GUIDE® Indexer™

User's Manual for GUIDE® Author™

InfoAccess™
INCORPORATED

GUIDE® Indexer™ User's Manual

InfoAccess Inc.

15821 NE 8th St
Bellevue, WA 98008-3905
USA

Technical Support

Phone	425-201-1916
Email	techsupp@infoaccess.com

Corporate Headquarters

Phone	425-201-1915
Sales	800-344-9737
Fax	425-201-1922
Web	www.infoaccess.com
Email	info@infoaccess.com

MAN5000-04B

Contents

4 USING GUIDE INDEXER

CHAPTER 1

WELCOME TO GUIDE INDEXER

GUIDE® IndexerTM creates full text indexes that record every significant word in every document of *GUIDE* electronic publications. Your readers can use *GUIDE ReaderTM* to view the distributed publications and can quickly search for words or logical combinations of words across all the documents in a given publication.

Full text indexes differ from a 'key word' indexes in several ways. A key word index (the traditional index at the back of a book or in an online help document) contains only words and terms specifically marked for inclusion in the index. In contrast, a full text index automatically records every significant words in the indexed publication, omitting only words it makes no sense to include: "an", "but", "or", etc.

When you distribute *GUIDE* electronic documents and full text index files with *GUIDE Reader*, your readers can create full text queries to search for any word or term they choose, even multiple words and terms. This provides readers with a fast, easy way to search through huge collections of documents. When a reader runs a query, *GUIDE Reader* not only records 'hits' that exactly match the text typed into a query text box, it also recognizes plurals and possessives and finds those occurrences as well. For example, a search for "query" would identify not only every occurrence of "query", but also all instances of the word in its possessive and plural forms ("query's" and "queries").

Creating Queries in GUIDE Reader

To search a publication with related full text index files, readers open the **Query** dialog in *GUIDE Reader* and then enter the words or terms they want to search for. A 'word wheel' turns to show matching words as readers type their query into a text box. For example, as the reader enters 'typesetter', the word wheel first turns to the first word in the full text index that starts with the letter 't' and highlights that word. As the reader continues to type, the word wheel turns progressively to highlight 'type' and then 'typeset,' assuming these words are in the index. The number to the right of each word is the number of occurrences or 'hits' of that word in the document collection.

With the word or term selected, the reader can click on **Run query** to search the publication. The **Search Results Hitlist** dialog displays a list of the documents in the indexed publication that contain the words sought, as well as the number of hits (including synonyms) for each document. Readers can click on a document title to open that document. All the hits are highlighted, and readers can use the **Hits** palette to move from hit to hit through the document.

The **Query** dialog offers sophisticated searching options. Boolean operators allow such refined queries as **(windshield OR wipers) AND glass**, which would result in *GUIDE Reader* locating every document that contains the word 'glass' and also 'windshield' or 'wipers.' Using the **NEAR** operator (as in **truck NEAR Europe**) would locate words or terms that appear in the same paragraph. Parentheses allow you to nest subexpressions in queries, as in **(((windshield AND wipers) OR glass) AND truck)**.

Numerals as well as letters can be entered in the **Query Text** box, permitting searches for numbers as well as words. (To include numbers in the word wheel, set the **Numbers In Word Wheel=** entry in the initialization file to **1**, the default; otherwise, set it to **0**.)

GUIDE Reader also provides tools for managing queries. From the **Query** dialog, readers can name and save queries, and create query files that can be used in future searches, even on different publications. Readers can also print queries, if they want. Please see the *GUIDE Reader User's Manual,* Chapter 2, for a fuller explanation of how these features enable readers to create and manage queries.

One especially useful *GUIDE Reader* feature is its ability to allow fielded search, which improves the efficiency of searches. Readers can mark portions of text in a publication as fields, and then choose to confine their search to those fields. This reduces the amount of data that has to be searched and improves readers' productivity. Please see the *GUIDE Author User's Manual,* Chapter 5, for an explanation of using Objects for fielded search.

About this Manual

This manual describes how to use *GUIDE Indexer.* This chapter introduces *GUIDE Indexer* and its functionality. Chapter 2 explains how to organize *GUIDE* documents into collections for full text indexing so that readers' full text queries run smoothly in *GUIDE Reader.* Chapter 2 also discusses proximity parameters (which are addressed in even more detail in Chapter 3); how indexing performance is affected by the computer hardware used for indexing; and what you need to do if you want to run *GUIDE Indexer* from a network drive.

Chapter 3 provides a detailed discussion of three files that affect how your document collection is indexed: stop, thesaurus, and term variants. Chapter 4 explains how to start *GUIDE Indexer* and how to use the application's menus, commands, and dialog options. Chapter 4 also explains the actual indexing process and describes the files that *GUIDE Indexer* creates.

CHAPTER 2

BEFORE YOU INDEX

GUIDE Indexer installs automatically with *GUIDE Author.* If the *GUIDE Indexer* icon appears in the *GUIDE Author* program group, the software has been installed successfully. Before you begin to use *GUIDE Indexer* to create indexes, you should ensure that the finished *GUIDE* documents that you want to create full text indexes for reside on your system.

This manual assumes that you are familiar with Microsoft® Windows and *GUIDE* publishing tools. If necessary, please refer to the documentation provided with those products for further information.

Organizing Document Collections

It's important to finalize the document collection and its structure before you create a full text index for a *GUIDE* electronic publication. A document collection consists of the *GUIDE* files that contain the 'body' of information you want to distribute; these files are also called 'body' or 'content' documents to differentiate them from table of contents, index, and control panel documents.

A full text index will not be accurate or complete if you change or add body documents to your publication after you index the collection. Moreover, if you move document files in the directory structure or rename them, full text index queries and links between *GUIDE* Objects may not work properly in *GUIDE Reader.*

The best way to ensure that *GUIDE Reader* can find referenced files is to organize all the files for your document collection in one directory.

There are two alternative approaches, but each has its drawbacks. You can:

- Edit the initialization file to include a path entry that lists all directories that contain *GUIDE* documents. Unfortunately, this forces *GUIDE Reader* to search through multiple directories. Also, because drive letter assignments vary, readers must modify their own initialization files — and many readers may lack the skills to make these changes independently. If you choose this method, you also need to provide documentation that specifies the path for indexed files.
- Turn on the **Full Path References** and **Make Default** options in *GUIDE Author's* **Document Properties** dialog before you author *GUIDE* documents to include full path names in references. But when you do this, *GUIDE Author* hard-codes the letter of the active drive into all interdocument Reference Buttons. As a result, reference links to these documents can be found only if the letter designation of the drive a reader uses happens to correspond to the letter designation of the drive where you created the documents. Since there's no way to ensure this, we recommend you avoid this method unless it's absolutely necessary.

For more information about document collections and index files, please see the *GUIDE Writer User's Manual,* Chapter 2.

Initialization Settings that Affect Indexes

Several sections of the **infacces.ini** initialization file can help you manage indexes in *GUIDE Author. GUIDE Indexer* refers to the entries in the **[gindexer]** section to create full text indexes. These entries are not required for *GUIDE Reader* queries, so you can delete the **[gindexer]** section from initialization files you distribute with *GUIDE* publications.

The entry **use_index_names=1** under the **[fulltext]** section directs *GUIDE Reader* to the **[IndexNames]** section and instructs it to display the index names listed there in the **Select Index** dialog. This allows users to use an index by an assigned name in the **Query** dialog. For example, if you create the index **index.idx** but prefer to display its assigned name **Facts on File,** you can ensure this by entering **Facts on File=index.idx** in the **[IndexNames]** section.

A fourth section, **[IndexDocuments]**, allows you to link an index on one drive to its related documents on another. For example, the entry **c:\gppindex\index.idx=f:\alldocs\corp\** would allow a search of files on the **f:** drive from the index **index.idx** on the **c:** drive. This would allow you to move a set of index files to a fast drive while leaving the documents on CD-ROM.

For a more detailed discussion of initialization settings, please see Chapter 2 in *Welcome to GUIDE Author.*

Be careful about changing the default parameters of these entries:

stop file=gindexer.stp
thesaurus file=gindexer.fth
variants file=gindexer.ftl
number of paragraphs=2000
maximum paragraph size=4000
Numbers In Word Wheel=1
Use Advanced Language Option=0
ALO Character Set=JAPAN_90_SJS_ASC
ALO Normalization=JAPANESE
ALO Parser=ftsjp

The last four entries are for the full text search of Japanese characters, including single-byte Katakana. To support full text search of Japanese characters, set **Use Advanced Language Option=** to **1**, and set the three "ALO" entries as shown. The default setting **Use Advanced Language Option=0** provides no support for the Japanese character set, in which case the settings for the three "ALO" entries have no effect.

The three ALO indexing parameters perform the following functions:

ALO Character Set=	Specifies which ALO character set to use (Japanese 90 Shift-JIS for Windows)
ALO Normalization=	Specifies which (case) normalization rules to follow during indexing.
ALO Parser=	Specifies which language parser to use while reading the source document. This parser translates the document's characters into an internal Fultext format for Japanese.

Proximity Parameters

GUIDE Indexer creates indexes based on the positions of characters within documents. These character positions are stored in the index and can be returned in response to queries. To conduct proximity searching, *GUIDE Reader* must be able to tell whether groups of characters reside in the same paragraph.

GUIDE Indexer uses two entries in the **[gindexer]** section of the initialization file to determine character positions. One entry specifies the maximum number of paragraphs that a single document can contain; the other defines the maximum number of characters any single paragraph may contain (maximum paragraph size). The relevant entries are:

```
number of paragraphs=2000
maximum paragraph size=4000
```

The defaults are 2,000 paragraphs per document and 4,000 characters per paragraph. Since the maximum number of characters *GUIDE Indexer* can handle during the indexing process is about 16 million, you should ensure that the following formula remains true before you change either of these parameters:

(Max Paragraph Size x 2) x No. of Paragraphs <= 16,000,000

Note: The maximum document size that can be indexed is 2 GB (but only the first 16 million characters are considered).

If you change the defaults, the value you choose for maximum paragraph size should be as small as possible, yet large enough that no paragraph in any *GUIDE* document you index ever exceeds that number of characters.

NOTE:

GUIDE Indexer interprets most GUIDE Objects as paragraphs.

GUIDE Indexer Performance

GUIDE Indexer initiates a batch process to create index files for large document collections. How long this batch process takes depends largely on the hardware configuration you're using to run *GUIDE Indexer:*

HARDWARE CONFIGURATION	AVERAGE INDEX SPEED
486/66 MHz	66 MB per hour
Pentium™/90 MHz/16 MB RAM	120 MB per hour

Based on these average speeds, *GUIDE Indexer* requires about 15 minutes to create full text index files for a 16 MB document collection on a 486 machine and about eight minutes on a Pentium computer (90 MHz).

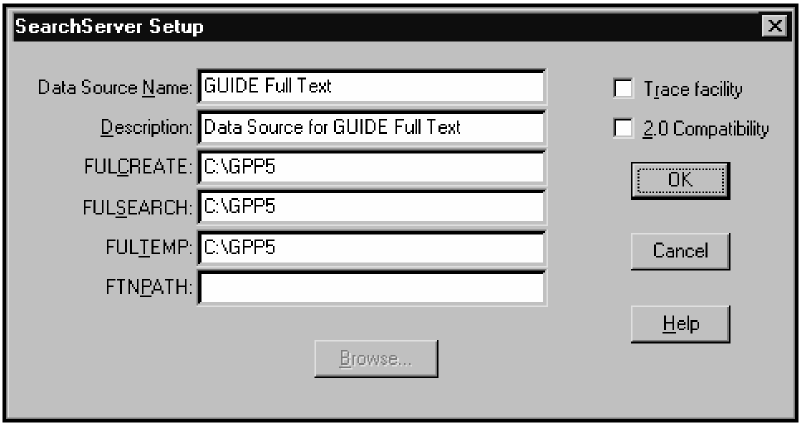
To improve *GUIDE Indexer's* performance, turn off your screen saver and shut down all other applications before you start the batch process. If you're concerned about monitor burn-in, dim your monitor or shut it off while *GUIDE Indexer* conducts its batch process.

Using GUIDE Indexer on a Network

When you installed *GUIDE Author*, the installation utility set up an ODBC data source for *GUIDE Indexer*. The data source is used by *GUIDE Indexer* to index your GUIDE documents. If you accepted the default directory, the path **c:\guide** was used for the ODBC data source. You can check this by opening the 32bit ODBC Setup dialog from the Windows Control Panel.

1. **From the Start menu, click Settings and then Control Panel.**
2. **In the Control Panel dialog, double-click the 32bit ODBC icon to open the Data Sources dialog.**
3. **Double-click GUIDE Full Text (SearchServer_3.0 Driver(*.cfg)).**

4 In the SearchServer Setup dialog (see Figure 2-1) there are three text boxes that contain current path information: FULCREATE, FULSEARCH, and FULTEMP. The path is the same in each case.



Data Source Name: GUIDE Full Text
Description: Data Source for GUIDE Full Text
FULCREATE: C:\GPP5
FULSEARCH: C:\GPP5
FULTEMP: C:\GPP5
FTNPATH:

□ Trace facility
□ 2.0 Compatibility

Figure 2-1
The SearchSaver Setup dialog box

When you install *GUIDE Reader* on client machines for your users, you must ensure that your install program sets up the ODBC data source so that those running the *GUIDE* publications you distribute can access the indexes that belong with those publications. So if you're using *GUIDE Indexer* on a network, you will need to ensure that the installer sets FULCREATE, FULSEARCH, and FULTEMP to the network path you want to use. In this way, GUIDE Indexer will be able to index the files when you run full text index queries.

CHAPTER 3

DETERMINING INDEX CONTENT

Before you run *GUIDE Indexer*, you should decide how you want to index your document collection. You can tailor the index to end users' needs by editing three files that were installed automatically when you installed *GUIDE Author:* the stop, thesaurus, and term variants files. You'll find these files in **c:\guide** (assuming you installed *GUIDE Author* in the default directory).

The stop file, **gindexer.stp,** determines how your document collection is indexed by specifying words you don't want to include in the index: "an", "the", etc. The thesaurus file, **gindexer.fth,** and term variants file, **gindexer.ftl**, influence the searching process that takes place in *GUIDE Reader*. You can edit the content of all three files in any text editor. In the case of the thesaurus file, edit the uncompiled version, **gindexer.fts**, and use it to compile a new **gindexer.fth** file.

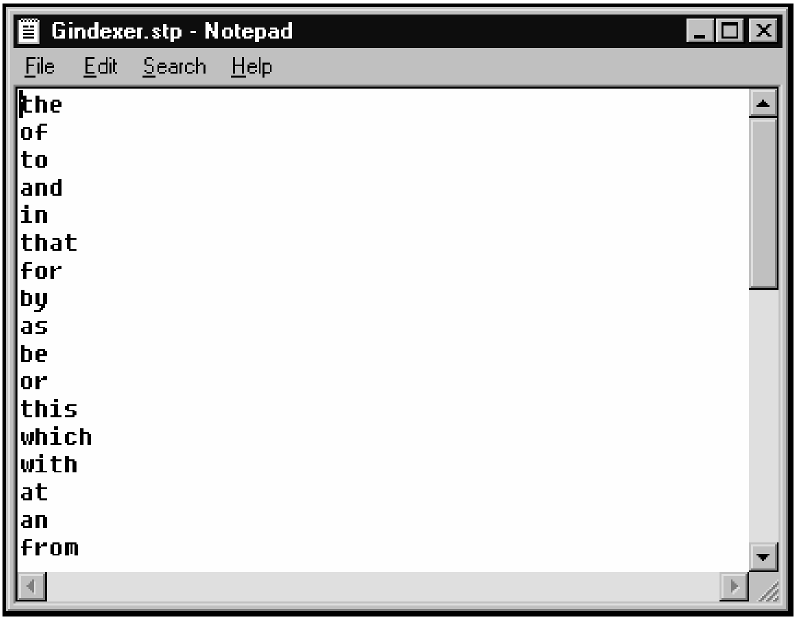
Stop File

The stop file, **gindexer.stp,** associated with your document collection contains a stop list that is simply a list of words that should *not* be indexed, which is usually those words that occur too frequently to be of value for search purposes ('an,' 'the,' etc.). To ensure consistent search results, *GUIDE Reader* follows the instructions in the stop file and ignores those words in the stop file when it searches. The default stop list provided with **gindexer.stp** can be found at the end of Chapter 4.

Depending on the language version of *GUIDE Author* you are using, the stop file may also contain instructions on how to index the collection. If so, these appear at the beginning of the file, followed by the line **STOPLIST=** and then the stop list itself. Do *not* change these instructions. Edit only the stop list (only that portion of the stop file that follows the **STOPLIST=** line). A duplicate (read-only) stop file named **master.stp** was installed with *GUIDE Author* to provides a backup in case **gindexer.stp** is ever damaged.

To customize stop files for indexing, open **gindexer.stp** in your text editor, edit the stop list, and use **Save As** to save the file under a new name. Be sure to save any new files in the same directory as the application executable file, **gindexer.exe** (**c:\guide** is the default directory).

Note: The file size limitation for a stop file is 1,024 words or 10,000 characters, whichever is smaller.



```
the
of
to
and
in
that
for
by
as
be
or
this
which
with
at
an
from
```

Figure 3-1
The stop file showing the stop list

GUIDE Indexer uses only one stop file at a time, specified by the entry in the **[gindexer]** section of the initialization file. The default entry is **stop file=gindexer.stp**. You can change the entry any time you want to use another stop file for a particular indexing session.

Whichever stop file is referenced in the initialization file will be used on all future indexing sessions until you change the entry. If you use different stop files for indexing different document collections, you must track these files and be sure to use the appropriate file if you re-index a particular document collection.

Editing Stop Lists

To edit a stop file in *GUIDE Indexer:*

1. **Open gindexer.stp (or whichever stop file you want to edit) in any text editor.**

2. **Add any words to the stop list (after the 'STOPLIST=' line, if there is one) that you want to exclude from the final index, and delete any words that you now want to include. Type each word on a line by itself.**

3. **Click on *Save* to write your changes to the file or choose *Save As* to save the file under a different name.**

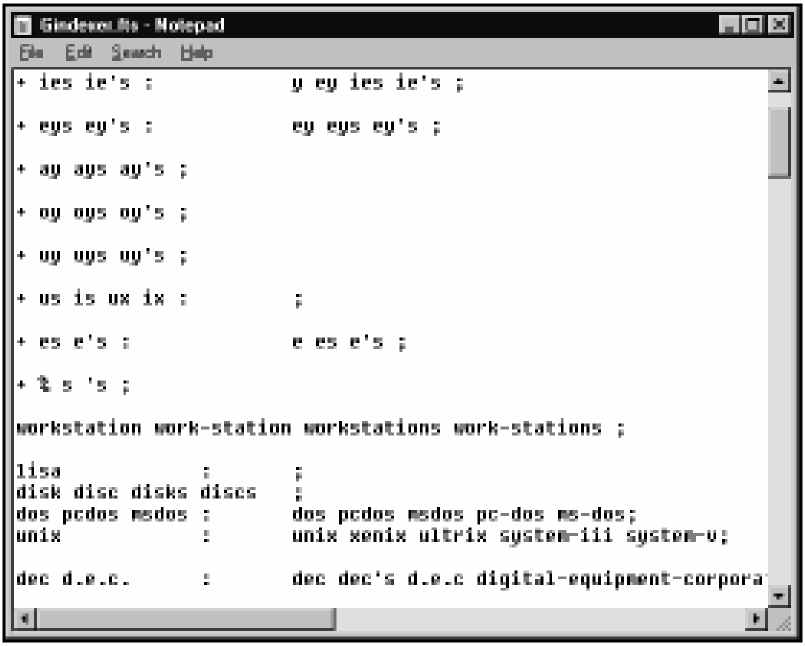
Thesaurus File

The thesaurus file, **gindexer.fth**, contains guidelines that *GUIDE Reader* uses to generate plural and possessive variants of search terms, long forms of some abbreviations, and selected synonyms. To revise this file, you must edit the uncompiled version of the thesaurus, **gindexer.fts**, and compile a new **gindexer.fth** file with the **FTHMAKE** utility supplied with *GUIDE Author*.

To edit a thesaurus file in *GUIDE Indexer*:

1. **Open gindexer.fts in any text editor (the file is in the directory where *GUIDE Indexer* was installed).**

2. **Edit the file, as appropriate, and save it to the same directory.**



```
+ ies ie's ;            y ey ies ie's ;

+ eys ey's ;            ey eys ey's ;

+ ay ays ay's ;

+ oy oys oy's ;

+ uy uys uy's ;

+ us is ux ix ;         ;

+ es e's ;              e es e's ;

+ % s 's ;

workstation work-station workstations work-stations ;

lisa            :       ;
disk disc disks discs   ;
dos pcdos msdos :       dos pcdos msdos pc-dos ms-dos;
unix            :       unix xenix ultrix system-iii system-v;

dec d.e.c.      :       dec dec's d.e.c digital-equipment-corpora
```

Figure 3-2
The thesaurus file displayed in Notepad for editing

Thesaurus Rules

A thesaurus file is really a list of rules. Each rule has a left side and a right side, separated by a colon (**:**) and ending with a semi-colon (**;**). The left side of a rule contains words or suffixes to be matched when a search term is sought in the thesaurus. The right side contains additional words and phrases (synonyms) or suffixes (plurals and possessives) that should also be recorded during the search. When a match is made with one of the entries on the left side of a rule, the alternatives from the right side, or substitutions formed by combining the original word stem with each of the alternative suffixes from the right side, are used for the search in addition to the original term.

White spaces separates words and suffixes, hyphens join phrases, and rules may span more than one line. If the colon separator and the right side alternative are missing, *GUIDE Indexer* assumes that the right side is the same as the left side (true equivalence). If the colon is present but the right side of the rule is missing, no alternatives are generated and the original term remains the same.

Synonym Rules

Synonym rules contain a list of words on the left side and a list of words or phrases, if applicable, on the right side. A phrase on the right side is denoted by a hyphen (**-**) or any other punctuation that joins its constituent words. During a search, thesaurus synonym rules take precedence over the suffix rules; a match between a search term and a word on the left side of a synonym rule prevents any suffix processing for that term, whether or not any alternatives were generated.

Plurals, possessives, or other alternatives that should be derived from the terms on the left side should be included on the right side of the rules. If the same word appears on the left side of more than one rule, a synonym search for that word generates a combined list of alternatives from the right side of all the matching rules.

Suffix Rules

A plus sign (**+**) as the first character distinguishes a suffix rule. The left side and right side of these rules contain lists of suffixes separated by white space; the right side is optional. The percent symbol (**%**) may be used to represent a null suffix. Suffix searching proceeds so that the longest suffix on the left side of all suffix rules is matched. The percent symbol represents the suffix of last resort and should be used on the left side of only one rule.

The *GUIDE Reader* search engine applies certain restrictions to the way it looks for search terms in the current thesaurus at search time. The restrictions are:

- Never seek words that are in the stop file
- Only find individual search terms, including words or phrases with embedded punctuation (for example, **F.2D**), but exclude word roots and any words generated by a root expansion as well as phrases that contain embedded spaces
- Only report alphabetic words with more than one letter

Since alternatives produced by the suffix rules are not likely to occur in any document, this type of rule is not strictly necessary. However, such rules can improve search performance because they prevent *GUIDE Indexer* from generating alternatives that otherwise would have to be looked up in the index files. If those words are included in the stop files associated with all collections, the rule is redundant.

Sample Rules

The thesaurus file includes the following sample rules (an explanation of its function follows each rule). The first examples are suffix rules, which, by convention, usually appear first in a thesaurus source file:

+ y : y ies y’s ;	**Pony** produces the alternative list **pony**, **ponies**, **pony’s**
+ us is ux ix : ;	Greek suffixes are not transformed at all; they are nearly impossible to do reasonably
+ % s ’s ;	**Pit**, **pits** or **pit’s** produces all three forms

Note that these rules don’t include the suffixes **s’** or **ies’**. Since the standard character classes associated with indexing ignore trailing apostrophes for indexing purposes, a search for ‘ponies’ retrieves **ponies’** and vice versa (except in a phrase). As a result, you don’t need to include normal plural possessive suffixes in the thesaurus. Table 3-1 illustrates various forms of synonym rules.

TABLE 3-1 — SYNONYM RULES

d.e.c dec dec’s:	**d.e.c. dec dec’s** **digital-equipment-corp** **digital-equipment-corporation** **digital-equipment-corporation’s**	**d.e.c** or **dec** produce alternatives **d.e.c**, **dec**, **dec’s** or various longer forms
dec december;		**dec** also produces **december**
one 1 ; **first 1st ;**		**one** or **1** produce both forms; similarly for **first** or **1st**
monkey monkeys monkey’s ;		**monkey** produces **monkey**, **monkeys** or **monkey’s**; this rule overrides the **+y...** suffix rule, which would produce **monkey**, **monkeies** or **monkeies’s**
whereas wherefore: ;		**whereas** and **wherefore** have no alternative forms

Compiling a Thesaurus File

If you want to change the thesaurus file, **gindexer.fth**, you must edit the uncompiled version, **gindexer.fts** (explained earlier), and recompile that file by using the **FTHMAKE** DOS utility supplied with *GUIDE Author*.

The executable for the utility, **fthmake.exe**, should be in the same directory as the executable for *GUIDE Indexer*. The uncompiled thesaurus file, **gindexer.fts,** can be in any directory. If it's not in the same directory as **fthmake.exe**, you'll have to provide the full path for the file. Likewise, you can provide a full path for the compiled thesaurus if you want to place it in a directory other than the one where the utility resides.

You can compile the thesaurus file from either a DOS prompt or the command line box in the **fthmake.exe** dialog. To compile from DOS:

1 **At a DOS prompt, change directories to c:\gpp5 (or whichever directory the files are in).**

2 **Type: fthmake gindexer.fts gindexer.fth**

where **gindexer.fts** is the uncompiled thesaurus file supplied with *GUIDE Author* and **gindexer.fth** is the thesaurus file you want to create to replace the one supplied with *GUIDE Author.* If you want to keep the original thesaurus file, use another name for the new file.

FTHMAKE compiles the new thesaurus and places it in the same directory as the utility. If you have given the thesaurus a new name and now want to use this file to generate an index, you must change the **thesaurus file=** setting in the **infacces.ini** file.

To compile from the command line in FTHMAKE:

1 **Double click on *fthmake.exe* in Microsoft Explorer or the File Manager to open the fthmake.exe dialog.**

2 **In the Parameters box, type: gindexer.fts gindexer.fth**

Again, **FTHMAKE** compiles the new thesaurus and places it in the same directory as the utility. If you have given the thesaurus a new name and now want to use this file to generate an index, you must change the **thesaurus file=** setting in the **infacces.ini** file.

Testing a Thesaurus File

Once you have compiled **gindexer.fth**, you should test the thesaurus file to be sure it provides the results you expect. You can do this by running the **FTHTEST** DOS utility supplied with *GUIDE Author.*

The executable for the utility, **fthtest.exe,** should be in the same directory as the executable for *GUIDE Indexer*. The thesaurus file, **gindexer.fth,** can be in any directory. If it is not in the same directory as **fthtest.exe**, you'll have to provide the full path for the file.

Before you run **FTHTEST**, you may want to review the terms in the thesaurus file by opening the uncompiled version, **gindexer.fts**, in any text editor. You can test the thesaurus file from either a DOS prompt or the command line box in the **fthtest.exe** dialog.

To test from DOS:

1 **At a DOS prompt, change directories to c:\gpp5 (or whichever directory the files are in).**

2 **Type *fthtest gindexer.fth* where *gindexer.fth* is the thesaurus file you want to test.**

You're prompted to enter a term.

3 **Enter the term you want to test; for example, *pound*.**

FTHTEST displays all the synonyms in the thesaurus:

```
Synonym:
    pound
    pounds
    lb
    lbs
```

FTHTEST follows the list of synonyms with a prompt to enter another term. You can go on entering terms in this way to test the thesaurus.

4 **To exit FTHTEST after you've tested the thesaurus, press *Ctrl+Z* followed by the *Enter* key.**

Alternatively, you can test several terms at once. For example, if you know that **pound**, **disk**, and **ton** are in the thesaurus, you could type the following at the MS-DOS prompt:

```
fthtest gindexer.fth pound disk ton
```

FTHTEST will list the synonyms for each term, in turn.

To test from the command line in FTHTEST:

1 **Double click on *fthtest.exe* or *fthtest* (the PIF file) in Microsoft Explorer or the File Manager to open the *fthtest.exe* dialog.**

2 **In the Parameters box, type *gindexer.fth*, and then follow steps 3 and 4 above.**

Again, **FTHTEST** tests the thesaurus file and places it in the same directory as the utility.

Term Variants File

The term variants file, **gindexer.ftl,** allows typographical variants of the same word to be treated equivalently for search purposes. This file contains character substitution rules that control how *GUIDE Reader* generates variations on a user's search terms. If the search engine cannot read the file, this feature is disabled without warning; the search engine still attempts to find the search term, but it generates no variant forms.

The character substitution rules in the term variants file are defined by a new-line character (**x0A**) or an end-of-file character (**EOF**). Each rule has three fields:

Opcode	One character that indicates the type of substitution
Target	The substring to be matched and replaced
Replacement	The substring to substituted for the target

These fields must conform to the format outlined below:

FIELD	*START COLUMN*	*LENGTH*	*POSSIBLE VALUES*
Opcode	1	1	":" or "%"
Target	2	<=4	Any short string
Replacement	6	<=4	Any short string
End of Rule	6-10	1	New line of end of file character

Spaces delimit the target and replacement strings if they occupy less than four characters. In addition, the replacements field may end at a new line or an end of file character. The search engine may reject the query if you deviate from this format.

A rule applies to a given word if the target substring is matched in accordance with the type of rule, as indicated by these opcodes:

:	Perform substitution anywhere within the original word (context-free matching target)
%	Perform substitution only at the end of a word (suffix matching target)

For context-free matching, the target field cannot be empty. A suffix matching rule may have an empty target, in which case every original term generates a variant with the replacement string as a suffix. An empty replacement field is always permitted.

Although context-free rules apply to the stem of an expansion term (root expansion), suffix rules do not assume that the expanded list of terms includes any suffix variants. In addition, suffix rules apply only to the last component of an implied phrase, not to the first or intermediate components. For example, given the terms **FRIEND** and **MICROCOMPUTER**, the context-free rules could be applied to all components (**FRIEND**, **MICRO**, **COMPUTER**), while the suffix rules could be applied only to **COMPUTER**.

Suffix rules do not apply to single-character words or if the last component of an implied phrase is a single character. The final component must contain at least two characters to be eligible for suffix substitution.

These rules described are case-sensitive: to activate a rule, its target field must be matched exactly in upper- and lowercase letters. Each rule with a non-empty target should be repeated once with the target substring in both upper and lowercase. Do not mix upper- and lowercase characters in the same query rule, and warn your end users not to use mixed cases in *GUIDE Reader* query statements.

Replacement substrings may be upper- or lowercase because the search engine normalizes the case of all words before it looks for them in the dictionary. These limits apply to the rules file:

- The maximum number of rules per file is 40
- The maximum size of target and replacement fields is 4
- A maximum of 30 substitutions may be applied simultaneously to a given word

If you exceed any of these limits, the search engine rejects the query. The total number of variants generated from a single query term can become very large when several substitution rules apply. Because the search engine must look up each generated variant form in the dictionary, a large number of variants (more than a few hundred) may cause an unacceptable response from the search engine, even if only a few variants actually occur in the collection.

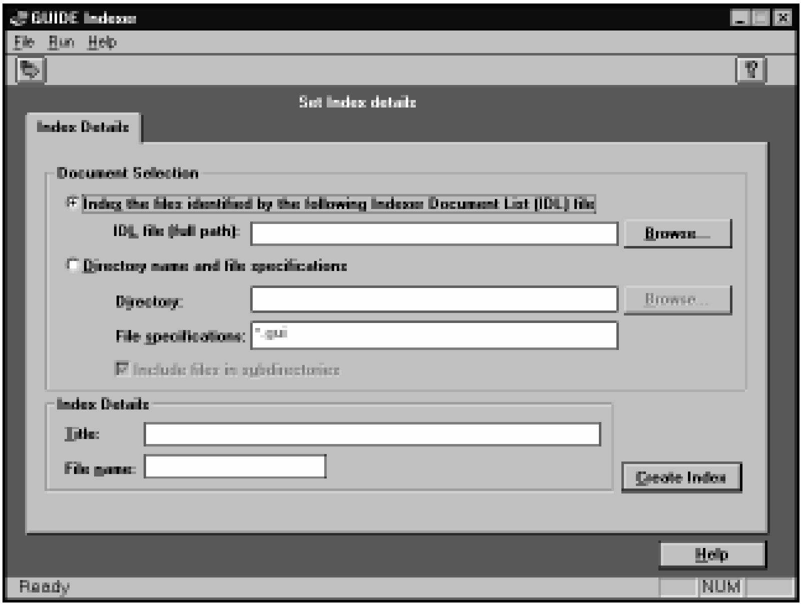
The **gindexer.ftl** term variants file is supplied with *GUIDE Indexer* as a sample document that you can edit or duplicate. It simply appends the suffixes **s** and **'s** to each word. If you want to create additional term variants files, you can open **gindexer.ftl** in any text editor and use **Save As** to save the file under a different name.

Again, the term variants file used during any indexing session will be the one listed in the **[gindexer]** section of the **infacces.ini** file. The default setting is **variants file=gindexer.ftl**. *GUIDE Indexer* will use whatever file is listed in the initialization file for all sessions until the setting is changed.

CHAPTER 4

USING GUIDE INDEXER

Now that you understand what *GUIDE Indexer* does and how it works, you're ready to use it. This chapter describes how to start *GUIDE Indexer,* introduces the application's menus, commands, and dialog options, and explains the indexing process and its results. To start, double-click on the *GUIDE Indexer* program icon.This opens *GUIDE Indexer* to the **Set Index Details** window and the **Index Details** tab dialog (see Figure 4-1). The options in this dialog enable you to select an indexer document list (IDL) file or select the directory you want to index, and name the index and the index file.



Document Selection

● Index the files identified by the following Indexer Document List (IDL) file

IDL file (full path):

○ Directory name and file specifications

Directory:

File specifications: *.gui

☑ Include files in subdirectories

Index Details

Title:

File name:

Figure 4-1
The Set Index Details window

Menus

In addition to the tab dialog on the main screen, *GUIDE Indexer* offers **File**, **Run**, and **Help** menus. The commands on these menus can help you create and manage indexes.

GUIDE Indexer's **File** menu features two commands: **View Index Log** and **Exit**. The **View Index Log** command launches Notepad and opens the log file created during the indexing process. This log provides important information about the indexes you create. You should read it carefully after you index each document collection to make sure that no errors have occurred during the indexing process. For example, the log lists all the documents indexed, noting any that were too large to have been indexed completely. The **Exit** command closes *GUIDE Indexer.* Choosing this command has the same effect as double- clicking on the close box on the title bar of the *GUIDE Indexer* application window.

The **Run** menu offers the single command **Create Index**, which starts the indexing process. This command duplicates the **Create Index** button on the **Index Details** tab dialog. You should make sure the index you're about to generate is configured to your satisfaction in the tab dialog before you select this command.

Use the commands on the **Help** menu to access *GUIDE Indexer's* online help system and to learn about the product. The **Indexer Help** command opens the help system; the **About** command displays the version number and copyright information for *GUIDE Indexer.*

Configuring an Index

To index a document collection, you must first configure the proposed index, select the documents you want to index and give the index a name (and possibly a title). *GUIDE Indexer* provides two options in the **Index Details** tab dialog that you can use to select the documents you want to index: by an indexer document list (IDL) file or from a directory name and file specification.

Index the files identified by the following Indexer Document List (IDL) file:

This option can only be used with an indexer document list (IDL) file. An IDL file is an ASCII text file that lists the documents to be included in an index. To create an IDL file, open a text editor such as Microsoft Notepad and follow this format:

```
<file name 1>
<file name 2>
<file name ...
<file name n>
```

The file name entries may begin with subdirectory names as long as those subdirectories are subdirectories of the directory specified as the top directory. Give the file a **.idl** extension when you save it.

Let's say the following directory structure exists:

c:\techdocs
c:\techdocs\overview.gui
c:\techdocs\toc.gui
c:\techdocs\chap1
c:\techdocs\chap1\doc1.gui
c:\techdocs\chap1\doc2.gui
c:\techdocs\chap2
c:\techdocs\chap2\doc1.gui
c:\techdocs\chap2\doc2.gui

The IDL file in this case should be stored in the **c:\techdocs** directory. An invalid IDL file for this publication would be:

```
c:\techdocs
overview.gui
doc1.gui
doc2.gui
chap2\doc1.gui
chap2\doc2.gui
```

because **doc1.gui** and **doc2.gui** aren't in the **c:\techdocs** directory

A valid file would be:

```
c:\techdocs
overview.gui
chap1\doc1.gui
chap1\doc2.gui
chap2\doc1.gui
chap2\doc2.gui
```

Directory name and file specification

Use this option to index specific documents or documents that are stored in more than one directory. Click on **Browse** to display the **Open** dialog and locate the highest directory that contains the *GUIDE* documents you want to index; **Browse** shows the selected file's full path under **Directories**. If **Include Files in Subdirectories** is checked, *GUIDE Indexer* includes any subdirectories below the main directory in the index.

The **File Specification** field displays ***.gui** by default to indicate that all *GUIDE* files in the specified directory should be indexed; you can, however, enter other file extensions. *GUIDE Indexer* skips files whose names don't end with the extension designated in the **File Specification** field.

If you want to exclude some publication files from the index (such as control panels, table of contents documents, or key word index files), give those file names a different extension than that used for *GUIDE* document file names. For example, if your body document file names use the default extension **.gui**, you might use **.cp** for control panel documents and **.toc** for table of content documents.

You can also include wildcard characters for either the extension or the file name, but not both. Try to create a DOS wildcard specification that matches all the file names. For example, if you want to index all *GUIDE* files (assuming their names include the **.gui** extension) in the **c:\techdocs** directory, type **c:\techdocs** in the **Directory** box, enter ***.gui** in the **File specifications** box, clear the **Include files in subdirectories** checkbox, and then click on **Create Index**.

You can specify multiple wildcards; for example, if you only want to index the first two chapters in a large publication, you could specify as wildcards both **chap1*.gui** and **chap2*.gui** so that *GUIDE Indexer* includes in the full text index only those files that have a **.gui** extension and **chap1** or **chap2** as the first five characters in their file name. Another example is ***.gui;*.gdl**, which is used to specify multiple types of *GUIDE* file extensions. The only file specifications not allowed are ***.*** and ***.???**. If you designate multiple file specifications, separate them with spaces, commas, or semicolons.

If you check **Include files in subdirectories**, *GUIDE Indexer* looks for files that match the wildcards specified in all subdirectories that are below the directory specified in the **Directory** edit box and their subdirectories. Be careful if you build an index on the root of a drive, because *GUIDE Indexer* searches every directory on the drive if this option is checked.

Index Details

You can specify a title for the document collection and a file name for the index file in the **Index Details** tab dialog in the **Set Index Details** window. The title is optional, but you must enter an index file name. *GUIDE Indexer* uses the index name as the prefix for the names of the files it creates during the indexing process.

The text you type in the **Title** text box displays in *GUIDE Reader's* **Query** dialog whenever you conduct full text searches in documents associated with the index file. If no index title is assigned, *GUIDE Reader* refers to the index as **<Untitled Index>**.

Indexing Documents

Once you've configured the proposed index, you're ready to generate the index. Here's a recap of the steps you need to take, using either an IDL file or a directory name and style specifications.

To index documents identified by an IDL file:

1 **In the *Index Details* tab dialog, select the radio button opposite *Index the Files Identified by the Following Indexer Document List (IDL) File*.**

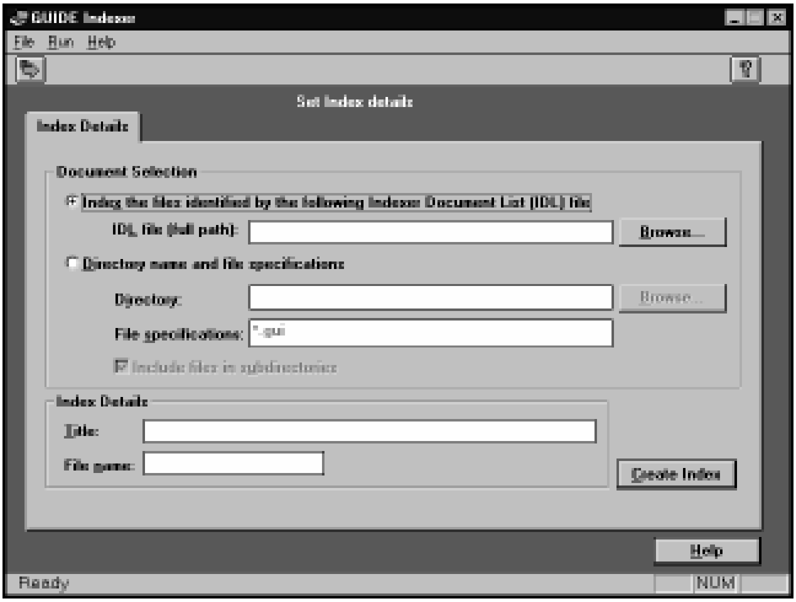
2 **In the text box, enter the IDL file's full path and name, or click on *Browse* to display the *Open* dialog to select the drive and directory where the IDL file is stored.**

3 **Under *Index Details*, specify a title for the document collection and a file name for the index file.**

The title is optional but you must enter an index file name. The IDX file should be a file name only and not a complete path.

4 **Click *Create Index*.**

The indexing process begins, as explained in the next section. To ensure a complete index at all times, you must re-index document collections each time you change one of the documents in an indexed collection.



● Index the files identified by the following Indexer Document List (IDL) file

IDL file (full path):

○ Directory name and file specifications

Directory:

File specifications: *.gui

☑ Include files in subdirectories

Figure 4-2
The Set Index Details window

To index documents with a directory name and file specifications:

1 **In the *Index Details* tab dialog, select the radio button opposite *Directory Name and File Specifications*.**

2 **In the *Directory* text box, enter the path for the highest directory that contains the *GUIDE* documents you want to index. Alternatively, click on *Browse* to display the *Open* dialog and locate the directory.**

DBCS cannot be a part of the directory path. Indexing from the root of a drive is illegal.

3 **In the *File Specifications* text box, enter the extension of the files you want to index.**

The default,***.gui**, indicates that all *GUIDE* files in the specified directory are to be indexed. You can enter other file extensions. Since *GUIDE Indexer* skips files if their names don't end with the extension designated, you can exclude some files from the indexing process by giving those file names a different extension than that used for *GUIDE* files.

Remember, if you check the **Include Files in Subdirectories** option, *GUIDE Indexer* includes all subdirectories below the main directory in the index.

4 **Under *Index Details*, specify a title for the document collection and a file name for the index file. The title is optional but you must enter an index file name. (The file name alone is sufficient; don't enter a complete path.)**

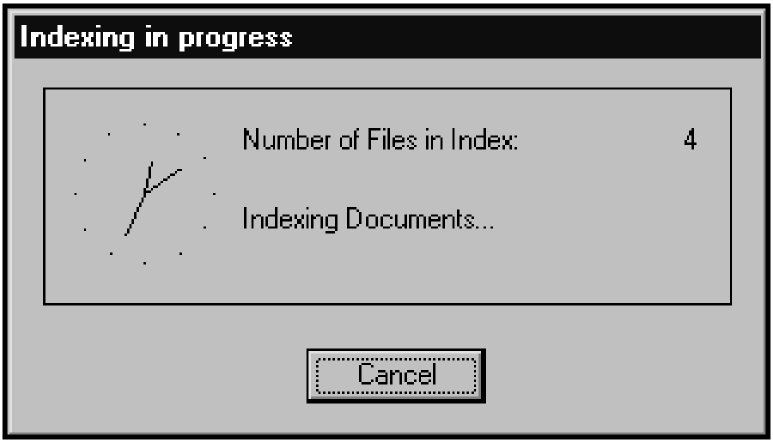
5 **Click on *Create Index*.**

The indexing process begins, as explained in the next section.

The Indexing Process

When you click on **Create Index**, the **Indexing in Progress** dialog provides feedback throughout the batch process; it displays a progress clock, the total number of files to be indexed, the number of files remaining, as well as the task *GUIDE Indexer* is currently working on, such as creating a catalog, adding files to a catalog, or indexing a particular file.

If you click on **Cancel** in the **Indexing in Progress** dialog, the application may not respond immediately because *GUIDE Indexer* interrupts its batch processing only periodically to check for the **Cancel** command. When it does respond, a message informs you that the indexing process was not completed and reminds you to start over if you want to index the publication. Also, if you try to create another index file for the same document collection, a dialog asks if you want to overwrite the existing index file in that directory.


Indexing in progress

Number of Files in Index: 4

Indexing Documents...

Figure 4-3
The Indexing in Progress dialog

Once the indexing batch process is complete, the **Indexing in Progress** dialog closes and a message appears to confirm that the indexing process is finished. This dialog also reminds you to check the index log file to see if any errors occurred during indexing.

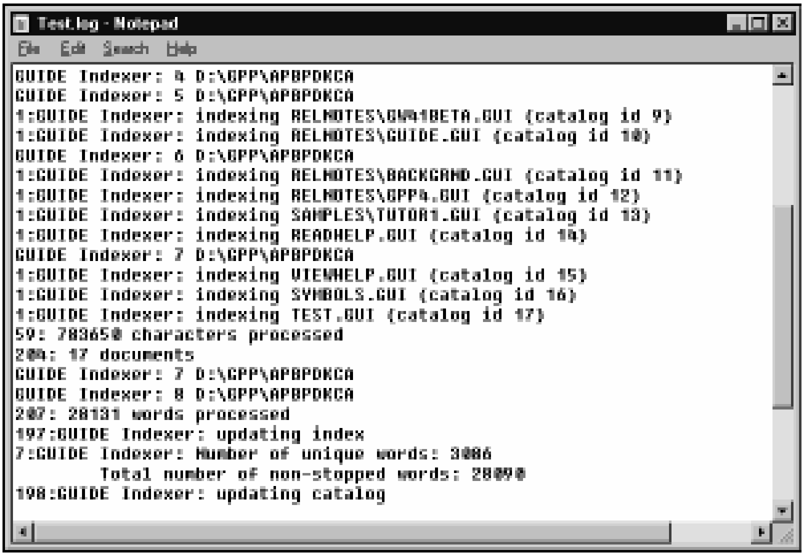
The index log file has the name you assigned for the index file with **.log** as the extension; for example, **techdoc.log**. You can open the log file in any text editor or use the **View Index Log** command on *GUIDE Indexer’s* **File** menu to automatically launch Microsoft Notepad and open the file. If the log file is too large to be opened in Notepad, a message asks you to use another text editor.

NOTE:

If GUIDE Indexer has a problem with a file, try to open that file in GUIDE Author or GUIDE Reader to verify whether or not it is a GUIDE document. To work around the problem, you can recreate the document or restore it from a backup, then run GUIDE Indexer again.

It’s important to check the index log carefully. If *GUIDE Indexer* didn’t recognize a file it was supposed to index, a message in the log tells you that *GUIDE Indexer* couldn’t open that file. This usually indicates that the file is corrupted or not a *GUIDE* file.

You must re-index a publication each time any of the documents in that collection change. To re-index a document collection, simply follow the same steps used in the original indexing.



```
GUIDE Indexer: 4 D:\GPP\APBPDKCA
GUIDE Indexer: 5 D:\GPP\APBPDKCA
1:GUIDE Indexer: indexing RELNOTES\GW41BETA.GUI (catalog id 9)
1:GUIDE Indexer: indexing RELNOTES\GUIDE.GUI (catalog id 10)
GUIDE Indexer: 6 D:\GPP\APBPDKCA
1:GUIDE Indexer: indexing RELNOTES\BACKGRND.GUI (catalog id 11)
1:GUIDE Indexer: indexing RELNOTES\GPP4.GUI (catalog id 12)
1:GUIDE Indexer: indexing SAMPLES\TUTOR1.GUI (catalog id 13)
1:GUIDE Indexer: indexing READHELP.GUI (catalog id 14)
GUIDE Indexer: 7 D:\GPP\APBPDKCA
1:GUIDE Indexer: indexing VIEWHELP.GUI (catalog id 15)
1:GUIDE Indexer: indexing SYMBOLS.GUI (catalog id 16)
1:GUIDE Indexer: indexing TEST.GUI (catalog id 17)
59: 783650 characters processed
204: 17 documents
GUIDE Indexer: 7 D:\GPP\APBPDKCA
GUIDE Indexer: 8 D:\GPP\APBPDKCA
207: 28131 words processed
197:GUIDE Indexer: updating index
7:GUIDE Indexer: Number of unique words: 3086
        Total number of non-stopped words: 28090
198:GUIDE Indexer: updating catalog
```

Figure 4-4
GUIDE Indexer’s log file

Command Line Indexing

GUIDE Indexer supports command line processing. If you have a publishing process that calls *GUIDE Writer* from a command line, you can now complete the process of creating your *GUIDE* publication by having your collection indexed from the same process.

The syntax is

```
   gindexer.exe <parameter> <value>
```

Note: To create a new index with the name of an index that already exists, you must first delete the old index; otherwise, *GUIDE Indexer* will not be able to create the new index.

The best way to delete an old index is to use the **del myindex** command in the batch file immediately before the command line to create the new index. For example, to delete the **index**, you would use the command line instruction **del index.*** so that all 12 index files associated with **index** are deleted.

Here is a list of the parameters and values that can be used:

PARAMETER AND VALUE	*EXPLANATION*
-IDX <IDX file name>	Specifies the name of the IDX file.
-TITLE <index name>	Specifies the name of the index title.
-DIR <base directory>	Specifies the base directory of the index.
-WILD <wildcard spec>	Specifies wildcards of GUIs to index.
-IDL <IDL file name>	Specifies the IDL file to use. This option takes precedence over the DIR, SUB, and WILD options.
-SUB	Toggles the use of subdirectories. (The default is don't include subdirectories.)
-RUN	Toggles the Auto Run feature. (The default is do not auto run.)
-SILENT	Suppresses dialog box error messages.
-EXIT	Exits the program.

Option names are not case-sensitive.

You can allow values with embedded spaces by using quotation marks to delimit the value of any option. The quotation marks do not form part of the value.

Whether working from a command line or the interface, valid characters for the path or file name are **A** through **Z**, **a** through **z**, dot (**.**), colon (**:**), and backslash(\).

C:\NUWC-Key\Vol_1\Allvols.idx would fail to index and a descriptive error message would display because of the hyphen in the directory name. This is a limtation of the search engine, not a *GUIDE Author* restriction.

Examples

The first example points to *GUIDE Indexer* in the **c:\guide** directory, creates a **master.idx** file in the docs directory with the title **Master Publication Index**, and then runs the index.

```
   c:\guide\gindexer.exe -idx master.idx -title
"Master Publication Index" -run
```

The second example calls *GUIDE Indexer* (**gindexer.exe**) from the **h:** drive, creates the index **resources.idx** with the title **Human Resources Index**, includes subdirectories, and uses an IDL list from the **m:** drive. Finally, it autoruns the process.

```
   h:\guide\gindexer.exe -idx resources.idx -title
"Human Resources Index" -sub -idl
m:\authoring\resources.idl -run
```

The third example calls *GUIDE Indexer* (**gindexer.exe**) from the **d:** drive, creates the index **test2.idx** with the title **Test 2**, specifies the base directory for the index as **d:\guidetest\CommandLineIndex**, selects all GUI files, and then autoruns the process.

```
   d:\guide\gindexer -idx test2 -title "Test 2" -dir
d:\guidetest\CommandLineIndex -wild *.gui -run
```

The fourth example shows how to exit the program.

```
   d:\guide\gindexer.exe -idx test1.idx - title
test1.idx -title "Test1" -dir
d:\guidetest\CommandLineIndex -run -silent -exit
```

Migration Issues for GUIDE Indexer

Use Convert4_to_5.gui in the C:\GUIDE\Samples directory to automate the saving of version 4.1 files to the 5.0 format. Open the file in *GUIDE Author* and click on the Details expansion button for specific instructions on how to set up the conversion. Each directory requires a text file listing the files to be converted. The path is placed inside the group. Then simply click the command button to convert the directory of GUI files.

About Your Files

GUIDE Indexer's batch process creates 12 files and places them all in the publication's highest directory. The names of these files consist of a prefix from the file name you assigned to the index in the Index Details section of the **Index Details** tab dialog plus an assigned extension. For example, if you designate **policies.idx** as the index name in the **Index Details** tab dialog, *GUIDE Indexer* creates the following files: **policies.stp**, **policies.fth**, **policies.ftl**, **policies.cat**, **policies.dct**, **policies.ref**, **policies.cfg**, **policies.idx**, **policies.cix**, **policies.zon**, **policies.wwl**, and **policies.log**. You must distribute all these files except the **.log** file with indexed document collections to enable readers to conduct full text searches in *GUIDE Reader.*

Remember, *GUIDE Indexer* automatically includes every significant word in the indexed documents in the full text index, ignoring only inconsequential words such as articles, conjunctions, and prepositions, as specified in a 'stop list'. Words excluded from full text indexes by default are:

- **after**
- **also**
- **an**
- **and**
- **as**
- **at**
- **be**
- **because**
- **before**
- **between**
- **but**
- **by**
- **for**
- **from**
- **however**
- **if**
- **in**
- **into**
- **of**
- **or**
- **other**
- **out**
- **since**
- **such**
- **than**
- **that**
- **the**
- **there**
- **these**
- **this**
- **those**
- **to**
- **under**
- **upon**
- **when**
- **where**
- **whether**
- **which**
- **with**
- **within**
- **without**

The full text indexes that *GUIDE Indexer* generates also take plurals and possessives into account so that a reader's full text queries in *GUIDE Reader* find those occurrences of search items as well as hits that appear exactly the way the reader types the text into a query. For example, if the reader searches for the word "query", the search results include not only every occurrence of 'query', but all instances of the word in its possessive and plural forms ("query's" and "queries"). Queries also recognize numerical values as their word equivalents, for example, 1 for one and 2 for two.

NOTE:

Because GUIDE Indexer ignores text strings that contain less than two characters, the full text index stop list does not include 'a'.

INDEX