European
Microsoft
Windows NT
Academic
Centre

WAIS Toolkit Version 0.7

COPYRIGHT NOTICE

The software described by this manual is largely based on the "freeWAIS" version 0.3 implementation. The copyright statement relating to the software and this documentation is as follows.

European
Microsoft
Windows NT
Academic
Centre

WAIS Toolkit Version 0.7

Chris Adie
Shangjie Jin

7 December, 1994

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

TABLE OF CONTENTS

1. INTRODUCTION

This manual describes a set of tools for preparing and searching full-text databases for computers running the Windows NT operating system. You should read it if you plan to use the searching capabilities of the Gopher Server (GOPHERS), the HTTP Server (HTTPS), or the WAIS Server (WAISS) for Windows NT. This manual assumes you have a reasonable degree of competence in the use of Windows NT, that you have read the manual for the Server software you plan to use, and that you have some experience of using WAIS (the Wide Area Information Server).

The tools in this toolkit are:

WAISINDEX	An indexing utility.
WAISLOOK	A searching utility.
WAISSERV	A Z39.50 protocol handler and search engine.

This manual covers the beta test version of the WAIS toolkit. Please direct bug reports about this version to `C.J.Adie@ed.ac.uk`.

The European Microsoft Windows NT Academic Centre (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. This manual forms part of the programme of EMWAC.

2. INSTALLATION

2.1. Requirements

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

- Intel, MIPS or Digital Alpha processor.
- Windows NT 3.1 or 3.5, with TCP/IP software installed. (TCP/IP is required by WAISINDEX for the `-export` option.)
- At least 16Mb of memory.

2.2. Installing

1. Log into your Windows NT system.
2. The WAIS Toolkit 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:

`WAISINDX.EXE`	The WAISINDEX program.
`WAISLOOK.EXE`	The searching program.
`WAISSERV.EXE`	The Z39.50 searching program.
`WAISTOOL.DOC`	This manual in Word for Windows format.

`WAISTOOL.WRI`	This manual in Windows Write format.
`WAISTOOL.PS`	This manual, in postscript ready for printing.
`READ.ME`	Summary of new features, etc.

4. If you have installed a previous version of the toolkit, remove it by deleting the old files, or by moving them to another directory (off the PATH) for deletion once you have validated that the new version works correctly.
5. Decide which directory you are going to put the tools in, and move the `.EXE` programs there. Ensure that the directory is on the PATH so that the commands may be executed from the command line. If you plan to use the WAIS Toolkit with the WAIS, Gopher or HTTP servers, you should put the `.EXE` programs into the `\WINNT\SYSTEM32` directory do that the servers can find them.
6. If you are using NTFS for the volume on which the tools are stored, you should rename the `WAISINDX.EXE` program to `WAISINDEX.EXE`. (It is not distributed with that name, because of problems when extracting the file to a FAT volume.) The remainder of this manual assumes you have done this.
7. Determine which version of the toolkit you have. To do this, at the Windows NT Command Prompt, type the commands:
   ```
   waisindex -v
   waislook -v
   waisserv -v
   ```
   and the version number for each program will be displayed. (In fact, two version numbers will be shown for WAISINDEX and WAISSERV - the first refers to the version of the freeWAIS code from which the programs were ported, the second is the number of the Windows NT version.) This manual covers Version 0.7. If the programs report a later version number, you will find a corresponding later manual in the files you unpacked from the ZIP archive.

2.3. Installation Problems

The system says that WAISINDEX.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.

2.4. Deinstalling

To deinstall the toolkit, simply delete the files.

3. USING THE TOOLS

Three programs are provided in the toolkit:

- WAISINDEX is a program which creates a WAIS index of all the words in a set of files. This is ported directly from the CNIDR program of the same name in the "freeWAIS" version 0.3 distribution.
- WAISLOOK is a program which takes one or more words and displays the names of those files in the index which contains those words, ranked according to frequency of occurrence.

- WAISSERV is a program which accepts WAIS protocol requests through stdin and sends back responses using the same protocol through stdout. It is designed for use with the WAIS Server for Windows NT (WAISS), and is of little use on its own.

This chapter documents the above programs. First is a short section describing how to create and search a simple index to verify that the programs are working. In the subsequent sections, the programs are formally documented.

The documentation will be expanded in future releases of this toolkit.

3.1 Creating and Searching a Simple Database

This section describes how to create a simple index using `waisindex`, and how to search it using `waislook`.

Preparation

- Create a directory to work in. Let's assume it's called `C:\TESTWAIS`.
- Create a subdirectory to hold the files we're going to index - say `C:\TESTWAIS\FILES`.
- Put some text files into the `C:\TESTWAIS\FILES` directory. They can be anything you like as long as they are ASCII text files.

Creating an Index

- Make `C:\TESTWAIS` the current directory.
- Execute `waisindex`, giving it parameters as shown below:
  ```
  waisindex -d myindex files\*
  ```
- Observe the messages from `waisindex` to check that there are no errors.
- Do a `DIR` command on the `C:\TESTWAIS` directory to check that waisindex has created the seven index files, named `myindex.*`.

Searching the Index

- Ensure the current directory is `C:\TESTWAIS`.
- Execute `waislook`, giving it parameters as shown below:
  ```
  waislook -d myindex word
  ```
  where `word` should be replaced by a word which you know occurs in the files you have indexed.
- Observe the output of `waislook`, which will show you the names of the files which contain the word you selected.

3.2 The WAISINDEX Program

The `waisindex` program is used to build and update WAIS databases. Note that this program cannot work with a database on a FAT partition, because the intermediate files it creates during the indexing process do not conform to the FAT 8.3 filename restriction.

Syntax

```
waisindex [ -d index_filename ] [ -a ] [ -r ]
      [ -mem mbytes ] [ -register ] [ -export ]
      [ -e [ file ] ] [ -l log_level ]
      [ -pos | -nopos ] [ -nopairs | -pairs ]
      [ -nocat ] [ -T type ] [ -t type ]
      [-filter process] [ -contents | -nocontents ]
      [-v] [-stdin] [-keywords "string"]
      [-keyword_file filename] [-M type,type]
      [-x filename[,...]]
      filename filename ...
```

Description

`waisindex` creates an index of the words in files so that they can be searched quickly by tools such as `waislook`. The index comprises 7 files, and takes about as much disk space as the original text. The files comprising the index have extensions as follows:

`.cat` — The catalogue of the indexed files, with about three lines of information for each file indexed. This is a text file.

`.dct` — The dictionary of indexed words. This is a binary file.

`.doc` — The document table. This is a binary file. A file may contain several documents, depending on the type specified in the `-t` option.

`.fn` — The filename table. This is a binary file. The filenames stored in this table are as supplied as the final parameters to `waisindex`. Thus, if filenames are supplied relative to the current directory (e.g. `files/*`), they will be stored in the filename table in that form, and the resulting filenames from a database search will also be in relative form.

`.hl` — The headline table. This is a binary file. A "headline" is (ideally) a line of descriptive text summarising the contents of a document. The headline is normally taken from the document itself - for instance it may be the `Subject:` line if the document is a mail message, or it may be the first line of the file, or it may simply be the filename itself. Which it is depends on the type of the file, as notified to `waisindex` using the `-t` option.

`.inv` — The inverted file index. This is a binary file.

`.src` — The source description structure. This is a text file.

Options

`-d` *`index_filename`*
This is the base filename for the index files. Therefore if `d:\wais\foo` is specified, then the index files will be called `d:\wais\foo.cat` etc. Default is `.\index`.

`-a` — Append this index to an existing one. Useful for incremental additions or updates. This will only add onto an index, so that if a file has changed, it will get reindexed, but the old entries will not be purged. Therefore, to save space, it is a good idea to reindex the whole set of files periodically. If you don’t specify this option, then the old index (if any) will get overwritten.

`-v` — Display the version number of the program.

`-r` — Recursively index subdirectories.

`-mem` *`mbytes`* How much main memory (in megabytes) to use during indexing. The usefulness of this option in the Windows NT environment is unknown.

`-register` The Windows NT version of `waisindex` cannot automatically register a WAIS database with the directory of servers. Specifying the `-register` option will cause the program to display instructions about how to register a WAIS database manually, using electronic mail.

`-export` This causes the source description file created by `waisindex` to include the host-name and the WAIS default TCP port (210) for use by the clients. Otherwise the source description file contains no connection information, and is expected to be used only for local searches.

`-e [` *`filename`* `]`
Redirect error output to the named file, or suppresses error output if *`filename`* is omitted. Error output defaults to `stderr` (usually the console) if `-e` is not used.

`-l` *`log_level`*
Set logging level. Currently only levels 0, 1, 5 and 10 are meaningful: Level 0 means log nothing (silent). Level 1 logs only errors and warnings (messages of HIGH priority), level 5 logs messages of MEDIUM priority (like indexing filename info). Level 10 logs everything.

`-pos (-nopos)`
Include (don't include - the default) word position information in the index. This will increase the index size, but will allow search engines to do proximity.

`-nopairs (-pairs)`
Don't build (build - the default) word pairs from consecutive capitalized words.

`-nocat` Inhibits the creation of a catalog. This is useful for databases with a large number of documents, as the catalog contains 3 lines per document.

`-contents (-nocontents)`
Include (exclude) the contents of the file from the index. The filename and header will still be indexed. The default is type dependant.

`-filter` *`process`*
Use an external document parser to process the documents. The external parser should be a standalone Windows NT console utility (not tested yet).

`-T` *`type`* The filename table (`.fn`) and the catalog (`.cat`) created by `waisindex` contain a "type" string for each file indexed. This option sets the type string to *`type`*. The default depends on the type of file being indexed - it is `TEXT` in most cases. Possible values are:

`TEXT`
`TEXT-FTP`
`WSRC` (WAIS `.src` structures)
`DVI`
`PS`
`PICT`

GIF
TIFF
HTML

This type information is used only by the WAIS server. The HTTP and Gopher servers have their own mechanisms for determining the type of a file.

`-t` *`type`*
: Tells `waisindex` the type of the files being indexed. The list of recognised types is given below. Default: `text`. This type information allows `waisindex` to derive an appropriate headline, which is stored in the headline table (`.hl`). It is also used to determine whether the files being indexed are deemed to consist of multiple documents.

`-stdin`
: Read the list of filenames to index from standard input (`stdin`), rather than from the command line.

`-keywords "`*`string`*`"`
: Keywords to index for each document.

`-keyword_file` *`filename`*
: File of keywords to index for each document.

`-M` *`type,type`*
: For multi-type documents.

`-x` *`filename`*`[,...]`
: The filename(s) are not indexed. Two or more filenames are separated with a comma and no space between them.

`filename filename...`
: These are the files that will be indexed according to the arguments above. The filenames given here will be stored in the filename table. Wildcards may be used.

The document table size is limited to 16 megabytes. This limits the indexer to databases with headlines that add up to less than 16 megabytes (since that is the principal component of the table). This is typically a problem for database types where a record is essentially a headline (`one_line`, `archie`).

Synonym Files

A synonym file is used to reduce the size of an index and to facilitate more effective searching. It consists of lines of words - the first is the "datum" or basic term, while subsequent words on the line are synonyms. Lines beginning with a hash (#) are treated as comments.

When indexing a database, the synonym file (if it exists) is read into a table. Each word from a document to be indexed is translated using the table to the corresponding datum value, and the translated word is recorded in the database instead of the original word.

When a database search is performed, the search word(s) are similarly translated using the synonym file before the search is performed.

The synonym file has the same name as the database, but must have the extension `.syn`. It must be located in the same directory as the rest of the database files.

If the `waisindex` program does not find a synonym file, it will issue a warning message.

Here is a sample synonym file:

```
# First word is base term, rest are synonyms
boat ship yacht launch galleon destroyer dinghy
shoe slipper boot sneaker trainer
```

File Types

This is the list of types which the `waisindex` program parses. (Further detailed explanation of these types will be included in a later edition of this manual.)

`bibtex`	BibTeX / LaTeX format.
`bio`	Biology abstract format.
`cmapp`	CM applications from Hypercard.
`dash`	Entries separated by a row of dashes. At least twenty dashes must be present in order for a line to be recognised as a separator. Each entry is indexed as a separate document.
`dvi`	DVI format.
`emacsinfo`	The GNU documentation system.
`first_line`	First line of file is headline.
`filename`	Uses only the filename part of the pathname for the title.
`ftp`	Special type for FTP files. First line of file is headline.
`gif`	GIF files, only indexes the filename.
`html`	Hypertext Markup Language (HTML). The text within the `<TITLE>` element is the headline.
`irg`	Internet resource guide.
`jargon`	Jargon File 2.9.8 format.
`listserv_digest`	LISTSERV mail digest format.
`mail_digest`	Standard Internet mail digest format.
`mail_or_rmail`	Mail or rmail or both.
`medline`	Medline format.
`mh_bboard`	MH bulletin board format.
`ms_kbase`	MS Knowledge Base format.
`netnews`	Netnews format.
`nhyp`	Hyper text format, Polytechnic of Central London.
`one_line`	Each line in the file is a separate document.
`para`	Paragraphs separated by blank lines. Each paragraph is a separate document.
`pict`	Pict files, only indexes the filename.
`ps`	Postscript format.
`refer`	Refer format.
`rn`	Netnews saved by the [rt]?rn newsreader.

`server` Server structures (`.src`) for the directory of servers.

`text` Simple text files. (This is the default.)

`tiff` Tiff files, only indexes the filename.

`URL what-to-trim what-to-add`
This type has been superseded by the `html` type, which should be used in preference.

`object` A structured object.

`inriadoc` INRIA library catalog.

`paradoc` INRIA library catalog para-mode.

`fortran` Fortran files, needs an external document parser.

`mime` Like mail

`oneline_phonix`
Uses phonix matching algorithm to search the documents. For example, phone books with names (and phone numbers) on each line of a document.

`oneline_soundex`
Uses soundex matching algorithm to search the documents. For example, phone books with names (and phone numbers) on each line of a document.

3.3 The WAISLOOK Program

The `waislook` program is used to search WAIS databases. It is executed automatically by the GOPHERS and HTTPS servers when they need to search WAIS databases, but it may also be executed manually from the console. In the latter case, many of the options listed are not relevant.

Syntax

```
waislook [-d dbname] [-h hostname] [-p port]
      [-debug] [-v] [-http¦-gopher] [-t title]
      [-q virtpath] search words ...
```

Description

This program searches an index for documents which contain the search words. It ranks documents according to the frequency of occurrence of the words, and according to whether they occur in the document headline. If more than (by default) 40 documents are found, only the 40 with the topmost ranking are returned.

The program supports boolean searches. The boolean operators are *and*, *or*, and *not*. Nesting boolean expression using brackets is not supported. Words are evaluated from left to right in the wais query string. When a *not* operator is found, the following single word is moved into a buffer of not-words. This not-word buffer is evaluated after all the other words are evaluated. If a document matches a not-word, that document is removed from the set of matches (given a score of zero). When an *and* word is found, the word following it is checked for matches to documents, and the set of documents matching this and-word is compared to the set of documents matching any words prior to the and-word. The intersection of prior and current matching documents is retained, others are removed (set to a zero score). When an *or* word is found, the word following it is checked for matches to documents. The union of prior and current matching documents is retained.

The program supports matching based on soundex and phonix matching. If the index was created with the `-t oneline_soundex` or `-t oneline_phonix` arguments, then a search for `soundex` *`term`* or `phonix` *`term`* respectively should return the documents which match *`term`* according to soundex or phonix rules. (Note - this facility is untested in this implementation.)

The program generates either an HTML document or a Gopher menu containing the result of the search, or else displays the names of the documents and their corresponding headlines on the console.

Options

`-debug`	Enable debugging. In this mode, debugging information is send to `stderr`.
`-v`	Display the version number of WAISLOOK.
`-h` *`hostname`*	Specifies the name of the host to quote when generating HTML output or Gopher menu output. Not used in interactive mode. No default value.
`-p` *`port`*	Specifies the number of the TCP/IP port to quote when generating HTML output or Gopher menu output. Not used in interactive mode. No default value.
`-d` *`dbname`*	Specifies the name of the WAIS database to search. The name should not have an extension or a trailing dot. Defaults to `.\index`. It is almost always necessary to use this option.
`-http`	Specifies that the program has been invoked from the HTTP Server and should output the results of the search in HTML. May not be combined with `-gopher`.
`-gopher`	Specifies that the program has been invoked from the Gopher Server and should output the results of the search as a Gopher menu. The Gopher type for each matching file is determined from the EMWAC Gopher Server (GOPHERS) configuration information stored in the Windows NT Registry. May not be combined with `-http`.
`-t` *`title`*	Specifies the title to use in the output HTML document if the `-http` option has been selected. If the title contains spaces, enclose it in double quotes.
`-q` *`virtpath`*	This option lets you specify a virtual path name to prepend to the filenames returned by WAISLOOK when the `-https` option is in effect. This option may be used by some versions of the HTTP Server for Windows NT. Note that the freeware HTTP server does not support virtual paths.
`search words ...`	One or more search words are specified after all the options. The first search word may not begin with a hyphen (to distinguish it from the options). If more than one search word is given, documents which contain any of the search words will be returned. Note that boolean combinations of search words are not (yet) supported.

3.4 The WAISSERV Program

The `waisserv` program is used to search WAIS databases. It is executed automatically by the WAIS Server (WAISS) when it receives an incoming call from a WAIS client. It may also be executed manually from the console, but is not particularly useful in this mode.

Syntax

waisserv [-d *directory*] [-e *file*] [-v] [-l *level*]

Description

This program reads WAIS protocol requests from its standard input (`stdin`) and writes the response to standard output (`stdout`). Like `waislook`, it ranks the documents it finds according to the frequency of occurrence of the words, and according to whether they occur in the document headline. If more than 40 documents are found, only the 40 with the topmost ranking are returned.

The program supports boolean searches. The boolean operators are *and*, *or*, and *not*. Nesting boolean expression using brackets is not supported. Words are evaluated from left to right in the wais query string. When a *not* operator is found, the following single word is moved into a buffer of not-words. This not-word buffer is evaluated after all the other words are evaluated. If a document matches a not-word, that document is removed from the set of matches (given a score of zero). When an *and* word is found, the word following it is checked for matches to documents, and the set of documents matching this and-word is compared to the set of documents matching any words prior to the and-word. The intersection of prior and current matching documents is retained, others are removed (set to a zero score). When an *or* word is found, the word following it is checked for matches to documents. The union of prior and current matching documents is retained.

The program supports matching based on soundex and phonix matching. If the index was created with the `-t oneline_soundex` or `-t oneline_phonix` arguments, then a search for `soundex` *`term`* or `phonix` *`term`* respectively should return the documents which match *`term`* according to soundex or phonix rules. (Note - this facility is untested in this implementation.)

Options

`-v` Display the version number of the program.

`-d` *`directory`*
Specifies the directory containing the WAIS databases. The name should not have an extension or a trailing dot. Defaults to the current directory.

`-e` *`file`* Specifies that log information should be written to *`file`*. Defaults to `NUL:`.

`-l` *`level`* Specifies the amount of logging information to write to the file. The *`level`* is a number from 0 (no logging information - the default) to 10 (full information).