



connecting people with information

---

# Verity Information Server

## Student Workbook



Education Department  
892 Ross Drive  
Sunnyvale, CA 94089  
(408) 541-1500

*all rights reserved*

## INDEXING WITH THE VERITY SPIDER

The Information Server's indexing spider can create collections from a variety of document types, including:

- ASCII text documents. This includes text documents and documents in Internet e-mail or unused news format.
- HTML documents
- Adobe Acrobat documents. This includes documents in Acrobat PDF format.
- WSIWYG documents. This includes documents supported by the MasterSoft filter and viewer kit, including Microsoft word, WordPerfect and other formats.

The spider can automatically index all of these various document types into a single master collection. While internally each document type is indexed into an appropriate collection for its type, the end result appears to be a single collection. This greatly simplifies collection management and making collections available to users.

### **WHAT IS A COLLECTION?**

Collections represent groups of documents that have been processed so they are retrievable by Verity Products. During processing, optimized indexes are automatically created to enable fast searching of words contained in the documents or attribute information about the documents like their titles or authors. A collection is a series of indexes, which store data about your documents.

- Some HTML tags are translated automatically into zones for more specific searching
- Zones allow full query language searching (including topics)
- Fields can be also be created for more targeted searching or for use in sorting or displaying results

All the words in the document are captured and include information to allow concept, Boolean and proximity searching.

### **THE COLLECTION DIRECTORY**

- Each time a new collection is created, a home directory is created
- The collection home directory includes sub-directories that store and maintain information about the collection, style files and documents

The collection directory contains all the files that are needed to provide access to your documents or to manage the collection. The main job of the collection is to store information about the document (attributes) and to store the word index. Because you index at different times, or index a large amount of data, the collection is optimized into smaller components called partitions, each having these two indexes (attributes and words) for a group of documents in the collection. A collection can have many partitions. Depending upon how the collection was created, there are other directories in the collection that store data. Here is an overview of the

basic role of these various collection directories. Remember that all collections do not have all of the same sub-directories, they only have the ones they need.

The **ASSISTS Directory** stores spanning word lists and information about the collection. The **PARTS Directory** stores partitions, consisting of two key elements, the “.ddd” which holds the attribute information and the “.did” which holds the word index.

The .ddd file is a binary list of all of the documents in a partition. There is a separate record for each document, which includes several internal fields (like the document name) along with user-defined fields. The .did file is a binary database of all the words in all of the documents in a partition. If the documents are large and the partition is large, the .did files will be quite large as well. The general rule of thumb is that the .did will probably be about 40% of the size of the original documents.

The **PDD Directory** stores the partition index, which manages all the partitions in the collection. The **STYLE Directory** stores configuration files. The **TRANS Directory** stores transaction files used in auto-administration.

Other collection directories you may find, based on the collection utility’s needs or features, are the **TOPICIDX Directory** which stores topic indexes for each partition, **WORK Directory** which stores task files used in auto-administration, **MORGUE Directory** which stores files that will be deleted.

### THE ROLE OF STYLE FILES

- Style files configure the series of indexes storing data about documents and your environment
- The choices made in style files direct how indexing utilities will create indexes and the guidelines for doing work
- The Information Server ships with a special style directory which allows you to incorporate many types of documents in your collection, without having to do any additional work

### EXPLORING STYLE GUIDELINES

You will probably not need to alter your style files. Verity tries to anticipate the kinds of information you will wish to extract from documents and do the prep work for you. However, it is possible that your documents will have special information, which could not be anticipated. When this is the case, you will need to edit style files to add the information you want to extract. This class does not attempt to cover everything related to style files for all utility programs, and there is additional training available. If you need to make changes to style files, we suggest that you review the user guide called Collection Building. It contains the details you need to modify style files or to choose which style files you should incorporate based on your specific needs.

Just to give you an idea about how the style files are defined, here are a few that are used in the main style directory provided by Verity.

## THE STYLE.DDD FILE CONFIGURES THE COLLECTION

```
# $Id: style.ddd,v 1.1.1.5 1997/02/23 04:19:02 wade Exp $
# Copyright (C) 1987-1997 Verity, Inc.
# Document Dataset Descriptor
$control: 1
#include style.prm
$subst: 1
descriptor:
  /collection = yes
{
  data-table:    _df
    /num-records = 1
    /max-records = 1
  {
    # Header information for partition management
    worm:    _DBVERSION text
    fixwidth:    _DDDSTAMP    4 date
    varwidth:    _DOCIDX      _dv
    fixwidth:    _PARTDESC    32 text
    constant:    _FtrCfg      text "${DOC-FEATURES:}"
    constant:    _SumCfg      text "${DOC-SUMMARIES:}"

    fixwidth:    _SPARE1      16 text
    fixwidth:    _SPARE2      4 signed-integer
  }
  # Required internal fields per document
  data-table:    _df
    /offset=64
  {
    autoval:    _STYLE          sirepath
    fixwidth:    _DOCID          4 unsigned-integer
    fixwidth:    _SECURITY      4 unsigned-integer
    /minmax = yes
    fixwidth:    _INDEX_DATE    4 date
    /minmax = yes
  }
}
#ifdef DOC-FEATURES
# Optional feature vector per document
data-table:    _dg
{
  varwidth:    VDKFEATURES     _dh
  /_implied_size
}
#endif
#ifdef DOC-SUMMARIES
# Optional generated summary per document
data-table:    _di
{
  varwidth:    VDKSUMMARY      _dj
}
```

```
        /_implied_size
    }
$endif
$include style.ufl
}
$$
```

## THE STYLE.DFT FILE DEFINES THE VIRTUAL DOCUMENT

### HTML Version

```
# $Id: style.dft,v 1.1.1.5 1997/04/17 00:04:43 ameyer Exp $
# Copyright (C) 1987-1995 Verity, Inc.
# Document Format
#
$control:1
dft:
{
    field: DOC
        # The following line invokes the HTML zone filter
        /filter="zone -html"
}
$$
```

### Auto-Filt Version

```
# $Id: style.dft,v 1.1.1.1 1996/09/12 22:46:56 ameyer Exp $
# Copyright (C) 1987-1995 Verity, Inc.
# Document Format
#
$control:1
dft:
{
    field: DOC
        # The following line invokes MasterSoft WYSIWYG filtering
        /filter = "auto"
}
$$
```

### PDF Version

```
# $Id: style.dft,v 1.1.1.3 1997/02/28 19:14:57 wade Exp $
# Copyright (C) 1987-1996 Verity, Inc.
#
# Document Format
control:1
dft:
{
    field: DOC
        /filter="flt_pdf -charmpto 850"
}
$$
```

## THE STYLE.PRM FILE GOVERNS WORD INDEXING FEATURES

```
# $ID: STYLE.PRM,v 1.1 1997/02/23 04:26:15 WADE EXP $
# COPYRIGHT (C) 1987-1995 VERITY, INC.
#
# STYLE.PRM - COLLECTION SCHEMA PARAMETERS
#
# THIS FILE IS USED TO ENABLE/DISABLE INDEX SCHEMA FEATURES THROUGH
# MACRO DEFINITIONS SIMILAR TO THOSE ALLOWED BY THE C PREPROCESSER.
# THIS FILE IS INCLUDED IN OTHER STYLE FILES USING $INCLUDE SO
# THAT THE SELECTED FEATURES ARE PROPOGATED TO THE SCHEMAS OF ALL
# TABLES IN THE INDEX. REFER TO THE "USING THE STYLE.PRM FILE"
# CHAPTER IN THE COLLECTION BUIDING GUIDE FOR MORE INFORMATION.

# -----
# THE IDX-CONFIG PARAMETER DEFINES THE STORAGE FORMAT USED TO
# ENCODE THE WORD POSITIONS IN THE INDEX. WCT (WORD COUNT) FORMAT
# IS A COMPACT FORMAT, STORING THE ORDINAL COUNTING POSITION OF THE
# WORD FROM THE BEGINNING OF THE DOCUMENT. PSW (PARAGRAPH, SENTENCE,
# WORD) FORMAT TAKES APPROXIMATELY 15-20% MORE DISK SPACE, BUT
# STORES SEMANTICALLY ACCURATE PARAGRAPH AND SENTENCE BOUNDARIES.
# OPTIONALLY, MANY MAY BE SPECIFIED WITH EITHER WCT OR PSW TO
# IMPROVE THE ACCURACY OF THE <MANY> OPERATOR AT THE EXPENSE OF
# DISKSPACE AND SEARCH PERFORMANCE.

# THIS EXAMPLE ENBLES WORD COUNT WORD POSITION FORMAT (THE DEFAULT).
$DEFINE IDX-CONFIG "WCT"

# THIS EXAMPLE TURNS ON PARAGRAPH/SENTENCE/WORD WORD POSITION FORMAT.
# IT ALSO ENABLES THE <MANY> OPERATOR ACCURACY IMPROVEMENT.
#$DEFINE IDX-CONFIG "PSW MANY"

# -----
# THE IDXOPTS PARAMETERS DEFINE WHICH INDEX OPTIONS ARE APPLIED TO
# THE VARIOUS INDEX TOKEN TABLES. THE FOLLOWING INDEX OPTIONS ARE
# SUPPORTED FOR EACH: STEMDEX ENABLES AN INDEX BY THE STEM OF EACH
# WORD. CASEDEX STORES ALL CASE VARIANTS OF A WORD SEPARATELY, SO
# ONE CAN SEARCH FOR CASE SENSITIVE TERMS SUCH AS "JOBS", "APPLE",
# AND "NEXT" MORE EASILY. SOUNDEX STORES PHONETIC REPRESENTATIONS
# OF THE WORD, USING AT&T'S STANDARD SOUNDEX ALGORITHM. THE
# APPLICATION MAY ALSO STORE 1-4 BYTES OF APPLICATION-SPECIFIC
# DATA WITH EACH WORD INSTANCE, IN THE FORM OF LOCATION DATA AND/OR
# QUALIFY INSTANCE DATA. THESE OPTIONS ARE SPECIFIED SEPARATELY
# FOR EACH TOKEN TABLE: WORD, ZONE, AND ZONE ATTRIBUTE.
$DEFINWORD-IDXOPTS "STEMDEX CASEDEX"
$DEFINZONE-IDXOPTS ""
$DEFINATTR-IDXOPTS "CASEDEX"

# -----
# CLUSTERING IS ENABLED BY UNCOMMENTING THE DOC-FEATURES LINE.
# THIS STORES A FEATURE VECTOR FOR EACH DOCUMENT IN THE
# DOCUMENTS TABLE. THESE FEATURES ARE USED FOR CLUSTERING
# RESULTS AND FAST QUERY-BY-EXAMPLE. SEE THE DISCUSSIONS ON
# CLUSTERING IN THE COLLECTION BUILDING GUIDE FOR MORE INFORMATION.
```

```
$DEFINE DOC-FEATURES "TF"
```

```
# -----  
# DOCUMENT SUMMARIZATION IS ENABLED BY UNCOMMENTING ONE OF  
# THE DOC-SUMMARIES LINES BELOW. THE SUMMARIZATION DATA IS  
# STORED IN THE DOCUMENTS TABLE SO THAT IT MIGHT EASILY BE  
# SHOWN WHEN DISPLAYING THE RESULTS OF A SEARCH.  
# SEE THE DISCUSSIONS ON DOCUMENT SUMMARIZATION IN THE  
# COLLECTION BUILDING GUIDE FOR MORE INFORMATION.  
# THE EXAMPLE BELOW STORES THE BEST THREE SENTENCES OF  
# THE DOCUMENT, BUT NOT MORE THAN 500 BYTES.  
$DEFINE DOC-SUMMARIES "XS MaxSENTs 3 MaxBYTES 500"  
# THE EXAMPLE BELOW STORES THE FIRST FOUR SENTENCES OF  
# THE DOCUMENT, BUT NOT MORE THAN 500 BYTES.  
#$DEFINE DOC-SUMMARIES "LS MaxSENTs 4 MaxBYTES 500"  
# THE EXAMPLE BELOW STORES THE FIRST 150 BYTES OF  
# THE DOCUMENT, WITH WHITESPACE COMPRESSED.  
#$DEFINE DOC-SUMMARIES "LB MaxBYTES 150"
```

## THE STYLE.WLD GOVERNS HOW WORDS ARE STORED

```
# $Id: STYLE.WLD,v 1.1.1.3 1997/04/02 23:26:47 WADE EXP $  
# COPYRIGHT (C) 1987-1997 VERITY, INC.  
# STYLE.WLD - WORD LIST DESCRIPTOR  
$CONTROL: 1  
$INCLUDE STYLE.PRM  
$SUBST: 1  
DESCRIPTOR:  
{  
  DATA-TABLE:  _PH  
    /NUM-RECORDS=1  
    /MAX-RECORDS=1  
  {  
    # HEADER INFORMATION FOR PARTITION MANAGEMENT  
    WORM:      _DBVERSION      TEXT  
    FIXWIDTH:  _DDDSTAMP       4 DATE  
    FIXWIDTH:  _DIDSTAMP       4 DATE  
    CONSTANT:  TYPES           TEXT "WORD ZONE ATTR"  
    CONSTANT:  CONFIG          TEXT "$IDX-CONFIG"  
    CONSTANT:  WORD            TEXT "$WORD-IDXOPTS"  
    CONSTANT:  ZONE            TEXT "$ZONE-IDXOPTS"  
    CONSTANT:  ATTR            TEXT "$ATTR-IDXOPTS"  
    CONSTANT:  TSPARE1         TEXT ""  
  }  
# THIS IS THE TABLE OF PARTS COVERED BY SPANNING WORDLIST  
DATA-TABLE:  _PP  
{  
  FIXWIDTH:  PARTNUM          4 UNSIGNED-INTEGER  
}  
# THE ACTUAL FULL TEXT INDEX DATA  
DATA-TABLE:  _PF  
{  
  VARWIDTH:  FWTEXT           _PW  
    /_IMPLIED_SIZE  
  VARWIDTH:  FWDATA           _PV  
    /_IMPLIED_SIZE  
  FIXWIDTH:  FWENCODE          1 UNSIGNED-INTEGER  
  FIXWIDTH:  FWFREQ            2 UNSIGNED-INTEGER
```

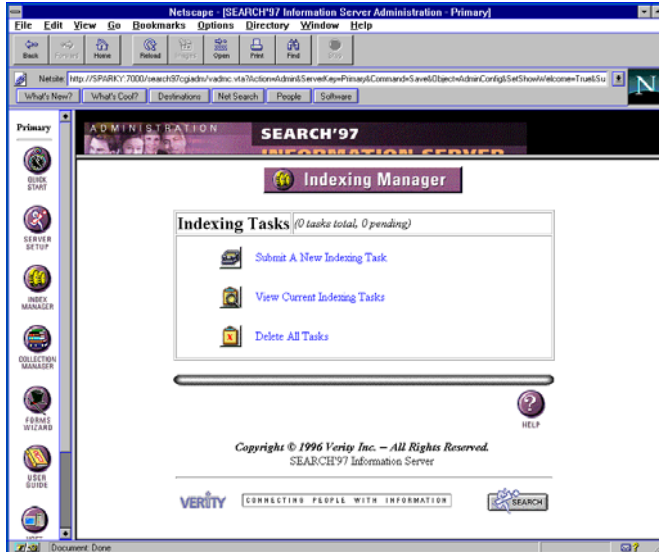




## USING THE GRAPHICAL COLLECTION TOOLS

There are two key areas in the Information Server that are oriented to creating and managing collections.

The **Indexing Manager** helps you to index your site or other sites of interest



Collections can be created

- ◆ Simple mode requires a name, description and path
- ◆ Advanced mode allows you to filter for specific document MIME types, exclude or include based on specific filename pattern matching, and set proxy information at the collection level

Indexing tasks are defined and can be submitted for initial creation and then maintained as a current task for resubmission

The **Collection Manager** allows you to

- ◆ Enable or disable collections for searching
- ◆ Import externally created collections
- ◆ View information about your collections

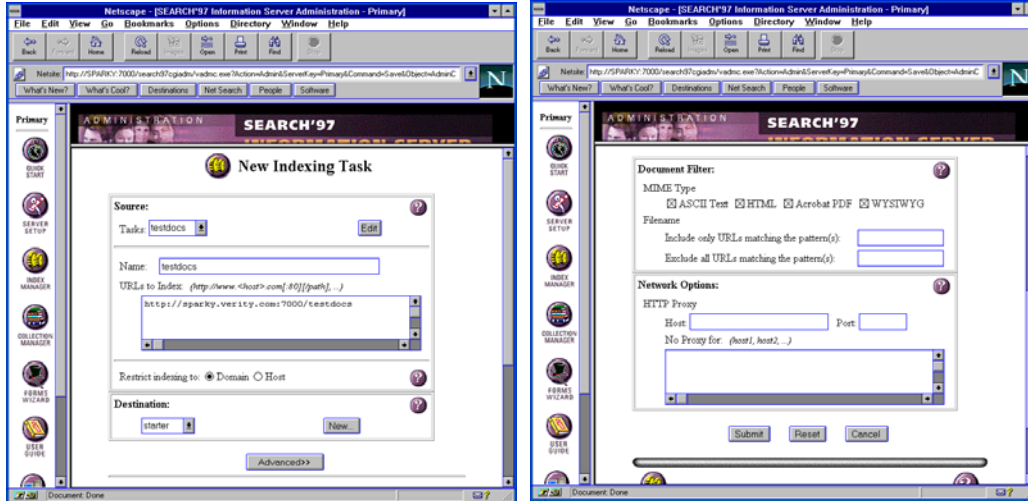


## SUBMITTING INDEXING TASKS

- Before you can submit URLs for indexing, you must have a collection ready to receive the information
- Once you have a collection ready to go, you submit an indexing task which includes a name for the task, a list of URLs to index and the destination collection
- As you build your list of tasks, you will be able to select them from a drop-down list
- Be sure you index into the right collection because mistakes can be costly!

You can use the Indexing Manager to submit a new indexing task. To submit a new indexing task do the following:

1. Click the Indexing Manager button on the menu bar.
2. Click the New Indexing Task icon or hyperlinks on the Indexing Manager



## USING THE COLLECTION LIST

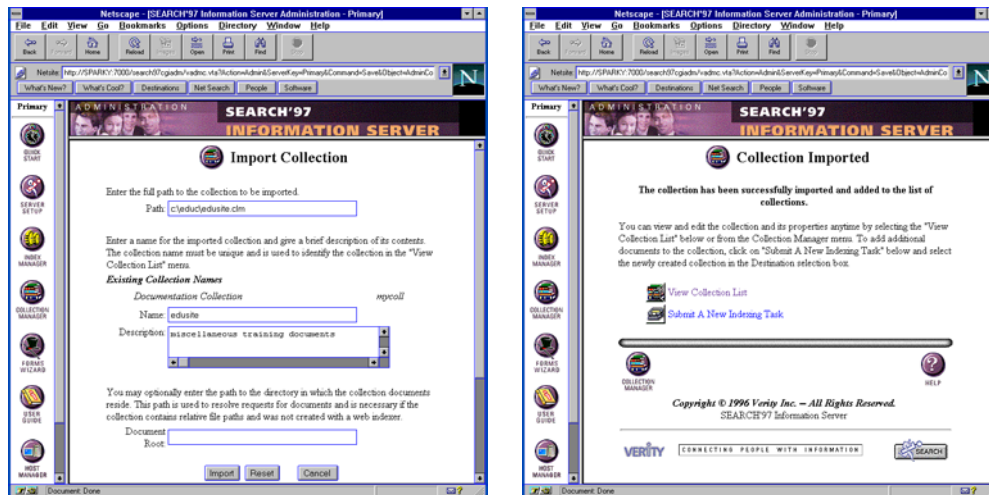
As you build a number of collections, you will work with them on the Collection List. It is easy to enable and disable searching. The “Edit” icon takes you to Collection Properties.



## IMPORTING COLLECTIONS

If you have been using Verity products, you will have other collections that you wish to enable through The Information Server.

- Collections that have been built with a file indexer will need to have the Document Root field completed
- Collections built with previous versions of Verity's web indexers should import without a problem



## REVIEWING COLLECTION PROPERTIES

- Properties provide information about the current status of the collection.



## MORE ABOUT COLLECTIONS

The SEARCH'97 Information Server ships with additional collection and topic building and reporting tools

- mkvdk for building collections stored on the file system
- mktopics for building topics to be used in the Information Server
- browse to see the field and zone contents of your collection
- didump to see the words in your index

It is a good idea to become familiar with the features of these utilities, which are covered in detail in the Collection Building reference manual. This workbook contains a bit of information and exercises on each.

## VIEWING DOCUMENT INFORMATION IN COLLECTIONS

There are two utilities you can use to better understand the information contained in your collection.

- Browse allows you to view values for each of the attributes captured in the .ddd file
- Didump allows you to view words in the word index (the .did) file

Here is information on using the Browse utility (found in the bin directory). You simply go to the partition directory (parts) under your collection, and choose which partition you wish to view. As appropriate for your platform enter this command:

```
browse 00000001.ddd
```

A short menu will appear. At this point you can press Enter and the first record will appear. If you have scroll bars, you can scroll back to view the field values. If not, just anticipate the commands and redirect output to a log file.

To do this, enter: `browse 00000001.ddd > browse.out`

Press Enter, the first enter displays the menu (you won't see anything on the screen) and enter a second time to capture values for the first record, Enter again if you want to see the next record) and then type quit. You will be back at the prompt and can now view the log file (browse.out) to see the values.

```
? ) help
q ) quit
c ) Number of entries in field
_ ) Toggle viewing fields beginning with '_'
v ) Toggle viewing selected fields
##) Display all fields in specified record number
Dispatch/Compound field options:
n ) No dispatch
d ) Dispatch
s ) Dispatch as stream
Action (? for help):
```

Your log file should look like this.

browse.exe - Verity, Inc. Version 2.2.0 (\_nti31, May 2 1997)

BROWSE OPTIONS

? ) help  
q ) quit  
c ) Number of entries in field  
\_ ) Toggle viewing fields beginning with '\_'  
v ) Toggle viewing selected fields  
## ) Display all fields in specified record number  
Dispatch/Compound field options:  
n ) No dispatch  
d ) Dispatch  
s ) Dispatch as stream

Action (? for help): Record number: 0

0	_DDFLAG	FIX-unsq ( 1) = 0x00
1	_DDVALUE	VAR-text ( 0) =
2	_DDVALUE_OF	FIX-unsq ( 4) = 0
3	_DDVALUE_SZ	FIX-unsq ( 2) = 0
4	_DBVERSION	WRM-text ( 6) = vdk21
5	_DDDSTAMP	FIX-date ( 4) = 06-May-1997 06:15:55 pm
6	_DOCIDX	VAR-text ( 12) = _
7	_PARTDESC	FIX-text ( 32) = mkvdk (Verity, Inc. Version 2.2.
8	_FtrCfg	CON-text ( 3) = TF
9	_SumCfg	CON-text ( 27) = XS MaxSents 3 MaxBytes 500
10	_SPARE1	FIX-text ( 16) =
11	_SPARE2	FIX-sign ( 4) = 0
12	_DOCIDX_OF	FIX-unsq ( 4) = 32
13	_DOCIDX_SZ	FIX-unsq ( 2) = 12
14	_STYLE	AUT-text ( 19) = ../style/style.ddd
15	_DOCID	FIX-unsq ( 4) = 1
16	_SECURITY	FIX-unsq ( 4) = 0
17	_INDEX_DATE	FIX-date ( 4) = 06-May-1997 06:15:56 pm
18	_SECURITY_MI	WRM-unsq ( 4) = 0
19	_SECURITY_MX	WRM-unsq ( 4) = 0
20	_INDEX_DATE_MI	WRM-date ( 4) = 06-May-1997 06:15:56 pm
21	_INDEX_DATE_MX	WRM-date ( 4) = 06-May-1997 06:16:23 pm
22	VDKFEATURES	VAR-text (229) = _
23	VDKFEATURES_OF	FIX-unsq ( 4) = 32
24	VDKSUMMARY	VAR-text (240) = This chapter covers basic information about SEARCH'97(TM) Information Server and the SEARCH'97 search technology from Verity© that is integrated in the product. The following subjects are included: Introduction to Verity Search Technology
25	VDKSUMMARY_OF	FIX-unsq ( 4) = 32
26	VdkVgwKey	VAR-text ( 37) = ../locale/english/doc/user/01_is.htm
27	VdkVgwKey_IX	FIX-unsq ( 3) = 123
28	VdkVgwKey_MI	WRM-text ( 41) = ../locale/english/doc/collbldg/01_bc.htm
29	VdkVgwKey_MX	WRM-text ( 40) = ../locale/english/doc/user/preface2.htm
30	VdkVgwKey_OF	FIX-unsq ( 4) = 32

```

31 VdkVgwKey_SZ      FIX-unsq ( 2) = 37
32 DOC              DSP-text ( -1) =
..\\..\\..\\locale\\english\\doc\\user\\01_is.htm
33 DOC_FN          VAR-text ( 37) = ../locale/english/doc/user/01_is.htm
34 _CACHE_FN       VAR-text ( 0) =
35 _CACHE_DELETE   FIX-unsq ( 1) = 0
36 _ParentID       VAR-text ( 27) = ../locale/english/doc/user
37 Title           VAR-text ( 13) = Introduction
38 Ext             VAR-text ( 4) = htm
39 Author          VAR-text ( 0) =
40 Subject         VAR-text ( 0) =
41 Keywords        VAR-text ( 0) =
42 Comments        VAR-text ( 0) =
43 Snippet         VAR-text (399) = Introduction . . . . 1 . Introduction .
This chapter covers basic information about SEARCH'97(TM) Information Server
and the SEARCH'97 search technology from Verity&#174; that is integrated in
the product. The following subjects are included: . Introduction to
Information Server . New SEARCH'97 Features . Introduction to Verity Search
Technology . Copyright &#169; 1997, Verity, Inc. All rig...
44 URL            VAR-text ( 0) =
45 MIME-Type       VAR-text ( 10) = text/html
46 Language        VAR-text ( 0) =
47 Encoding        VAR-text ( 0) =
48 _Created        VAR-text ( 26) = Wed May 7 01:04:19 1997
49 _Modified       VAR-text ( 26) = Wed Apr 30 00:09:02 1997
50 Created         FIX-date ( 4) = 07-May-1997 01:04:19 am
51 Modified        FIX-date ( 4) = 30-Apr-1997 12:09:02 am
52 Size           FIX-unsq ( 4) = 1068
53 DOC_OF         FIX-unsq ( 4) = 0
54 DOC_SZ         FIX-unsq ( 4) = 4294967295
55 DOC_FN_OF      FIX-unsq ( 4) = 32
56 DOC_FN_SZ      FIX-unsq ( 2) = 37
57 _CACHE_FN_OF   FIX-unsq ( 4) = 0
58 _CACHE_FN_SZ   FIX-unsq ( 2) = 0
59 _ParentID_OF   FIX-unsq ( 4) = 104
60 _ParentID_SZ   FIX-unsq ( 2) = 27
61 Title_OF       FIX-unsq ( 4) = 530
62 Title_SZ       FIX-unsq ( 2) = 13
63 Ext_OF         FIX-unsq ( 4) = 100
64 Ext_SZ         FIX-unsq ( 2) = 4
65 Author_OF      FIX-unsq ( 4) = 0
66 Author_SZ      FIX-unsq ( 2) = 0
67 Subject_OF     FIX-unsq ( 4) = 0
68 Subject_SZ     FIX-unsq ( 2) = 0
69 Keywords_OF    FIX-unsq ( 4) = 0
70 Keywords_SZ    FIX-unsq ( 2) = 0
71 Comments_OF    FIX-unsq ( 4) = 0
72 Comments_SZ    FIX-unsq ( 2) = 0
73 Snippet_OF     FIX-unsq ( 4) = 131
74 Snippet_SZ     FIX-unsq ( 2) = 399
75 URL_OF         FIX-unsq ( 4) = 0
76 URL_SZ         FIX-unsq ( 2) = 0
77 MIME-Type_OF   FIX-unsq ( 4) = 32
78 MIME-Type_SZ   FIX-unsq ( 2) = 10

```

```

79 Language_OF      FIX-unsg ( 4) = 0
80 Language_SZ      FIX-unsg ( 2) = 0
81 Encoding_OF      FIX-unsg ( 4) = 0
82 Encoding_SZ      FIX-unsg ( 2) = 0
83 _Created_OF      FIX-unsg ( 4) = 68
84 _Created_SZ      FIX-unsg ( 2) = 26
85 _Modified_OF     FIX-unsg ( 4) = 42
86 _Modified_SZ     FIX-unsg ( 2) = 26

```

## Viewing Words in the Index

You can use the `didump` utility to see words in a collection. The `didump` utility includes information about the occurrences of words in a single partition. Output can be viewed on the screen or redirected to a file. If words have been auto-stopped, you may find a value of 0 in the first four columns of the index. If you are concerned that a particular word might occur frequently and wish to check the status, you can use the `didump` utility and specify the word.

For instance, from the `coll` a collection, you could use the `didump` to see the occurrences of the word “the” in the word indexes which are stored under the `parts` directory.

```
didump -pattern the 00000000.did
```

Word	Size	Doc	Word
THE	8	1	1
The	98	5	27
the	605	6	195

## BUILDING COLLECTIONS WITH THE COMMAND-LINE SPIDER: VSPIDER

The optional command-line spider (**vspider**) provides additional indexing options and greater control than is available in the Information Server Indexing Manager. Following are listed some of the more common commands for building a collection or maintaining a collection using `vspider`. Remember that `vspider` is an optional product. It is very helpful for routine collection building, for instance at night, when you can set it to run in `chron`, without having to issue indexing commands directly.

All commands need to be entered on one line:

### Creating a New Collection

```
vspider -collection mycoll.clm -style verity/s97is/locale/english/styles
-start http://www.yoursite.com
```

### Re-indexing a Collection

```
vspider -collection mycoll.clm
```

### Using domain and exclude options

```
vspider -collection mycoll.clm -style verity/s97is/locale/english/styles
-start http://www.yoursite.com -exclude *secrets* -domain verity.com
```

### Using the proxy option:

```
vspider -collection mycoll_2.clm -style verity/s97is/locale/english/styles
-start http://www.verity.com -proxy proxysvr: 8010
```

### Using an authorization file (for password challenge)

```
vspider - collection mycoll_3.clm
        -style verity/s97is/locale/english/styles
        -start http:// www.verity.secure.com/index.htm
        -auth authfile.txt
```

### VSPIDER Options

```
-style      Style directory (if metaweb add .clm to collection name)

-start      Starting URL or URL's for the spider to follow
            -start http://www.verity.com http://www...

-collection Path to where the collection is to be created
```

### Indexing options:

```
-allold     forces reparsing of all documents in the collection

-cgiok      allows indexing of URL's containing the ? symbol to permit
            indexing CGI's. Default is to exclude these.

-domain     limits indexing to specified domains

-exclude     excludes files matching a specified expression
            exclude *bat *exe

-host       limits indexing to specified hosts.

-include     includes files matching a specified expression
            include *rpt

-maxdocsize maximum size of document to index (default is 1 megabyte).
            Enter values in kilobytes

-mimeexclude specifies mime types to be excluded. -mimeexclude text/html

-mimeinclude default is to include all mime types - specify only one

-nofollow   disables the following of href commands

-norobo     specifies that any robots.txt files encountered are ignored.
            The default is to honor all robots.txt files.
```



`-nouupdate` disables updating of any documents already in the collection

`-pathlen` allows you to specify how far down a URL or directory path the spider should go. The web host name is not included (`www.verity.com:9000/`) but elements following it are. Each slash counts like `this/that/then/what` would be `-pathlen 4`.

`-submitsize` specifies the number of documents to be submitted at one time (default is 50)

`-topicset` specifies the topicset to be used for indexing the collection

`-unlimited` specifies no limits are to be placed on the spider, if neither `-host` or `-domain` is specified. The default is to limit based on the host of first URL listed. Also removes the 1 megabyte `maxdocsize` default unless specified.

`-verbose` displays summary information for each URL or file accessed. Indicates indexed, ignored or failed.

`-debug` displays debugging-level messages (more than verbose)

`-trace` provides highest level of information about indexing.

`-auth` specifies the name of the authorization file for sites requiring a password.

`-casesen` makes processing case-sensitive so the spider separately keeps differing only in case.

`-charmap` specifies character map to use

`-common` specifies path to VDK home.

`-datefmt` specifies import format for date Y-M-D M-D-Y etc.

`-help` displays syntax options

`-language` specifies local language

`-jobs` maximum number of sockets to allocate (default is ten)

`-msgdb` specifies the path to the `ind.msg` message database

`-noindex` specifies to submit, but not index (used with `mkvdk-persist`)

`-noproxy` specifies the host listed should bypass the proxy server (wildcards ok `*.mysite.com`)

`-proxy` specifies the host & port for the proxy server

`-purge` purges the contents of the collection specified

`-repair`            repairs a damaged collection  
`-temp`             specifies directory for temporary files (default is /tmp)

## BUILDING FILE-BASED COLLECTIONS WITH MKVDK

The mkvdk utility is an all-purpose collection maintenance tool. When you use this utility, you supply command-line options to tell mkvdk what you want to do. For example, when you are creating a new collection you will provide different options than when you want to do collection maintenance. On-line help with mkvdk is available by entering: `mkvdk -help`. Your Collection Building Reference Guide includes a full description of each of the values. A complete list with brief description is included in the Additional Materials section of this workbook. Here are some you will use often.

### General Processing Options

<code>-collection colls/memos</code>	The collection name to be created or opened
<code>-synch</code>	Performs work immediately, rather than in the background as processing time permits
<code>-about</code>	Shows information about the collection (description and date last modified)
<code>-datapath ../db</code>	Specifies a path to use to find documents. All relative paths will be relative to this setting. If not used, mkvdk looks for the documents next to the collection directory.

### Collection Creation Options

<code>-create</code>	Specifies the collection named in the <code>-collection</code> option is to be created. If it already exists, mkvdk exits and does not write over it.
<code>-style styles/wysiwyg</code>	Only used in conjunction with the <code>-create</code> option, this option specifies the path to your selected style directory.
<code>-description "My Memos"</code>	Sets the collection's description. Place in quotes.
<code>-words</code>	Builds the word list for each of the individual partitions in the collections

### Document Processing Options

<code>-extract</code>	Indicates to parse field values as defined in the <code>style.tde</code> file.
<code>-insert</code>	Adds documents to the collection (this is the default for mkvdk).
<code>-update</code>	Add documents to the collection, replacing all previous information about the specified documents.
<code>-delete</code>	Deletes documents from the collection

### Messaging Options

<code>-quiet</code>	Overrides standard <code>-outlevel</code> (which is set to 15 to show fatal, error, warning and status messages) and only shows fatal and error messages.
---------------------	---

The process of building a collection with mkvdk is accomplished in three steps:

- Build the collection style
- Locate the documents to be indexed and put their names into a flist
- Run the mkvdk utility that creates the collection from the style files, flist, and other parameters required

The easiest way to create a new style is to borrow from an existing style and make any necessary changes or additions. The second step, building an flist, can happen in one of many ways. You could just type the names and paths directly into an ASCII file and use that as your flist, or you can use external utility programs that will help you build the list automatically. The third step requires that you run the mkvdk utility with the command-line arguments to create the kind of collection you desire. Housekeeping and optimization are handled automatically as the collection is built.

## PRACTICE LAB #2

### CREATING COLLECTIONS

1. In the IS97 folder, you will find a **colls** directory (the master directory containing the collections you will create), and a **docs** directory containing sub-directories of sample documents.
2. Open Netscape and return to the Information Server's Welcome page (you should have a bookmark (<http://train1.verity.com:7000/search97secure/s97a.vta?>)). Click the "Configure Server" button to return to the Quick Start. If you don't already have a bookmark for this page, now would be a good time to make one. From the Quick Start, select the Index Manager from the menu bar.
3. Create a new collection called "**legal**" and use "Miscellaneous Legal Documents" as the description. Remember the collection has to be created empty first, then you add an indexing task to actually perform the indexing. Enter the full path to the collection directory: **c:\is97\colls\legal**
4. In URLs to index, enter: **file://c:\is97\docs\doc1**

Click the Submit button to start the indexing task. When your indexing task has been submitted, click on the View Current Indexing Tasks icon.

5. The Current Indexing Tasks form displays current tasks, even if finished. You will notice that there are several indexing tasks submitted. For each indexing task you will see the name of the collection, the path to where the collection is created, and the status. While indexing is performing, a working icon is displayed. When finished the icon will change. Click on the Collection Manager icon from the menu bar.
6. From the Collection Manager, click on the View Collection List icon to see all of the completed collections. You should now see your new collection.
7. To search against your new collection, click the Quick Start icon, choose Search Your Website, and select your new collection from the pop-down list. Enter a word or phrase and click the Search button. If you do not have results in your collection, please notify your instructor.

### Building Collections Using VSPIDER

8. Click Start and open an MS-DOS Command for some practice building collections with **vspider**. This indexer can be used to index documents managed by a remote http server (with appropriate licensing). You will use the command line to build your collection using the main style directory, as indicated in the example below. Substitute your "site of

choice” for the **-start** command, and build your collection. The vspider indexer has already been added to your environment path which means, if you are at the c:\> prompt, you will not have to enter the full path to vspider.exe. If you are in another directory, enter the full path c:\search97\\_nti31\admin\vspider.exe. Enter the following command on one line:

```
c:\search97\_nti31\admin\vspider.exe
-collection c:\is97\colls\mysite.clm
-style c:\search97\s97is\locale\english\styles
-start http://www.yoursite.com (choose a small site to index for yoursite)
```

## Browse an Existing Collection

9. View the information that is contained in your new collection using the browse utility. To do this, you will need to move to the parts directory under your collection (c:\is97\colls\mysite\html\parts). Type **dir** to view the partition contents. You can use the browse utility to view document attribute information contained in any of the files that end with the **.ddd** extension. Choose one of these now and substitute it in the command shown below. Don't be surprised when the information flies by quickly in this window, in a moment we will have you redirect the output to a file. Enter the following command to view a partition:

```
c:\search97\_nti31\bin\browse.exe 0000001.ddd
```

A short menu will appear. At this point you can press Enter and the first record will be presented.

```
?) help
q) quit
c) Number of entries in field
_) Toggle viewing fields beginning with '_'
v) Toggle viewing selected fields
##) Display all fields in specified record number
Dispatch/Compound field options:
n) No dispatch
d) Dispatch
s) Dispatch as stream
Action (? for help):
```

There is a lot of information contained in the **.ddd**. As this is a very basic utility you will have to anticipate the commands as you redirect the output to a file. Enter the last command again (use up arrow to recall the previous command) adding the redirect as shown below:

```
c:\search97\_nti31\bin\browse.exe 0000001.ddd > browse.txt
```

**Press Enter twice (first time to execute browse and second time for the first record). Now type quit and your prompt will return. Enter dir to see the file that was created and use the (type) command to view the contents: type browse.txt**

Your log file should look like this.

```
browse - Verity, Inc. Version 1.5.6 (_nti31, Nov 18 1996)
BROWSE OPTIONS
  ?) help
  q) quit
  c) Number of entries in field
  _) Toggle viewing fields beginning with '_'
  v) Toggle viewing selected fields
  ##) Display all fields in specified record number
Dispatch/Compound field options:
  n) No dispatch
  d) Dispatch
  s) Dispatch as stream
Action (? for help):

Record number: 0
0  _DDFLAG          FIX-unsq ( 1) = 0x00
1  _DDVALUE         VAR-text ( 0) =
2  _DDVALUE_OF     FIX-unsq ( 4) = 0
3  _DDVALUE_SZ     FIX-unsq ( 2) = 0
(followed by the rest of the field information...)
```

## Updating and Deleting Documents in the Collection

10. Create another collection for practice on updating and deleting documents. Using **vspider**, build this collection as shown below and note that we have turned on advanced logging (called **-trace**) to learn more about the indexing process. Again, redirect the output to a file for your review. Enter the following command on one line:

```
c:\search97\_nti31\admin\vspider.exe -collection
c:\is97\colls\personal.clm
-style c:\search97\s97is\locale\english\styles
-start c:\is97\docs\doc5 -trace > build.txt
```

11. Another utility that ships with the Information Server is **rcvdk**. This utility provides a little command line retrieval client. Review the information that is contained in your **personal** collection using the **rcvdk** utility, by moving to the bin directory (**c:\search97\\_nti31\bin**) which contains all the utilities you've been working with, and entering the following command.

```
rcvdk.exe c:\is97\colls\personal.clm (press Enter)
```

The following message will appear:

```
Attaching to collection: c:\is97\colls\personal.clm  
Successfully attached to 1 collection.  
Type 'help' for a list of commands.  
RC>
```

Type help to view the list of commands, and practice with these a bit on your collection.

12. Because information does not remain static, vspider comes with utilities which allow the updating of information already in the collection. This exercise will give you the opportunity to delete a document and appropriately update the collection, and then to change a document, and again see the update process. Remember that your “personal” collection was created using the documents in the doc5 directory (c:\is97\docs\doc5). Move to the **doc5** directory and delete the file called “Upside.doc.” Run **vspider** to update the collection as shown below:

```
c:\search97\_nti31\admin\vspider.exe -collection  
c:\is97\colls\personal.clm
```

13. Using what you already learned about rcvdk, take a look at the collection and note the document has been deleted.
14. Move back to the **doc5** directory (c:\is97\docs\doc5). Open the “Scan.doc” file and change the date of the document to today’s date and add your name. Save your changes and run vspider once more to pick up your changes:

```
c:\search97\_nti31\admin\vspider.exe -collection  
c:\is97\colls\personal.clm
```

15. Using rcvdk, search for your name.
16. Add some of your collections to your server (and remember to reference your collections at the .clm level. Do you remember why?



## INDEXING TOPICS AGAINST A COLLECTOIN

1. In order to take advantage of search performance benefits provided by topics, you must first create a compiled topicset of your outline file, then index the compiled topicset against a collection. Open the IS97 directory and explore the topics directory. This directory already includes a news.otl file you will use to create a compiled topic set. There is a batch file called Bld\_top.bat that is already set up to execute the mktopics.exe utility on the news.otl file or you can use the mktopics.exe at the command line.
2. Open the news.otl file using Wordpad or Notepad. This file contains many different topics. Take a moment to review the different topics and their structure. Also, notice the **control: 1** at the top of this file. This keyword indicates that this file will be parsed when read by the Search97 compiler. For this reason, format is important.
3. Now you're ready to compile your news.otl file. You may choose to use the mktopics.exe utility at the command line or execute the batch file (bld\_top.bat). To run it from the command line, use the mktopics.exe utility, which we have copied into this directory (it originates in the bin directory). At the command prompt, type the following:

```
mktopics.exe -topicset c:\is97\topics\news -outline c:\is97\topics\news.otl
```

To use the batch file, enter the following: **bld\_top.bat**

Once the topicset has compiled, you will have a sub-directory called news in the topics directory.

4. Create a new collection called "misc" using the documents in the c:\is97\docs\doc7 directory, and indexing them using the compiled news topicset. At the command prompt, type the following:

```
vspider -topicset c:\is97\topics\news -collection c:\is97\colls\misc.clm  
-style c:\is97\styles -start c:\is97\docs\doc7
```

5. When the indexing is complete, and before launching the Information Server, you will need to update your inetsrch.ini file to add the directory path to your topicset. To do so, move to c:\search97\\_nti31\bin directory and open the inetsrch.ini file using Wordpad. You will see the [Common] area towards the top of this file, enter the path to your compiled topic set as shown below:

```
[Common]  
Topicset=c:\is97\topics\news  
topichome=c:\search97\common  
ProductLongName=SEARCH'97 Information Server V3.1
```

```
ProductShortName=Information Server
PatchLevel= 10
ProductHomeLink=http://www.verity.com
VTopicScriptName=/search97cgi/s97_cgi.exe
```

Save your edits, then exit inetsrch.ini file.

6. Changes made to the inetsrch.ini file require restarting the server to read in new changes to the configuration file. To stop and restart the server, select **Services** from the Control Panel. Scroll down the list of services and click on **Search97 IS Admin** to highlight it, then click the "Stop" button to stop the service, and then "Start" to restart the service. Close the Services dialog and launch the Information Server.
7. From the Collection Manager import your new collection **misc**.
8. Using the default search page under Quick Start, select the **misc** collection from the pop-down list, and enter the topic **politics** in the search box. Click the Search button.
9. Open several of the documents in the results list and notice the highlighted words include terms contained in the politics topic and not only the word politics.

## ENABLING SEARCH AT THE SERVER

One of the most important things you can do to ensure a successful application is to present information to your users, about how to search and how to work with results. Information applications are more about this than anything else and it is up to you to know how your data can best be used and to support a variety of user experience levels in your applications. Knowing how to query effectively will help you design information systems that make using the Verity Query Language easy.

### EXPLORING THE QUERY LANGUAGE

The value of a query language is seen in two areas: how easy is it for a novice to ask for information and get good results and how rich is the language for knowledge workers to precisely target specific results?

For the novice, the query language provides defaults

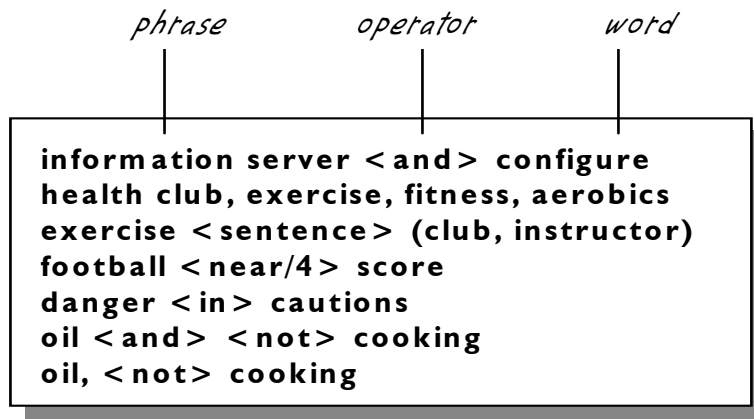
- standard variant endings
- density tie-breakers
- accrue on groups of words
- ranking by score with highest first

For the knowledge-worker, the query language provides

- a rich set of operators and modifiers
- parenthetic representation of complex ideas
- weighting of terms or groups of terms
- topics

### WHAT IS A QUERY?

A *query* is simply the criteria you provide for performing a search. When you create queries, you can combine words, phrases, fields and topics with operators and modifiers to direct which documents will be selected and how they will be ordered on the results list



## SYNTAX ALTERNATIVES

When you use *simple syntax* (the default), the query is interpreted with a broad focus. Your searches are case-insensitive and the STEM operator is automatically applied to search words, by selecting root word and standard variant endings. The MANY modifier is applied for search words to score documents higher based on word density words are automatically interpreted as topics when a matching topic exists. Groups of words are evaluated using Verity's proprietary ACCRUE operator at the parent level, to specify selection of any of the words entered, but higher scoring for those documents containing additional occurrences of unique words

When you use *explicit syntax*, you instruct the engine about how the search is to be handled. There are shortcuts for some explicit syntax operators and modifiers:

- <WORD>film or "film"
- <STEM> film or 'film'
- <SOUNDEX> @film@

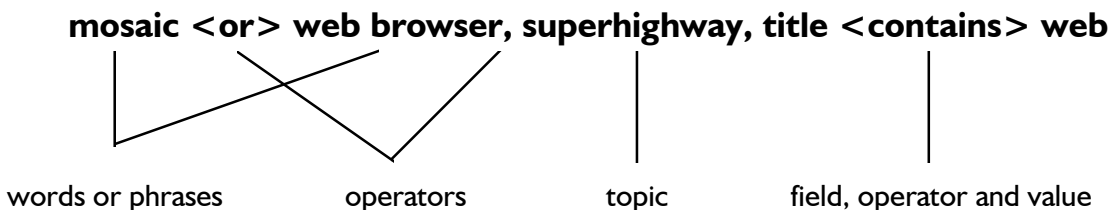
When you want to search the text of the documents, you may do so by:

- Query** entering search words directly  
**information superhighway, online provider, internet, web**
- Topic** entering the label specified for a group of words already defined or organized to represent a more complete subject  
**superhighway**

## QUERY COMPONENTS

Your queries can include any of these components:

- **WORDS** representing words in your documents (and found in the internal word index)
- **TOPICS** representing predefined groups of words and information about how the words relate to each other
- **FIELD VALUES** representing the value you will find in a particular field defined by the DBA
- **OPERATORS** to specify how a group of words should be searched



## OPERATOR CLASSES

**Evidence Operators** search for words and can expand into a list of related search words, depending on the operator selected. **Proximity Operators** are used with groups of words to define how closely they are related to each other. **Concept Operators** combine the meaning

of search words to identify a concept in a document. **Relational Operators** are used with fields. **Boolean Operators** are used with topics and words to retrieve the elements you describe without operator precedence conflicts.

### EVIDENCE OPERATORS

Evidence Operators represent your actual search words. Depending on the operator used, they may expand into a list of related search words:

<u>Operator</u>	<u>Shortcut</u>	<u>Rule</u>
<word> film	"film"	must locate an exact match on the word as entered (no variant endings to be included)
<stem> film	'film'	must locate a match on the root of the word and includes all standard variant endings (filming, filmed, films). This is the default.
<thesaurus> film		must search for all synonyms listed in the thesaurus for this word
<wildcard> tech*		must match the character string entered with selected variables <i>fil*</i> <i>substitutes any characters for *</i> <i>fil?</i> <i>substitutes single letter for ?</i> <i>fi[ln]e</i> <i>substitutes specified letters in brackets</i> <i>film(s,ed)</i> <i>substitutes specified variables only as included in parentheses</i>

### PROXIMITY OPERATORS

Proximity Operators define how closely words must be found near each other to qualify your subject. Proximity refinement often improves query results dramatically.

<u>Operator</u>	<u>Shortcut</u>	<u>Rule</u>
<phrase> nice job	nice job	must locate words next to each other, in the order defined
new <sentence> film		must locate words in the same sentence (any order)
hit <paragraph> film		must locate words in the same paragraph (any order)
weather <near> report		must locate words within 1000 words of each other and reflects proximity by score next to each other has highest score)

**football <near/5>score** must locate words within the number of words specified by /n

**danger <in> cautions** locates documents containing values in specific zones (like title). Only available for web-based collections or where DBA captures zones during indexing process.

### CONCEPT OPERATORS

Concept Operators combine the meanings of a group of search words to identify a particular subject or concept in a document:

<b><u>Operator</u></b>	<b><u>Shortcut</u></b>	<b><u>Rule</u></b>
<b>&lt;accrue&gt;</b>	,	matching documents must contain at least one of the words entered but the more unique words, the better. This is the default.
<b>&lt;and&gt;</b>		matching documents must contain all of the words entered
<b>&lt;or&gt;</b>		matching documents must contain at least one of the words entered

### RELATIONAL OPERATORS

Relational Operators are used with fields

<b><u>Operator</u></b>	<b><u>Shortcut</u></b>	<b><u>Rule</u></b>
<b>&lt;contains&gt;</b>		the string must be found within the field. This is the default.
<b>&lt;starts&gt;</b>		the field must start with this value
<b>&lt;ends&gt;</b>		the field must end with this value
<b>&lt;matches&gt;</b>		the field must contain a matching string
<b>greater than</b>	> (only)	the numeric field value must be greater than this number
<b>less than</b>	< (only)	the numeric field value must be less than this number
<b>equals</b>	= (only)	the numeric field value must be equal to this number

## BOOLEAN OPERATORS

Boolean Operators are used with topics and words to retrieve the elements you describe without operator precedence conflicts

<u>Operator</u>	<u>Shortcut</u>	<u>Rule</u>
<any>		matching documents must contain at least one of the words entered
<all>		matching documents must contain all of the words entered

## MODIFIERS

The behavior of operators can be modified or enhanced:

The <NOT> modifier is used to exclude documents

oil <AND><NOT> cooking

The <MANY> modifier is used to count the density of a word or phrase topic within a document

<MANY><WORD> earning

The <CASE> modifier is used to perform a case-sensitive retrieval on a word

<CASE><WORD> NeXT

The <ORDER> modifier is used to indicate the order of the words you have entered important

diver <ORDER><NEAR/5> kills <ORDER><NEAR/5> shark

## THE IMPORTANCE OF TOPICS

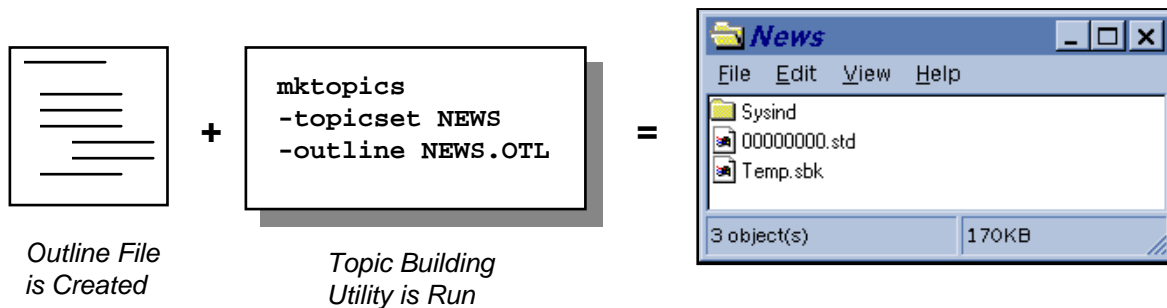
You can dramatically enhance the value of your applications and simplify searching for your users by incorporating topics. Topics represent proven knowledge about a particular subject. The more complex a subject, the greater the value of the topic. Differing points of view or levels of expertise can be addressed by your topics and searching is so much more effective as even novices benefit from expert knowledge captured in the topics

- Topics save money by eliminating redundant work and speeding you to the right information
- They can include words, phrases, field values, and the vast array of relationships between them
- Topics include all of the components of the query language (operators, modifiers, and weights)

- Topics can be indexed against collections. Searches using indexed topics are more than 30 times as fast as those without.

### THE TOPIC BUILDING PROCESS

Topics are built by constructing the knowledge in a text file called an outline (OTL) file. Words are grouped in an information tree and operators, modifiers and weights are applied. Structure and syntax are important. The topics are then compiled by running the topic set creation utility, called `mktopics`, against the OTL text file. A special index is created which consists of the terms, relationships and when indexed against a collection of documents, the locations of matching documents.

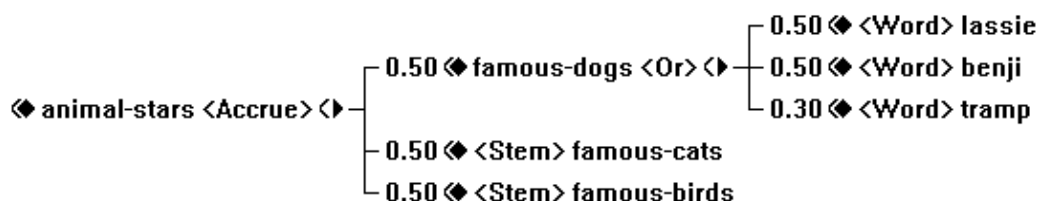


Topics are typically defined to serve as a knowledgebase for commonly asked questions and include definitions of relationships between words and rules for evaluating and scoring documents that are relevant to the search objective:

**Evidence topics** are search words, phrases or values that indicate the evidence or presence of your subject.

**Branch topics** represent subject categories within the general subject. Each branch of the tree defines a separate cluster of related subjects within the main topic. These are the subtopics

The **Root topic** (top-level topic) is a name or label which describes the general subject covered by the topic. This is the main topic.





### Practice Lab #3 Practicing with Query Language

1. Select the Users Guide from the Quick Start (right under the Forms Wizard). Click on the link to Search'97 Search Tips. This section has been designed to help you understand the query language and how to write really effective queries. Go through this section and then using the documents you have available through your Information Server, try each of the operators. Remember that you need to know a little something about what is in the documents to make your search practice valuable, so look at them first.
2. For those operators having shortcuts, practice using the shortcuts in your query strings.
3. Add a more complex query by using parentheses to select groups of documents. For example:  
  
<ANY> (stock, trade, securities ) <SENTENCE> <ANY> (market, exchange)
4. List some pre-defined queries that would be helpful at your site and be prepared to discuss how you could refine them for more precise results.

#### Build a Basic Topic

1. Create an outline file called **verity.otl** using wordpad and save it under the c:\is97\topics directory. Edit this file by creating a new topic called **verity**. Remember to include the keyword **control:1** at the top of the file.

Below are the components and search terms you will use to create your new topic. Organize each search word or phrase under the appropriate subtopic category:

Topic Name:            verity

Subtopic Names:       verity\_products  
                          verity\_services  
                          verity\_technologies

Search Words:         search97, agent server, topic, search97 information server, topic agents, s97 information server, verity technical support, verity, verity consulting, document indexing, cd-web publisher, vspider, verity sales, document retrieval, verity training, search enabling, knowledgebase creation, push technology

When complete, save your edits and exit verity.otl file.

2. Create a compiled topicset on your verity.otl file. In the c:\is97\topics directory you will find a mktopics.exe utility you can use to compile your verity.otl file or simply update the Bld\_top.bat file and execute this batch file.

At the command prompt type the following:

```
mktopics.exe -topicset c:\is97\topics\verity -outline c:\is97\topics\verity.otl
```

3. At this point, you will have a new compiled **verity topicset** directory. Edit the inetsrch.ini file to replace the current topicset with the new one that you have created. It is important to know that you can only have one topicset referenced in this configuration file .

In the [Common] area towards the top of this file, enter the path to your new topicset as shown below:

```
[Common]  
Topicset=c:\is97\topics\verity  
topichome=c:\search97\common  
ProductLongName=SEARCH'97 Information Server V3.1  
ProductShortName=Information Server  
PatchLevel=10  
ProductHomeLink=http://www.verity.com  
VTopicScriptName=/search97cgi/s97_cgi.exe
```

4. Save your edits, then exit inetsrch.ini file. Remember to stop and restart the Information Server.
5. Launch Information Server Admin, and use any of the default search forms to test your topics. s