gophers` to run under. See your Windows NT documentation for details.

Pausing the Gopher Server (using the **Pause** button in the Services dialog) causes the following behaviour:

Any Gopher transactions currently underway will be unaffected. They will run to completion.

Any new Gopher connections will be queued.. When the service is Resumed, they will be accepted and processed.

If more than five incoming connections are received while the service is paused, the extra connections will be rejected.

4.2. Error Logging

If an error in the operation of the server occurs, the error will be logged in the Application Event Log. This log may be viewed with the Event Viewer, which you will find in the Administrative Tools program group in the Program Manager. See your Windows NT documentation for details of how to use the Event Viewer.

The errors logged in the Application Event Log are usually associated with a `gophers` problem (eg a file I/O error, or a system call failure caused by lack of resources, or a problem with the configuration information).

Problems associated with the client (*eg* the client sends an invalid Gopher selector, or a selector which points to a file which does not exist) are recorded in the Application Event Log as Warning events.

When the Gopher Server is started or stopped, Information events are recorded in the Application Event Log.

Further information on events which may be logged by the Gopher server is given in the chapter on troubleshooting later in this manual.

Note that the Event Viewer uses the `GOPHERS.EXE` program to interpret messages associated with events. Therefore, if you delete the `GOPHERS.EXE` file, the Gopher Server events in your Application Event Log will be unintelligible.

4.3 Gopher Transaction Logging

If you check the "Log Transactions" box in the Gopher Server configuration dialog, then for every request which the server receives, it will record a line of information in a log file. The log file is stored in the log file directory, which can also be configured in the dialog.

A new log file is created every day. The file name is of the form `GSyymmdd.LOG` - so that for instance the file corresponding to 4 July 1994 would be `GS940704.LOG`. For performance reasons, the current log file is kept open until the first transaction of the following day. When this transaction

occurs, the preceeding day's log file is closed, a new log file is opened, and the transaction is logged to it.
The information recorded is: the time and date of the request, the IP address of the server, of the client, and the Gopher selector string sent by the client.

5. The Gopher Directory Tree

5.1. General

The Gopher Directory (also known as the Data Directory) is the root of the directory tree which `gophers` makes available to Gopher clients. Directories within the tree are treated as Gopher menus, and files are treated as Gopher documents. Points in the file system above the Gopher Directory, or on other disks, are not accessible to Gopher clients. The Gopher Directory may be located on a disk which uses the FAT, HPFS or NTFS file systems.
The Gopher Directory must be accessible by the user ID under which `gophers` runs. By default, this is the SYSTEM user ID.
Files with the "hidden" or "system" attributes are ignored by `gophers`.

5.2. Link Files

The Gopher Server treats files ending with `.GFR` (or, in UNIX Compatibility Mode, files starting with a dot) in a special way. They are assumed to contain link information, which points to another Gopher server.
The format of such link files is very similar to the equivalent for the UNIX `gopherd` program. The file contains a number of `key=value` pairs, one to each line. Here is a typical example of such a link file:

```
Name=The Edinburgh University Gopher Server
Host=gopher.ed.ac.uk
Port=70
Numb=5
Path=
Type=1
```

The `Name` gives a user-friendly string which is displayed instead of the file name in the Gopher menu. The `Host` is the name of the computer on which the linked-to server resides, and the `Port` is the TCP port number for the server. (Specifying a + sign for the `Host` or `Port` means "use this Gopher server's host name or port number".) The `Path` (empty in this example) is the Gopher selector for the linked-to directory or file. The Type is the Gopher type character for the linked-to object, which overrides the type inferred from the filename extension mapping table. The `Numb` is a number which is used to control the order in which items appear in the Gopher menu (all numbered items appear, in order, before all unnumbered items).
The case of the keyword to the left of the equals sign is not significant. The order of the lines in the file are not significant.
Here is another example of a link file. This file contains three links (which must be separated by a line containing a single hash (#) character.

```
Name=Gophers in the UK
Host=pcserver2.ed.ac.uk
Port=70
Type=1
Path=1/gophers/our_links
#
Name=Go straight to my data
Ho
```



```
st=+
Port=+
Type=0
Path=0\data\mine\mydata.txt
#
```

```
Name=Gophers in Europe
Host=sunic.sunet.se
Port=70
Type=1
Path=1/Other Gopher and Information Servers/Europe
```
The first and third links in the file point to other Gopher servers. The second link points to a file in the Gopher directory tree on this server.

5.3 Gopher Selectors

In general, the format of the Gopher selector for a file or directory depends on the server software. It is entirely opaque to the Gopher client.
The format of the Gopher selector used by GOPHERS is as follows.

- The first character is the Gopher type of the corresponding item.
- The next character is a backslash.
- Subsequent characters are the pathname of the corresponding file, relative to the Gopher data directory.

For example, consider the Gopher selector used in the example in the previous subsection:
`0\data\mine\mydata.txt`
The first character indicates that the object is a text file. It and the first backslash are removed to obtain the path (relative to the gopher data directory) of the indicated object: the file `mydata.txt` in the directory `mine` which is a subdirectory of `data` which is a subdirectory of the Gopher data directory.
Forward slashes may also be used in selectors - the Gopher Server will convert them to backslashes before analysing the selector.
It is important that when you write a link file (or an alias file - see below) that the type character in the selector (the `Path=` field) is the same as the type character in the `Type=` field.
Note that the format for selectors is different from the format used in GOPHERS version 0.2 (which did not use the leading type character and backslash).

5.4. Aliases

Unless special action is taken, a Gopher client will simply use file and directory names when constructing a Gopher menu display. The Gopher protocol makes provision for the server to transmit to the client a "user-friendly name" - an alias - instead of a file or directory name.
The alias information is stored in an alias directory `ALIAS.GFR` (or `.cap` if UNIX Compatibility Mode is enabled). This directory can exist at each level in the Gopher directory tree. When `gophers` prepares a menu to transmit to the client, for each file in the directory, it will look in the alias directory for a file with a matching name. If it finds such a file, `gophers` assumes that the file is in the format of a link file, as defined above. It will construct a menu entry based on the information in the file.
The `Name` item contains the user-friendly string for display by the client. The `Port` and `Host` items are optional, but if present should indicate the current host and port number (eg using the "+" convention). The `Type` item can be used to override the type deduced from the file's extension. The `Numb` item functions as described above. The `Path` item must be present if the `Type` is present, and it would usually point to the object being aliased.
Here is an example directory tree.

```
SOMEDATA  MYFILE.DAT

          README.TXT

          ALIAS.GFR  MYFILE.DAT
```

Suppose the file SOMEDATA\ALIAS.GFR\MYFILE.DAT looks like this:

```
Name=The data from my experiment
Path=9\SOMEDATA\MYFILE.DAT
Type=9
```

In this situation, when constructing the menu for the SOMEDATA directory, gophers will arrange for the client to display the string "The data from my experiment" instead of the filename MYFILE.DAT. It will also inform the client that the file is binary (type 9) instead of the default type for extension .DAT (which would be '0' - text - unless gophers has been configured otherwise). Because there is no file called README.TXT in the ALIAS.GFR directory, the client will simply display the name README.TXT in the menu.

If the Name= item is present but empty, then the file will not be included in the Gopher menu. This is a good way of "hiding" files so that Gopher clients can't see them.

5.5 Caching

In order to improve performance, the Gopher Server implements a caching strategy. When it creates a menu for a client from a directory, alias files and link files, it stores the menu in a "cache file" in the directory called CACHE.GFR (or, in UNIX Compatibility Mode, .cache). The next time it receives a request for a menu from that directory, it uses the cache file. Caching is particularly useful for improving performance when a directory contains a large number of files and aliases.

Because the contents of directories may change, it is important that the cache files are recreated periodically. If the Gopher Server finds a cache file older than a certain timeout period, it recreates it. The cache file timeout period (which is specified in minutes) is configurable in the Gopher Server Control Panel applet. The default is 5 minutes.

If the cache timeout is set to 0, then cache files are never timed out - if you change anything in the Gopher directory, you must delete the cache file for the change to be visible to clients.

Caching may be enabled and disabled using a check box in the Gopher Server Configuration dialog. NOTE - by default, caching is DISABLED.

5.6. UNIX Compatibility Mode

If you are planning to move a Gopher service from a UNIX platform to Windows NT, you may be interested in the UNIX Compatibility Mode of gophers.

The UNIX Gopher server program gopherd treats any hidden files (with names starting with a dot) in its directory tree as special cases - they are assumed to contain link information, which points at information on another Gopher server. Files called .cache are assumed to contain menu cache information. The gopherd program also treats any .cap directories it finds in a special way - the directory is assumed to contain information about how to display the files in its parent directory.

The Gopher Server for Windows NT offers the same functionality as the gopherd program in these respects, but using different filenames. This is because some filesystems which Windows NT uses (such as FAT) do not support filenames which begin with a dot.

However, to cope with the case where a Gopher directory tree is copied from a UNIX system to a Windows NT system (*eg* using the Windows NT rcp command with the -r flag), gophers has a special "UNIX Compatibility Mode".

In UNIX Compatibility Mode, the gopherd conventions are employed for

`gophers` - the `.cap` directory contains alias information, the `.cache` file is used for cache information, and other files beginning with a dot contain link information. This mode is selected through the Gopher Server configuration dialog in the Control Panel. It is only available if the Gopher Directory is on a volume using a filesystem which supports filenames starting with a dot.
In Normal Mode (the default), `gophers` will not display files or directories to the client if they have names which start with a dot. This mode is mandatory if the Gopher Directory is located on a FAT volume.

6. Searching WAIS Indexes

GOPHERS can search local WAIS databases. This section describes how to configure the Gopher Server to do this. Before reading this section, you should read the WAIS Toolkit for Windows NT manual. (The WAIS Toolkit for Windows NT is available from the same place you obtained this Gopher Server software.)

By default, files with extension `.SRC` are treated as of type 7 (search files). When a Gopher selector indicating a search file is received from a Gopher client, the Gopher Server pass

es the filename (and the search term supplied by the user) to the WAISLOOK program. The output from the WAISLOOK program is passed back to the client.

To ensure that the Gopher Server can execute the WAISLOOK program, the `WAISLOOK.EXE` file should be located in the `\WINNT\SYSTEM32` directory.

There are a number of ways to exploit the search capabilities of the Gopher Server and the WAIS toolkit. The following example indicates one way of using these capabilities.

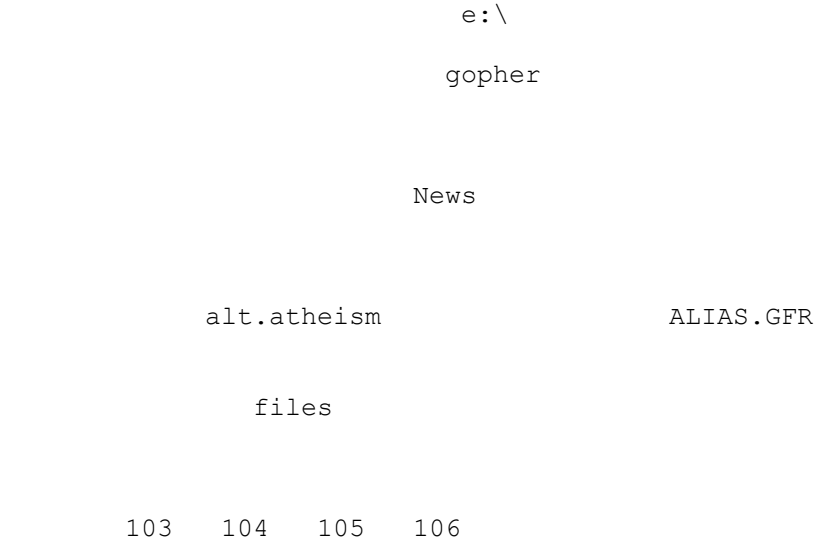
6.1. Example: A Simple Index of Text Files

This example illustrates how to set up a very common situation - you have a large number of text files (eg an archive of Usenet news messages) which you want to allow Gopher clients to search.

Prepare the directory structure

Let's assume that the Gopher data directory is `e:\gopher`, where drive `e:` is on an NTFS partition. Within that directory, you've created a subdirectory `News` to hold a number of newsgroup indexes. Under `News`, you've created a directory `alt.atheism` for that newsgroup. Now, create a directory `files` under `alt.atheism`, which is to hold the actual news messages, one to each file. Also, create a directory `ALIAS.GFR` to hold alias files under the `News` directory.

Here is the situation in graphical form:



Note that the news messages themselves (in this example) have filenames which are simply numbers.

Index the messages

Using a Windows NT command prompt, change to the Gopher data directory

(e:\gopher). This is so that the filenames in the filename table which will be created by the indexing program are relative to the Gopher data directory.
Delete the cache file News\alt.atheism\files\cache.gfr if it exists, and move any link files in News\alt.atheism\files\ out of the way, so that they don't get indexed. Then run the indexing command:

```
waisindex -d News\alt.atheism\index -t netnews News\alt.atheism\files\*
```

This will create files called index.src, index.inv, etc, in the e:\gopher\News\alt.atheism directory.

Test the index

To verify that the index functions correctly, use the WAISLOOK program to examine it:

```
waislook -d News\alt.atheism\index religion
```

This should return a list of files and headlines which contain the word "religion".

Create the alias file

Now, you need to create a menu entry which points to the index. In the e:\gopher\News\ALIAS.GFR directory, create a file alt.atheism which looks like this:

```
Name=Search the alt.atheism news
Host=+
Port=+
Type=7
Path=7\News\alt.atheism\index.src
```

If the Gopher client user selects the News directory from the root of the Gopher directory tree, she will see (instead of the alt.atheism directory) an item in the resulting menu "Search the alt.atheism news". Selecting this will cause the Gopher client to prompt her for a search word. The Gopher Server will invoke the WAISLOOK program to search the index, and will return a Gopher menu containing the list of matching messages.

Updating the index

When a new message (say 107) arrives in the e:\gopher\News\alt.atheism\files directory, you should run the WAISINDEX command to update the index. Change to the e:\gopher directory, and issue the command:

```
waisindex -d News\alt.atheism\index -t netnews -a News\alt.atheism\files\107
```

This will add the new file to the directory. Normally, of course, you would arrange for this command to be executed automatically in a batch file.

7. Troubleshooting

This chapter lists some of the problems which you may have in running the Gopher Server, and describes how to overcome them.

7.1 Errors Starting the Gopher Server

When starting the Gopher Server, you may see one of the following error messages.

Could not start the Gopher Server service on \\yourmachine.
Error 0002: The system cannot find the file specified.

The Service Manager could not locate `GOPHERS.EXE`. This probably means it has been moved, or has not been installed correctly. Remove and reinstall `gophers` - see section 2.4 for details.

Could not start the Gopher Server service on \\yourmachine.
Error 0005: Access is denied.

`GOPHERS.EXE` is inaccessible to the SYSTEM user. By default, the Service Manager starts the Gopher Server process running under a user ID of SYSTEM. The executable file for the service must be readable by this user.

Could not start the Gopher Server service on \\yourmachine.
Error 2140: An internal Windows NT error occurred.

This usually means a problem with the configuration of the Gopher Server. Further information detailing the precise problem will be recorded in the Application Event Log - see the subsection on Error Logging elsewhere in this manual for details on how to view this information. A description of some of the most common errors is given below.

7.2 Errors Recorded in the Event Log

This section records some of the error messages which may appear in the Application Event Log. Many of these are self-explanatory.

Windows Sockets library function "bind" failed. The address or port is already in use.

One of the following situations may give rise to this error:

The TCP/IP port you have specified in the Gopher Server configuration dialog is conflicting with another application. Choose a different port number.

The IP address which the Gopher Server is using is incorrect. Start the Network Control Panel applet, and configure the TCP/IP software to use the correct IP address.

brary function "accept" failed. The call was cancelled.

This indicates that the Gopher Server terminated abnormally for some reason. Restart the service.

7.3 Other Problems

This section describes other common problems in using the Gopher Server, and their solutions.

GOPHERS puts the local IP address instead of the host name into menu items which it returns to clients.

`GOPHERS.EXE` obtains the local host name from the `gethostname()` Windows Sockets call. If the name returned by that call is not a fully-qualified domain name, the Gopher Server uses the IP address instead of the name. Your feedback is sought on how the TCP/IP software in Windows NT can be configured to eliminate this problem (which the author can't duplicate). For a way round, see the `AlternateHostName` registry entry in the Appendix.

Appendix I - Registry Entries

This appendix lists entries added *explicitly* to the Windows NT Registry by the Gopher Server. A number of other entries are added to the Registry implicitly by the Service Control Manager. The information in this appendix is not guaranteed to remain unchanged between releases of the Gopher Server. The information is intended for advanced users of Windows NT who understand the function and structure of the Registry.

The Service Control Manager creates the following entry in the `HKEY_LOCAL_MACHINE` database:

`SYSTEM\CurrentControlSet\Services\GOPHERS`

Under this entry, the Gopher Server itself creates a Parameters key containing the following configuration entries.

Entry name
Entry type
Description
AlternateHostName
REG_SZ
Contains an alternative host name to use when returning Gopher menus.
ExtensionMapping
See below

CacheDir
REG_SZ
.cap or ALIAS.GFR according to UNIX mode setting.
CacheFile
REG_SZ
Name of file to store cache in.
CacheTimeout
REG_DWORD
Cache timeout in minutes
CachingEnabled
REG_DWORD
Non-zero if caching is enabled
DefaultFileType
REG_SZ
Gopher type byte used for files whose extensions are not explicitly mapped.
Directory
REG_SZ
Gopher Data Directory name
LinkFileExtension
REG_SZ
GFR for normal mode, null string for UNIX mode
LogDirectory
REG_SZ
Directory where log files are stored
LoggingEnabled
REG_DWORD
Non-zero if logging is enabled
Port
REG_DWORD
TCP/IP port number

The Extension Mapping entry holds the mapping table. It contains a key name for each file extension, and the value of the key (type REG_SZ) is the corresponding Gopher type byte.
Note that the AlternateHostName entry is not set by the Control Panel applet. It must be entered manually using a registry editor - not something which should be attempted unless you are very familiar with the Registry.