Basic Collection Building

- What is a Collection?
- Tools for Building Collections: Utilities & Spiders
- The Collection Servicer Utility
- Managing Your Collections
- Adding Topics and Indexing Against Collections
- Working with Utility Programs
- Practice Lab

What is a Collection?

- A Verity Collection is a series of indexes which work together to enable the searching of documents
- Quality searching requires
  - Access to attributes about the document
    - Title <contains> Verity
    - Date > 1/1/98
  - Limiting to zones in the document
    - Verity <in> Title
    - Danger <in> Subject
    - (danger, caution, warning) <IN>(h1, h2, title)
  - Proximity information on words in the document
    - Verity <sentence> new products
    - Printer <near>/4 problems
    - White House
- Verity's indexing utilities automatically capture values for these types of searches and write this data to a collection

How Are Values Populated?

- Each collection has a style directory that contains the rules for populating the collection's fields, zones and word indexes
  - The indexer "senses" each document's type and load rules for processing that document
  - Standard fields and zones (Title, Date, Keywords, etc.) for various document types are pre-defined by Verity and are populated by automatic filtering utilities
  - You can define custom fields by modifying your style directory
    - Meta-tags in HTML documents are very easy to use

      ```
      <META name="Dept" content="Sales">
      ```

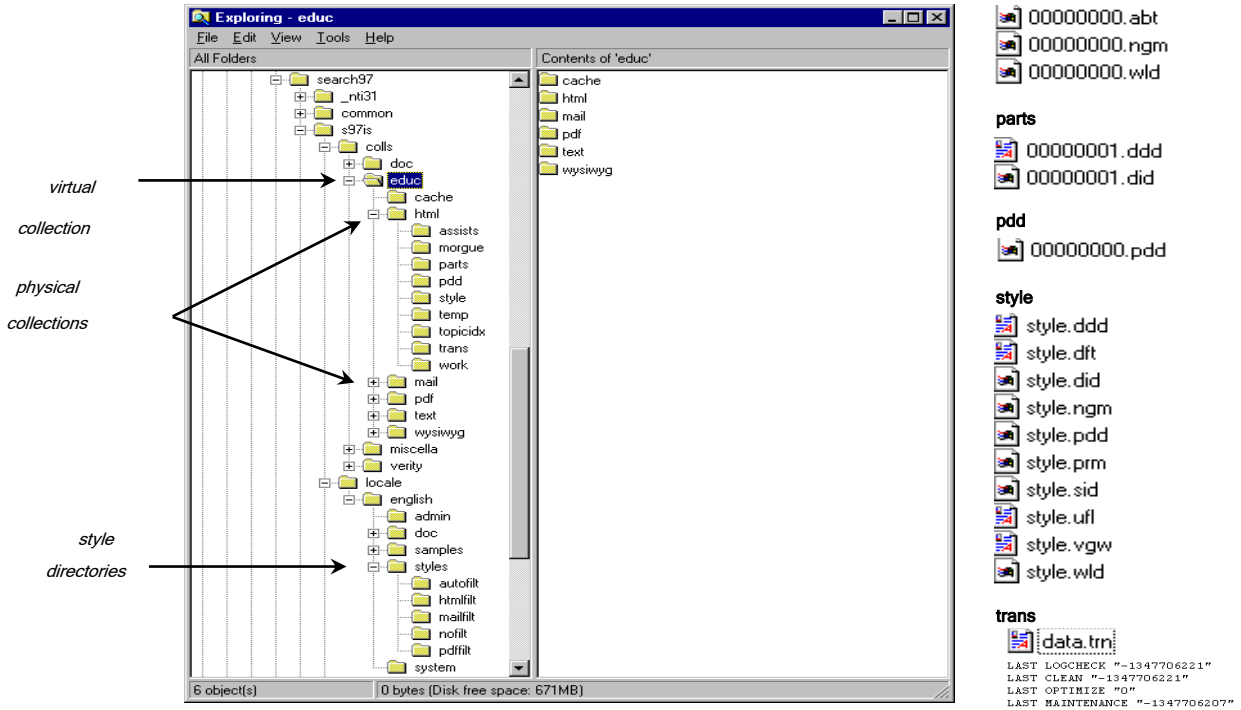
    - Document properties can be populated and extracted by the indexer
    - A "bulk-submit" file can be created which is handed to the indexer with all field values already defined

Document Indexes

- Actual document information is stored in the parts (partitions) directory
  - The ".ddd" stores attribute and location information
  - The ".did" stores word information
- Verity provides utilities you can use to better understand the information contained in your collection
  - **Browse** allows you to view the values for each of the attributes captured
  - **Didump** allows you to view words in the word index
- The Information Server also comes with **RCVDK**, a lightweight command line retrieval client which allows you to attach to one or more collections and perform full text searching, viewing of documents and displaying of results
- Your workbook includes information about these utilities and an exercise to try each of them

Collection Structure

- The collection directory contains sub-directories that maintain information about:
  - How the collection is to be created and managed (style directory)
  - Data captured from documents like fields, zones, words (parts directory)



How Indexing Works

- You provide 4 important pieces of information to begin the creation of a collection:
  - Which utility you will use (vspider or mkvdk)
  - Which documents you wish to include (starting directory, URL or specific file names)
  - What you will call your collection and where it will be located
  - Which style files you will use (path to the top-level style directory)

vspider -start http://www.yoursite.com
-collection d:\webcenter\colls\mycoll.clm
-style d:\verity\s97is\locale\english\styles

commands are entered on one line
there are many additional options you can add

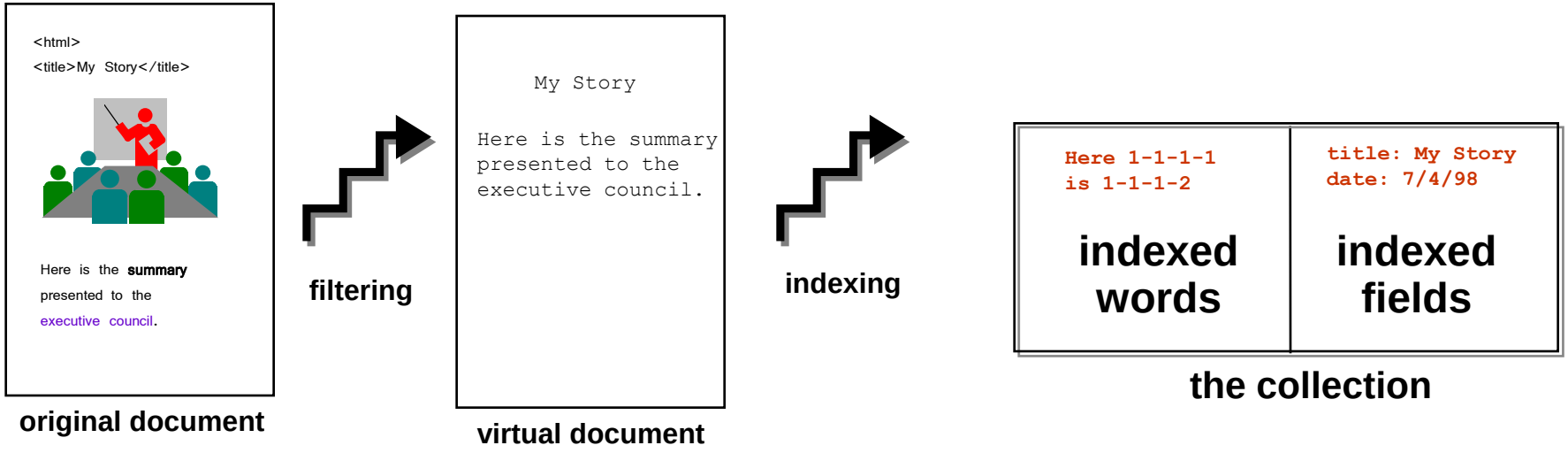
Accessing Documents

- The Verity indexing engine uses a gateway to access document files or other repositories of data (on the web, file system or in a database)
  - Default is file system (local host or mapped network drives)
  - Verity provides pre-defined gateways for access through HTTP servers and databases. Custom gateways can be defined.

- Gateways are specified in the style.vgw (**V**erity **G**ate**W**ay file). If this file is not found, the default gateway is used.
- Gateway fields are captured (vgw_url, mime-type, modified date, vgw_odbc: database fields)

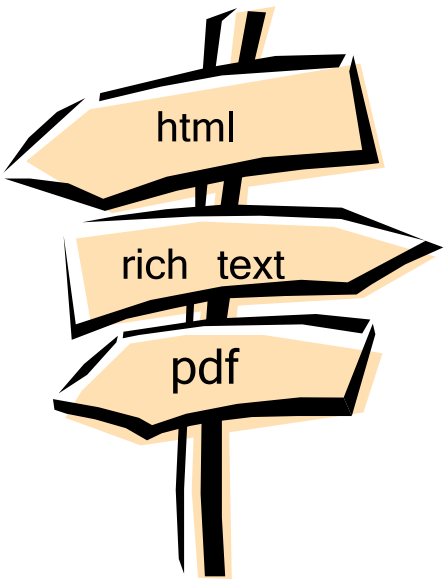
Filtering Documents

- Each document headed for the collection goes through a temporary conversion process to create a ***virtual document*** of indexable text
- An auto-recognition program identifies the type of document to be filtered and calls on one of the helper sub-filters to handle the processing of the selected document
  - Handles characters or binary data as appropriate
  - Removes general formatting tags
  - Creates word, punctuation, markup and field tokens



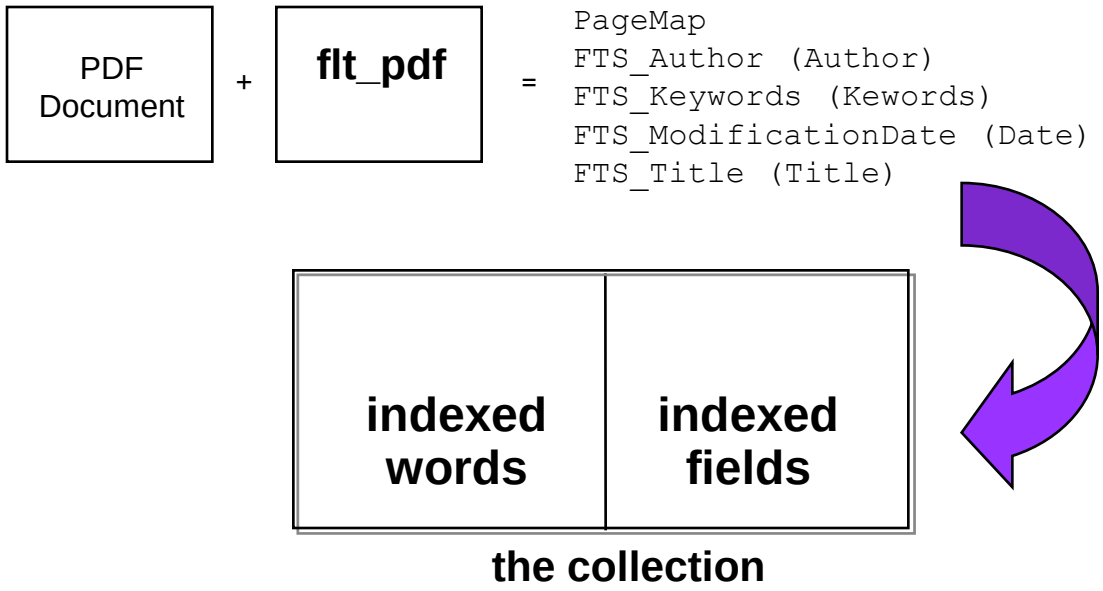
Helper Sub-Filters

- There are 3 helper sub-filters:
  - **flt_zon** is a Verity filter with options for html, email and news
  - **flt_kv** is a kit of KeyView filters enabling indexing of more than 45 of the most popular document formats (word processing, spreadsheets, presentations)
  - **flt_pdf** is a Verity filter designed to handle PDF documents
- The number of fields and content varies based on the type of filter that is used



Indexing Documents

- Document attributes (like title and author) are captured in a field index. They can be populated by gateways, document filters or bulk-submit files.
- Each document has a record in the index and the fields associated with it are determined by the style files that were used to create the collection
  - Verity identifies key fields for various document types
  - You can customize style files to add additional fields



Browsing Collection Contents

```
browse - Verity, Inc. Version 2.2.2 (_nti31, Sep 03 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-unsg (  1) = 0x00
1  _DDVALUE         VAR-text (  0) =
2  _DDVALUE_OF      FIX-unsg (  4) = 0
3  _DDVALUE_SZ      FIX-unsg (  2) = 0
4  _DBVERSION       WRM-text (  6) = vdk21
5  _DDDSTAMP        FIX-date (  4) = 03-Jan-1998 12:25:42 pm
6  _DOCIDX          VAR-text ( 12) =
7  _PARTDESC        FIX-text ( 32) = (Verity, Inc. Version 2.2.2) -
8  _FtrCfg          CON-text (  8) = 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-unsg (  4) = 32
13 _DOCIDX_SZ       FIX-unsg (  2) = 12
14 _STYLE           AUT-text ( 19) = ../style/style.ddd
15 _DOCID           FIX-unsg (  4) = 3
16 _SECURITY        FIX-unsg (  4) = 0
17 _INDEX_DATE      FIX-date (  4) = 03-Jan-1998 12:25:43 pm
18 _SECURITY_MI     WRM-unsg (  4) = 0
19 _SECURITY_MX     WRM-unsg (  4) = 0
```

menu options

field types

feature vectors for clustering and stored summaries

when document was indexed

more…

Browsing Collection Contents

```
20 _INDEX_DATE_MI    WRM-date (  4) = 03-Jan-1998 12:25:43 pm
21 _INDEX_DATE_MX    WRM-date (  4) = 03-Jan-1998 12:25:43 pm
22 VDKFEATURES       VAR-text (232) =
23 VDKFEATURES_OF    FIX-unsg (  4) = 539
24 VDKSUMMARY        VAR-text (271) = Beau Geste. Based on Christopher Wren's novel
about the undying devotion shared among three brothers serving in the French Foreign
Legion.  Gary Cooper and Ray Milland join a number of to-be-famous character actors in
this exciting adventure movie. Released: 1939 Date: 13 Nov 1993
25 VDKSUMMARY_OF     FIX-unsg (  4) = 857
26 VdkVgwKey         VAR-text ( 27) = c:\is97\docs\doc3\Rev1.txt
27 VdkVgwKey_IX      FIX-unsg (  3) = 2
28 VdkVgwKey_MI      WRM-text ( 28) = c:\is97\docs\doc3\MSG17.TXT
29 VdkVgwKey_MX      WRM-text ( 27) = c:\is97\docs\doc3\REV6.TXT
30 VdkVgwKey_OF      FIX-unsg (  4) = 88
31 VdkVgwKey_SZ      FIX-unsg (  2) = 27
32 DOC               DSP-text ( -1) = c:\is97\docs\doc3\Rev1.txt
33 DOC_FN            VAR-text ( 27) = c:\is97\docs\doc3\Rev1.txt
34 _CACHE_FN         VAR-text (  0) =
35 _CACHE_DELETE     FIX-unsg (  1) = 0
36 _ParentID         VAR-text ( 18) = c:\is97\docs\doc3
37 Title             VAR-text (  5) = Beau Geste Review
38 Ext               VAR-text (  4) = txt
39 Author            VAR-text (  0) =
40 Subject           VAR-text (  0) =
41 Keywords          VAR-text (  0) =
42 Comments          VAR-text (  0) =
43 Snippet           VAR-text (288) = Beau Geste. Based on Christopher Wren's novel
about the undying devotion shared among three brothers serving in the French Foreign
Legion. Gary Cooper and Ray Milland join a number of to-be-famous character actors in
this exciting adventure movie. Released: 1939 Date: 13 Nov 1993
```

auto-generated summary

primary key

the document locator

Index - Min/Max

Offset - Size

document file name

where document was found

document title as identified by the filtering helper used

fields built by default

but not populated by filter

first 400 printable characters

more…

Summary vs. Snippet

Different because the summary is limited to only the best two lines

```
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.

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...
```

Same because the document is so short and summaries was set to a 3 sentence maximum

```
24 VDKSUMMARY       VAR-text (271) = Beau Geste. Based on Christopher Wren's novel about
the undying devotion shared among three brothers serving in the French Foreign Legion.
Gary Cooper and Ray Milland join a number of to-be-famous character actors in this exciting
adventure movie. Released: 1939 Date: 13 Nov 1993

43 Snippet          VAR-text (288) = Beau Geste. Based on Christopher Wren's novel about
the undying devotion shared among three brothers serving in the French Foreign Legion.
Gary Cooper and Ray Milland join a number of to-be-famous character actors in this
exciting adventure movie. Released: 1939 Date: 13 Nov 1993
```

more…

Browsing Collection Contents

```
44 URL              VAR-text (  0) =
45 MIME-Type        VAR-text ( 11) = text/plain 46 Language
47 Encoding         VAR-text (  0) =
48 _Created         VAR-text ( 30) = Sun, 29 Dec 1997 23:37:35 GMT
49 _Modified        VAR-text ( 30) = Sun, 23 Oct 1997 02:27:44 GMT
50 Created          FIX-date (  4) = 29-Dec-1997 03:37:35 pm
51 Modified         FIX-date (  4) = 22-Oct-1997 06:27:44 pm
52 Size             FIX-unsg (  4) = 313
53 DOC_OF           FIX-unsg (  4) = 0
54 DOC_SZ           FIX-unsg (  4) = 4294967295
55 DOC_FN_OF        FIX-unsg (  4) = 88
56 DOC_FN_SZ        FIX-unsg (  2) = 27
57 _CACHE_FN_OF     FIX-unsg (  4) = 0
58 _CACHE_FN_SZ     FIX-unsg (  2) = 0
59 _ParentID_OF     FIX-unsg (  4) = 2246
60 _ParentID_SZ     FIX-unsg (  2) = 18
61 Title_OF         FIX-unsg (  4) = 2237
62 Title_SZ         FIX-unsg (  2) = 5
63 Ext_OF           FIX-unsg (  4) = 2242
64 Ext_SZ           FIX-unsg (  2) = 4
65 Author_OF        FIX-unsg (  4) = 0
66 Author_SZ        FIX-unsg (  2) = 0
67 Subject_OF       FIX-unsg (  4) = 0
68 Subject_SZ       FIX-unsg (  2) = 0
69 Keywords_OF      FIX-unsg (  4) = 0
70 Keywords_SZ      FIX-unsg (  2) = 0
71 Comments_OF      FIX-unsg (  4) = 0
72 Comments_SZ      FIX-unsg (  2) = 0
73 Snippet_OF       FIX-unsg (  4) = 2264
74 Snippet_SZ       FIX-unsg (  2) = 288
75 URL_OF           FIX-unsg (  4) = 0
```

if web document, URL here

mime type impacts viewing

date information:

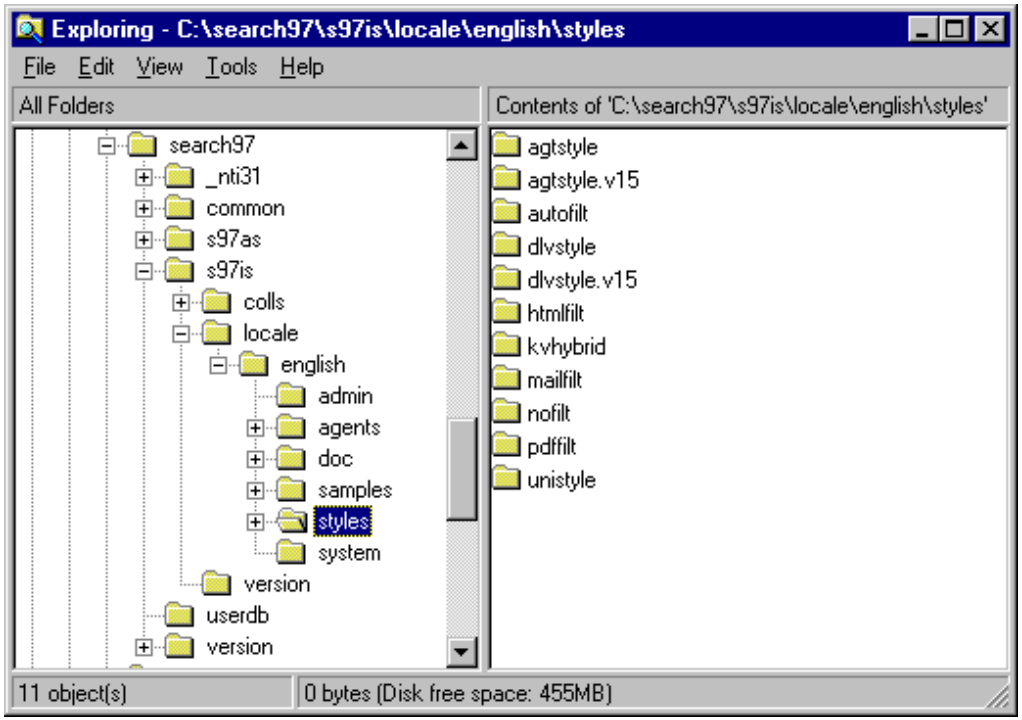
some as text fields

some as Verity-Internal Dates

```
76 URL_SZ           FIX-unsg (  2) = 0
77 MIME-Type_OF     FIX-unsg (  4) = 174
78 MIME-Type_SZ     FIX-unsg (  2) = 11
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) = 185
84 _Created_SZ      FIX-unsg (  2) = 30
85 _Modified_OF     FIX-unsg (  4) = 215
86 _Modified_SZ     FIX-unsg (  2) = 30
```

Configuring Collections

- Style files configure the indexes that store data about documents
  - Choices made in style files direct how indexing utilities will create and maintain collections
  - Information Server ships with a directory of styles for your use (default is shown below)
- Advanced Collection Building on day 3 covers style files for all document types and features, in greater detail



Indexing and Logging

- You can build or add to collections from either the GUI Indexer or the command line indexing utilities (vspider or mkvdk)
- Log files are automatically created and you can set the level of feedback you desire

  - For the GUI indexer set this in the inetsrch.ini file

```
LogLevel = Verbose    (less information)
LogLevel = Debug      (more information)
LogLevel = Trace      (most information)
```

  - For vspider, set this on the command line

```
-verbose     -debug          -trace
```

Skipping Documents

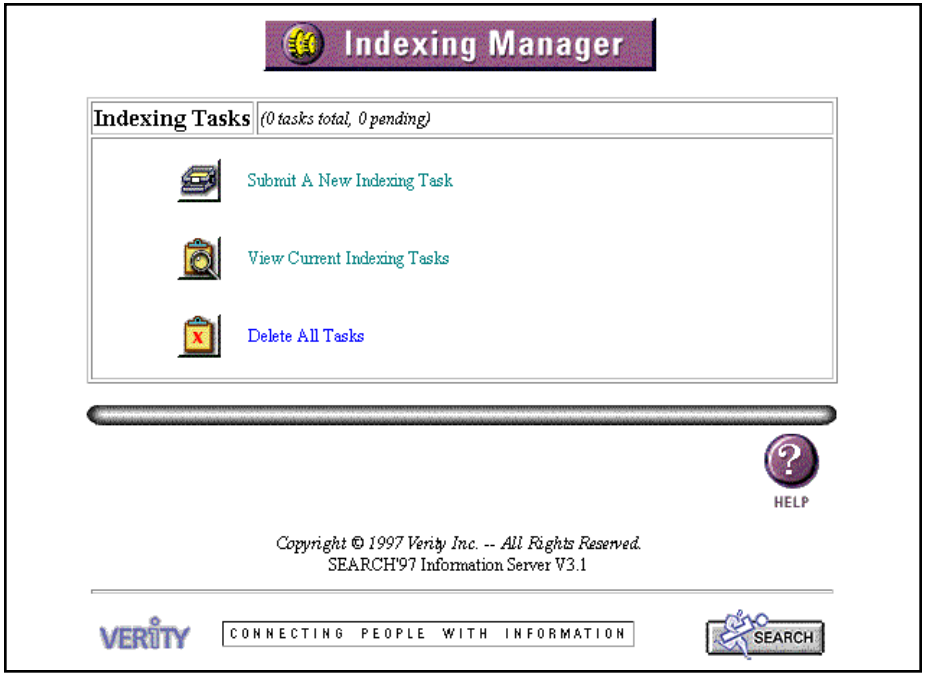
- You may see messages about skipped documents in the GUI Indexing Manager or in your log files
- Skipping messages: "Skipping key because of…"

`HOST`	`Document is not on the host you are indexing`
`DOMAIN`	`Document is not in the domain you are indexing`
`DATE`	`Document is up-to-date in the index and does not require reindexing`
`VDKVGWKEY`	`Path to the document does not match your -include statement or does match your -exclude statement`
`MIME-TYPE`	`The document is of a mime type you are not indexing`

- Also skips if off-site, forbidden access (401) or just doesn't exist (404)

The Graphical Indexing Manager

- The Indexing Manager organizes features related to indexing specific sites
  - Simple mode only requires a name, description and path
  - Advanced mode filters for MIME types, include or exclude patterns (file or domain names), and sets proxy information at the collection level
- Indexing tasks can be defined and submitted for initial creation and then maintained as current tasks for resubmission



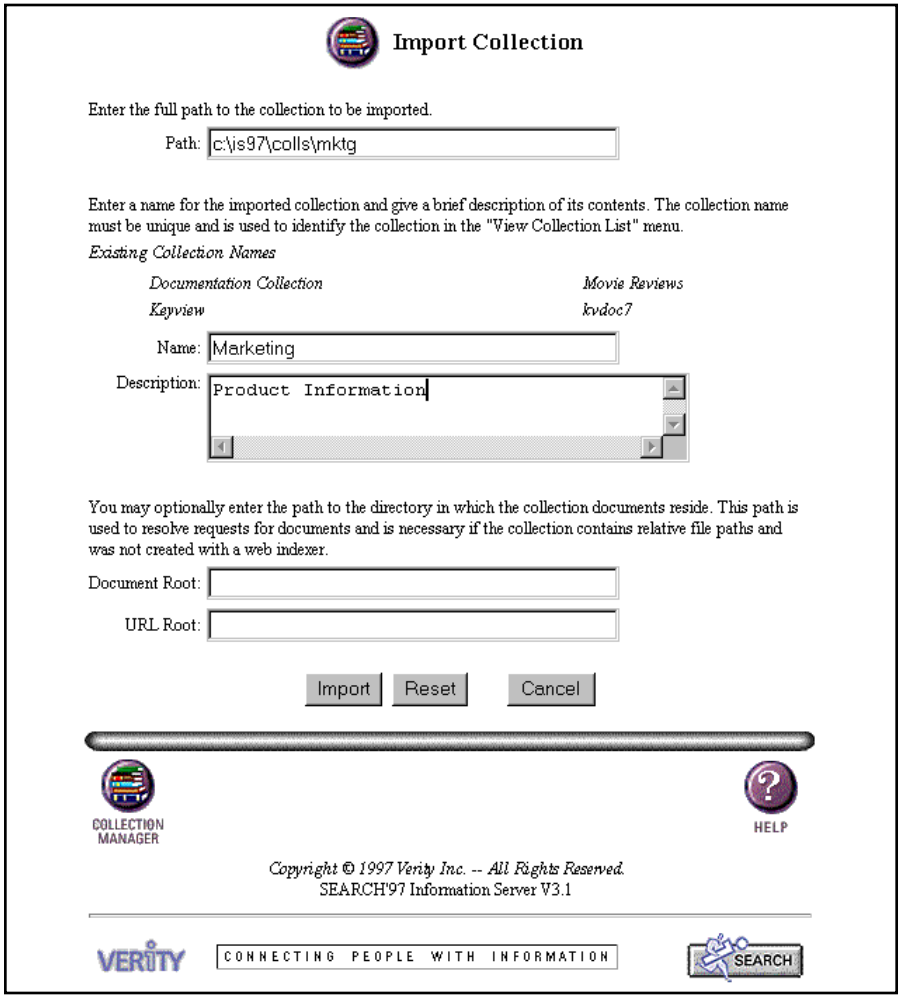
The Graphical Collection Manager

- The Collection Manager allows you to
  - Work with the current set of collections on your server
  - Import a new collection created or used by another Verity application
  - View information about the state and contents of each collection

Importing Collections

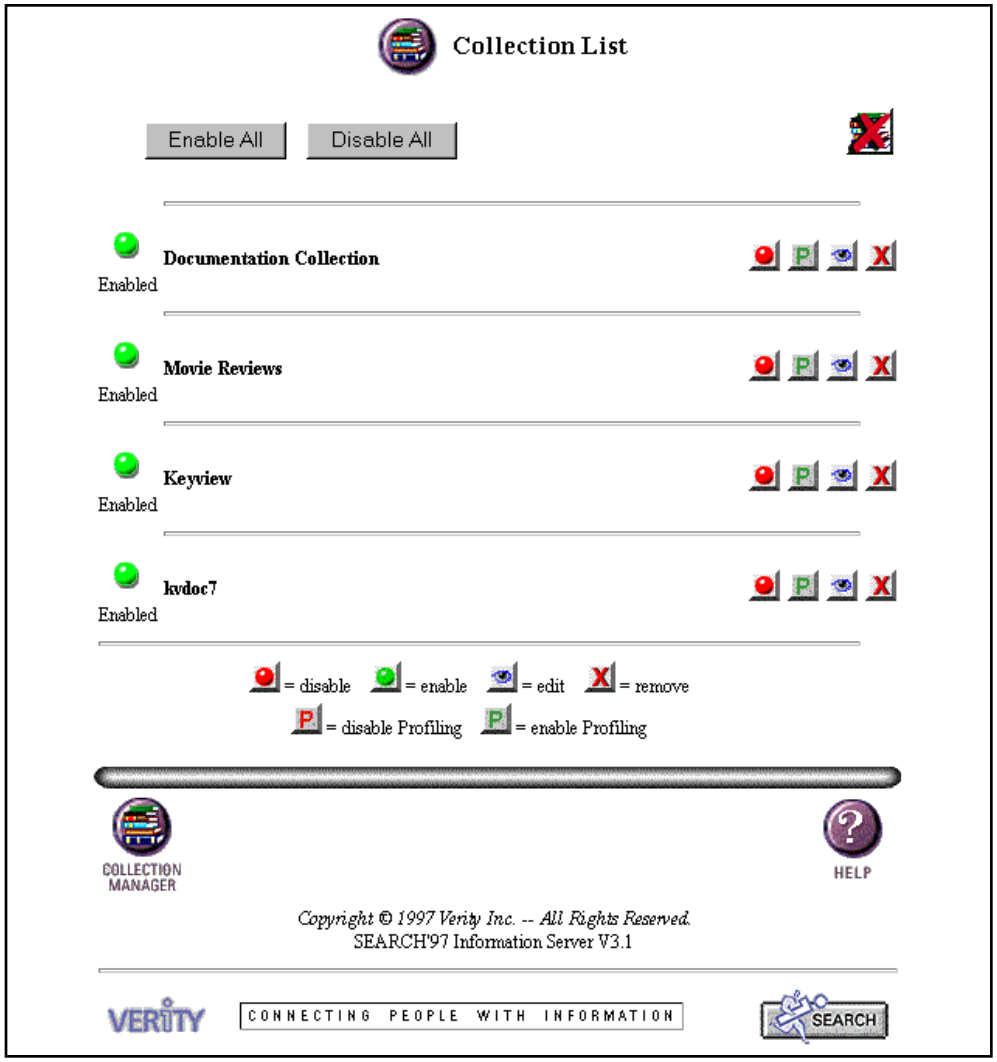
- Collections built with previous versions of Verity's web indexers import easily
- If you wish to take advantage of
  - The latest filtering options provided by KeyView filters
  - Performance improvements provided by the new vspider

  you will need to rebuild your collections



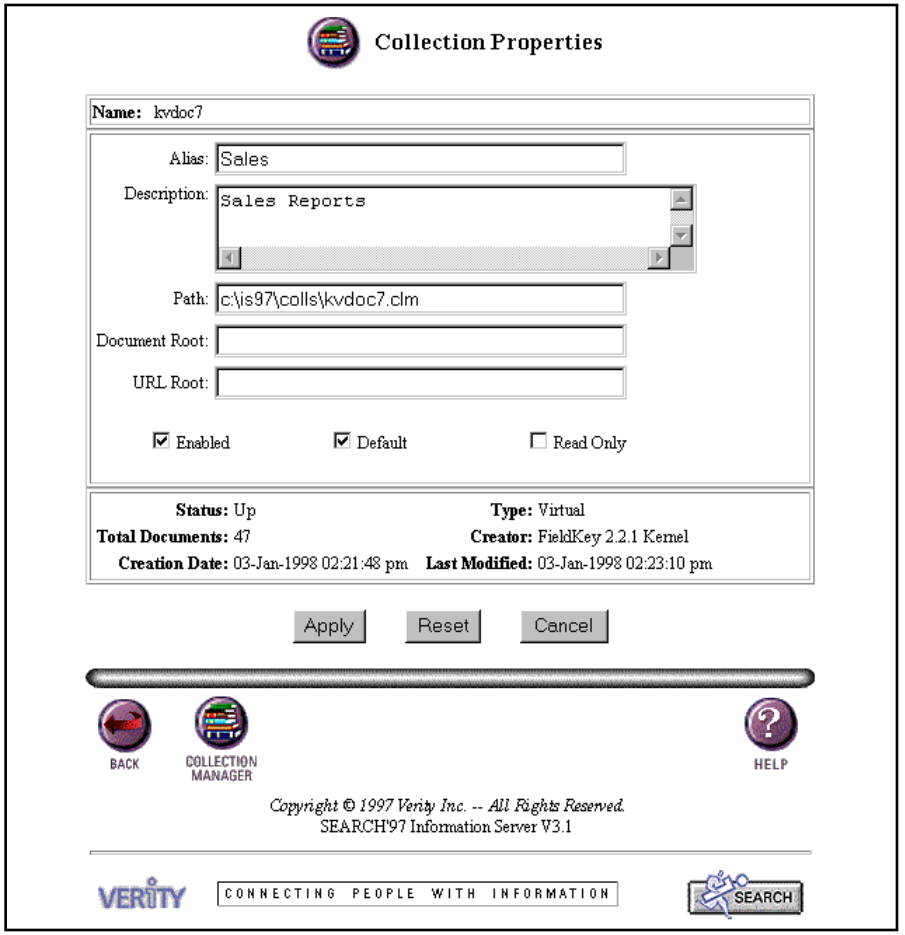
Using the Collection List

- As you build a number of collections, you will work with them on the Collection List
  - Allows you to easily enable and disable searching
  - "Edit" takes you to Collection Properties
  - You can remove collections from the list, but this does not delete them from the system



Collection Properties

- Collection Properties provides information about the current status of the collection
- It accepts edits for collection alias and description



The Command-Line Verity Spider

- The command-line spider (**vspider**) provides additional indexing options and greater control than available through the GUI spider found in the Information Server Indexing Manager
  - Walks the file system or crawls one or more web sites
  - Installed with Information Server by default to allow web spidering for the local host and the file system

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

vspider -collection mycoll.clm
-style \verity\s97is\locale\english\styles
-start c:\webctr\docs\

vspider -collection mycoll.clm
-style \verity\s97is\locale\english\styles
-start http://www.yoursite.com/
-exclude *secrets* -domain verity.com -proxy proxysrvr:8010
```

Features of Verity Spider

- This spider allows you to:
  - Specify multiple “start” URLs or starting directories
  - Force the re-parsing of all documents in the collection
  - Limit indexing to a particular domain or host
  - Include or exclude files matching regular expressions or mime types
  - Set a maximum size for documents to be indexed
  - Disable the following of links
  - Disable updating of documents already in the collection
  - Specify topics to be used when indexing the collection
  - Specify logging message levels (verbose, debug, trace)
  - Read command-line syntax for vspider from a file
  - Specify the import date format to use
  - Specify the number of URLs to be streamed simultaneously (default is 10)
  - Disable DNS look ups (for faster spidering)
- A complete list of options are available in your workbook

Verity Spider Licensing

- Licensing options include
  - File walking through any files available on the network (automatically included with any of the other options)
  - Web spidering of local host for files and links contained on the Information Server's host machine (default)
  - Web spidering of default domain for the local host
  - Web spidering of remote sites
- Host refers to physical machine - these are different hosts
  - http://www.verity.com
  - http://uk.www.verity.com
  - http://web.verity.com
- Domain refers to the last two entries of the DNS name, regardless of location

Authentication

- Some sites secure documents by requiring authentication through their web server
- You can use the -auth option to specify the name of your authentication file. This file includes the server name, realm, username and password:

```
#server          #realm   #username   #password
www.verity.com    field    Alladin     "sesame"
```

- Storing the username and password allows users to see all documents within the authenticated path
- Users will be able to view highlights within the retrieved documents

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

Recognized MIME Types

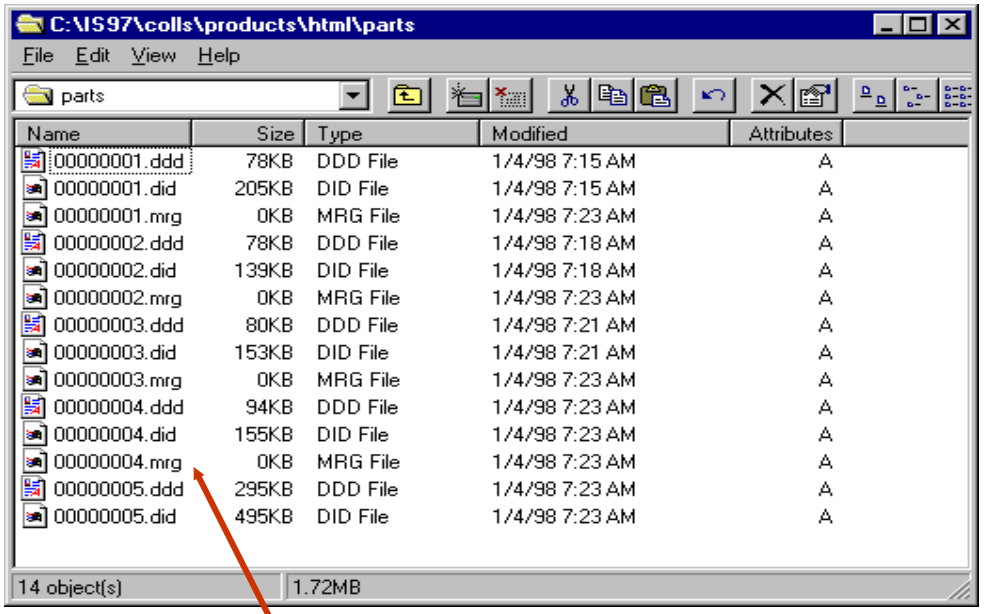
- The vspider utility recognizes these MIME types when Universal filtering or matching for -include and -exclude options

Format	MIME Type	Filter
HTML	text/html	zone filter
SGML	text/sgml	zone filter
News	message/news	zone filter
Email	message/rfc822	zone filter
ASCII	text/*	(no filter)
RTF	application/rtf	Keyview
PDF	application/pdf	PDF filter
GIF	image/gif	(no filter)
MS Word	application/msword	Keyview
MS Excel	application/x-ms-excel	Keyview
MS Powerpoint	application/x-ms-powerpoint	Keyview
WordPerfect	application/wordperfect5.1	Keyview
	application/x-corel-wordperfect	
MS Works	application/x-ms-works	Keyview
MS Project	application/x-ms-project	Keyview
Lotus AMI Pro	application/x-lotus-amipro	Keyview
Lotus 1-2-3	application/x-lotus-123	Keyview

- Other MIME types can be filtered by invoking the appropriate filter in style.uni

Maintaining Your Collections

- As documents are added to a collection, multiple instances of the .ddd and .did are created
  - Multiple indexing tasks are submitted
  - Large numbers of files or long indexing processes create “chunks” at a time
- Merging is the process of taking several small partitions and creating a larger single file that is faster to read
- Housekeeping is the process of cleaning up what is no longer needed



indicates this partition has been merged and will be deleted

Collection Performance

- Collections begin merging when the number of partitions exceeds 4 (4 ➔ 1)
- Housekeeping occurs when
  - An indexing process has been running for more than 5 minutes
  - A scheduled service is run by the Collection Servicer utility
  - An mkvdk instruction to service the collection has been issued
    - Housekeep deletes unneeded files
    - Optimize enables background optimization
    - Data prep handles any outstanding work including optimization and housekeeping

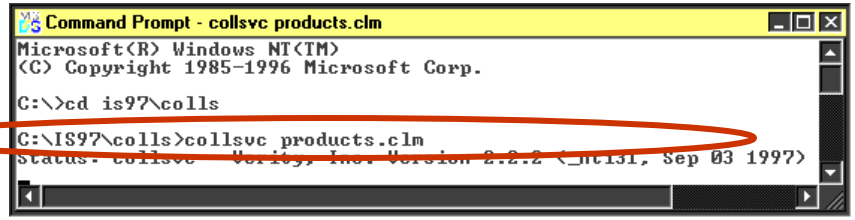
```
mkvdk -collection c:\is97\colls\products.clm -servlev housekeep

mkvdk -collection c:\is97\colls\products.clm -servlev optimize

mkvdk -collection c:\is97\colls\products.clm -servlev dataprep
```

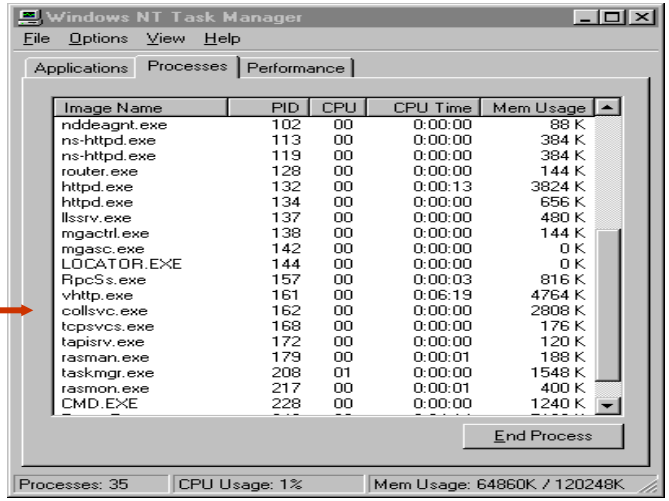
The Collection Servicer Utility

- Command-line utility for
  - Performing maintenance on collections
  - Offloading processing from vspider
- Performs any combination of these tasks
  - Insertion of new documents (indexing)
  - Collection optimization (merging)
  - Periodic deletion of document references (housekeeping)
  - Periodic recovery of no-longer-needed disk space (squeezing)
- Once a collsvc process is executed from the command-line, it continues to run
  - Scheduling parameters for maintenance actions
  - Uses resources as needed
  - On UNIX set a shared library environment variable

Using collsvc

- Issue the command to service your collection
  - Add document deletion information (delete query and delete schedule)
  - Add partition squeeze information to recapture space after a delete (squeeze schedule)
  - Virtual collections don't allow deletes or squeezes so you will have to set a collsvc process for each of the real collections






```
Sun 04-Jan-98 09:48:14 AM - 162 - Status: Opening collection
vdksvc.exe (Sun Jan 04 09:48:15 1998): Status  E1-0103
(Vdksvc: Coll): Commencing servicing collectionvdksvc.exe
(Sun Jan 04 09:48:16 1998): Status  E1-0007 (Vdksvc): vdksvc
shutting down98 09:48:15 AM - 220 - Status: vdksvc shutting
down
```

collsvc.log

The browse Utility

- The browse utility provides information regarding the fields and other document attributes in a single partition
- Output can be viewed on screen or redirected to a file

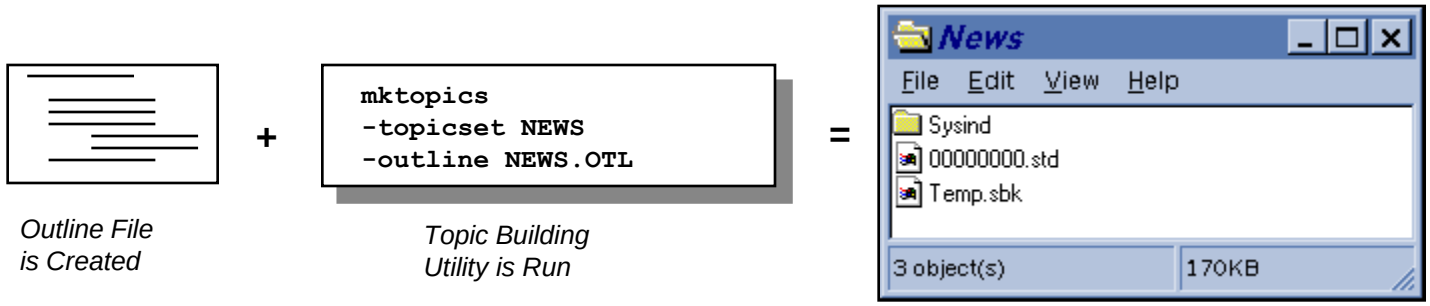
```
browse 00000001.ddd > myfile.txt
```

- It is helpful to understand menu options, as you have to simulate these to redirect output. Your workbook takes you through this process.

```
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
```

The mktopics Utility

- You create topic definitions for a topic set in a topic outline file
- The mktopics utility creates a compiled topic set from an outline file
  - Syntax is checked. If an error is detected the utility provides you with information to correct the error.
  - If no errors are found, the mktopics build the topic set in the directory specified
  - The maximum number of topics is 500,000 and the maximum number of topic links is 800,000
  - Create a new topicset: `mktopics -topicset news -outline news.otl`



Indexing Collections with Topics

- There are two ways to work with a compiled topic set:
  - Add it to the Information Server to help users write better queries
  - Create a special topic index for your collection to speed retrievals (33x faster)
- Enter the path to your topicset in the “common” section of the inetsrch.ini file
- Topics will be substituted for the words entered by users. You can also use these topics on your search forms.

```
Common
TopicSet=c:\is97\topics\news
```

inetsrch.ini

searches expand automatically against the word index

sports = sports, baseball, soccer, football

```
vspider -collection c:\is97\colls\mycoll.clm
-style c:\verity\s97is\locale\english\styles
-start http://www.yoursite.com/
-topicset c:\is97\topics\news
```

pre-built answer sets instantly presented

sports: 7,52,133,619,705

The didump Utility

- The didump utility provides information regarding the occurrences of words in a single partition
- Output can be viewed on screen or redirected to a file

```
didump 00000000.did > myfile.txt
```

- Details can be obtained on all words or a single word

```
didump [-v] [-pattern pattern] partition.did

-v          produce verbose listing per word
-pattern    word or regular expression to display
```

Sample Output

```
Word    Size    Doc    Occurrences
A       371     42     81
a       4382    66     1394
```

RCVDK

- RCVDK is a great little command line retrieval client that will allow you to quickly test your collections.

```
Available commands:

search              s       Search documents
results             r       Display search results
clusters            c       Display clustered search results
view                v       View document
summarize           z       Summarize documents
attach              a       Attach to one or more collections
detach              d       Detach from one or more collections
quit                q       Leave application
about                       Display TDK "about" info
help                ?       Display help  (use help help for details)
expert              x       Toggle expert mode on/off
```

```
rcvdk collection-name
rc v2.2
Attaching to collection:  collection-name
Successfully attached to 1 collection
Type 'help' for a list of commands
RC>
```

Practice Lab

- Please complete the exercises for Practice Lab #2 in your student workbook
  - These exercises will give you a chance to gain an understanding of
    - Basic structure and contents of a collection
    - Collection management features
    - Collection building with vspider
    - Using mkvdk to optimize collections
    - Adding topics to your application
    - Using collection information utilities

Enabling Search at the Server

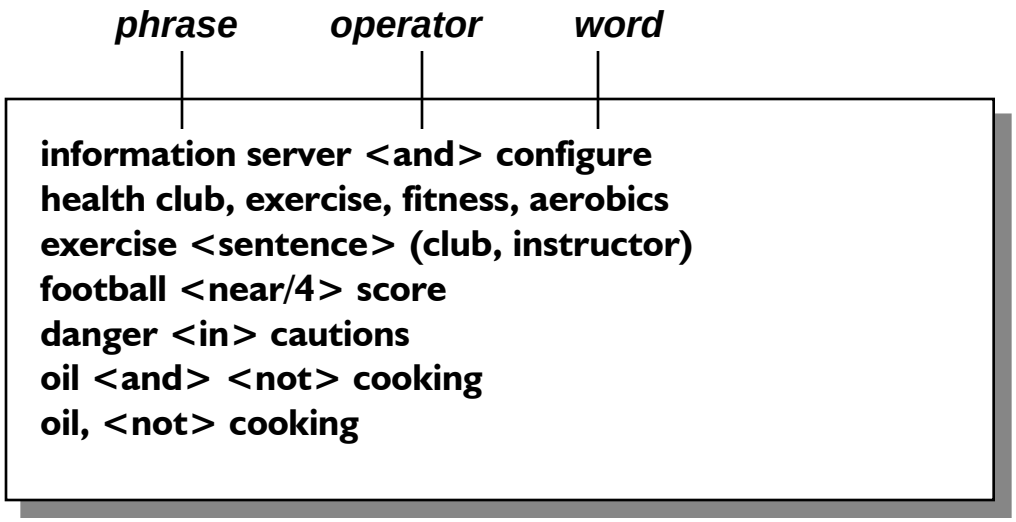
- Exploring the Query Language
- Search Tips Online Guide
- Search Form Alternatives
- Practice Lab

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?
  - 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 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



Query Components

- Queries can include any of these components
  - **WORDS** (in your documents and stored in the word index)
  - **TOPICS** (pre-defined groups of words and information about how the words relate to each other)
  - **FIELD VALUES** (attributes about the documents captured as they are added to the collection and stored in the field index)
  - **OPERATORS** (provided by Verity to help you specify words should be searched for and how results should be evaluated)

Syntax Alternatives

- When you use **simple syntax** (the default), the query is interpreted with a broad focus
  - Searches case-insensitive
  - Applies STEM operator to search words, selecting variant endings
  - Applies the MANY modifier for search words to score documents higher based on word density
  - Automatically interprets words that are topic names as topics
  - Activates the ACCRUE operator at the parent level to specify selection of any of the words entered, but higher scoring of the document for 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@

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
- **Zone Operator** is used with HTML zones
- **Boolean Operators** are used with topics and words to retrieve the elements you describe without operator precedence conflicts
- **Natural Language Operators** use natural language syntax to a search

more...

Evidence Operators

- **Evidence Operators** search for words and can expand into a list of related search words, depending on the operator selected

Operator	Shortcut	Rule
<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)
<thesaurus> film		must search for all synonyms listed in the embedded thesaurus for this word
<wildcard> tech*		must match the character string entered with selected variables: *fil** *substitutes any characters for ** *fil?* *substitutes single letter for ?*

Proximity Operators

- **Proximity Operators** are used with groups of words to define how closely they are related to each other. Proximity refinement often improves query results dramatically.

Operator	Shortcut	Rule
<phrase> nice job	nice job	must locate words 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. Reflects proximity by score, with closest achieving highest score)
football <near/5>score		must locate words within the number of words specified by /n
danger <in> title		locates documents containing values in specific regions (identified during indexing by the zone filter)

Concept Operators

- **Concept Operators** combine the meaning of search words to identify a concept in a document
- **Boolean Operators** are used with topics and words to retrieve the elements you describe without operator precedence conflicts (hit or miss)

Operator	Shortcut	Rule
<accrue>	,	matching documents must contain at least one of the words entered but the more unique words, the better.
<and>		matching documents must contain all of the words
<or>		matching documents must contain at least one of the words
……		
<any>		same as <or> but without weighting
<all>		same as <and> but without weighting

Relational Operators

- **Relational Operators** are used with fields

Operator	Shortcut	Rule
<contains>		the string must be found within the field
<starts>		the field must start with this value
<ends>		the field must end with this value
<matches>		the field must contain a matching string
greater than	>	the numeric field value must be greater than this number
less than	<	the numeric field value must be less than this number
equals	=	the numeric field value must be equal to this number

Natural Language Operators

- **Natural Language Operators** provide “fuzzy searching” capabilities

Operator	Shortcut	Rule
<FREETEXT>		locates documents with content that matches the natural meaning of the words given
<LIKE>		locates documents using a query-by-example parser

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

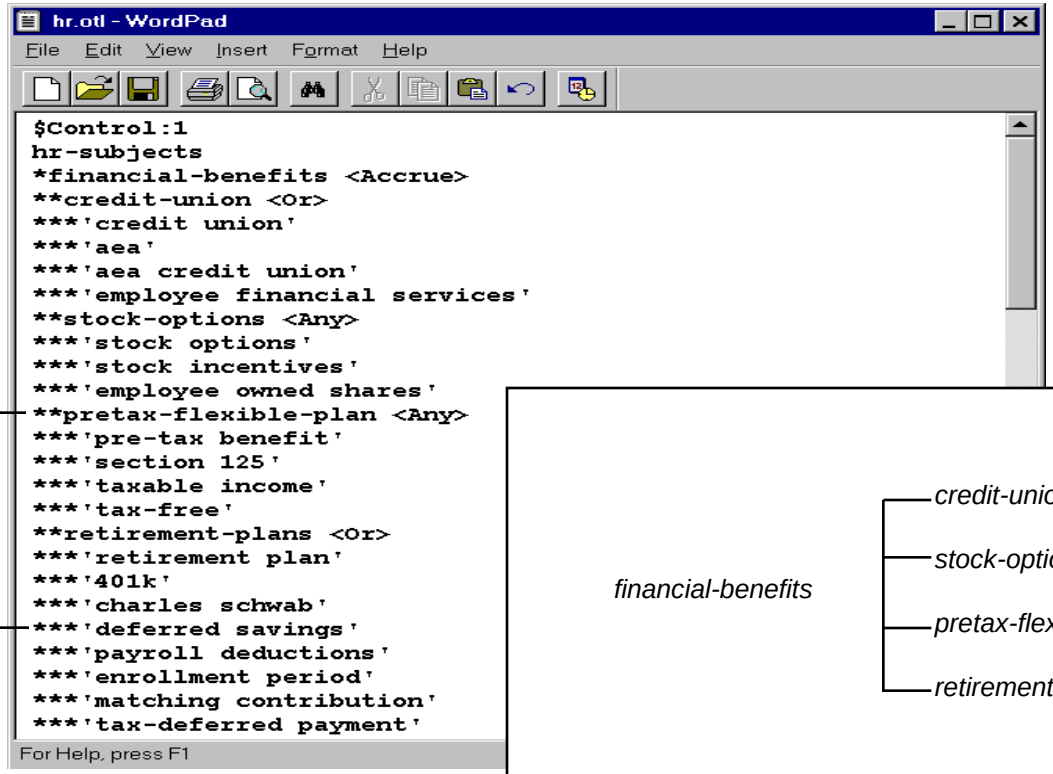
The Importance of Topics

- 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
  - Searching is so much more effective as even novices benefit from expert knowledge captured in the topics
- Topics are cost effective, 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.

Creating an OTL File

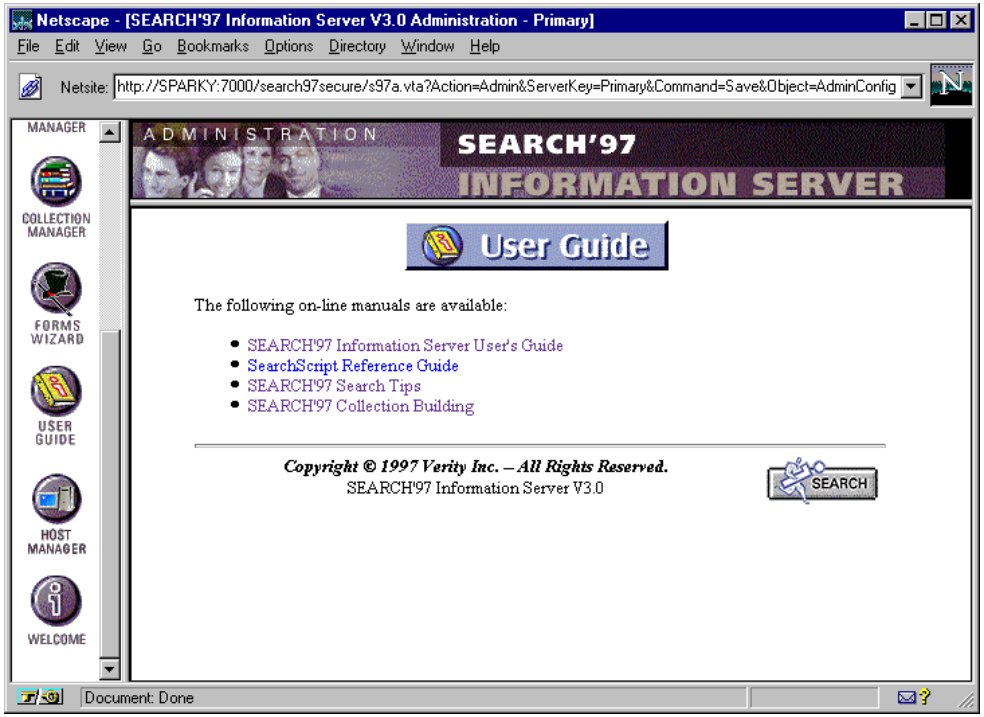
- You can create a topic outline file
  - Using any text editor
  - Using the Topic Editor

A new Java Topic Editor is under development



Search Tips Online Guide

- This online guide provides users with advice on how to conduct organized searches for specific information
  - Operators and modifiers are introduced through a variety of search tasks which allow broadening and narrowing of searches
  - Provides practice on excluding documents



Search Form Alternatives

- It is very important to match the search form functionality to the anticipated experience level of the the user
- Add “intuition” by providing more information for those who are interested
  - Search Tips
  - Information about what is contained in document collections
    - Pre-defined queries or topics
    - Options they can set on the search form
      - How sorting works
      - What threshold is
      - Setting maximum documents
      - The difference between types of queries
    - Options they can set in terms of the results that they see
      - Setting levels of detail
      - Using advanced features of clustering and summarization

Practice Lab

- Please complete the exercises for Practice Lab #3 in your student workbook
  - These exercises will give you a chance to gain an understanding of
    - Writing queries
    - Query language basics
    - Online search training provided by Verity
    - Features of search forms

Building Your Search Application

- Exploring SEARCHScript Features
  - Using SEARCHScript
  - Customizing Form Wizard Templates
  - Creating Custom Search Forms and Result Templates
- Practice Lab

Using SEARCHScript

- SEARCHScript is Verity's proprietary scripting language
  - Designed specifically for search-related functionality
  - Allows for alternatives in processing based on action types
- SEARCHScript is used to:
  - Design templates for use in Verity search applications
  - Design HTML web page forms and select templates for
    - Processing search requests
    - Formatting search results
    - Viewing documents
  - Override configuration options and parameters defined in the Information Server Configuration File: inetsrch.ini

SEARCHScript Syntax

- The SEARCHScript Language is typically embedded in a pair of *<% control statement %>* markers

```
<% IF Document.Title = " " Then %>
```

- SEARCHScript Language can also be placed outside of these control markers using substitution

```
Your Query $$QueryText matched $$DocsFound
```

- Text inside of control markers is parsed and executed while text outside of the control markers is simply output
- Specify the operation you want to perform in structured statements
  - Can be part of a URL
  - Can be part of an HTML <FORM> tag

```
http://sparky:7000:search97secure/s97a.vta?ACTION=SEARCH&QueryText="sports"

<FORM METHOD="POST" ACTION="$$Web.Scriptname">
<INPUT TYPE="hidden"  NAME="Action"  VALUE="Search">
```

Exploring SEARCHScript

- SEARCHScript is typically defined in a separate, template file with an .HTS extension
  - Substitutions are resolved and dynamic results are created
  - SEARCHScript is typically written in name-value pairs and can include conditional and iterative statements
    - If -- Then -- Else -- EndIf
    - For -- EndFor
    - ForEach -- EndFor

```
<% -- Print a list of collections --%>
<% Foreach coll in Result.Collection %>
$$Coll
<% Endfor %>

<% -- Figure out pages for non-PDF documents --%>
<% if doc.mimeType=="pdf" then %> $$doc.Pages
<% else %>
<% Val(doc.Size)/2048 %>
<% endif %>
```

SEARCHScript Actions

◆ SEARCH	Specifies query or search operations and results list formatting
◆ FILTERSEARCH	Specifies query or search operations and results list formatting and allows you to specify a filtering template to perform query pre-processing
◆ ADMIN	Performs configuration, indexing, or other administration tasks
◆ FORMGEN	Generates search forms automatically, based on configuration parameters
◆ DOCVIEW	Displays the contents of a document (with highlights) when selected from the results list

SEARCHScript Objects

- There are dozens of object properties and methods that use SEARCHScript actions. These properties are organized into classes.

Object	Property Description
Cluster	Information about documents included in clusters
Collection	Information on configured collections
Config	Default configuration file settings (applicable to all actions)
Document	Information about a specific retrieved document
DocView	Information useful for creating optional, customized views of documents
Request	Name-value pair information for a requested action
Result	Information on current query and documents retrieved in the search
Web	Information on the server and client software (applicable to all actions)

```
<% Request.QueryText = " " %>
<% Foreach coll in Result.Collection %>
<% Foreach coll in Config.Collections %>
<% Print (Web.ServerURL + Web.Scriptname) %>
<% Print (Web.Cookie) %>
Indexed into:  $$doc.CollectionName
Found at: $$doc.ParentID

 <% If exists(Doc.Title) Then %>
      $$Doc.Title
    <% Else %>
      Oops, No Title
    <% EndIf %>
```

Exploring Forms Wizard Templates

- The Forms Wizard uses FORMGEN to create all of the search, result and view forms
- When you “apply” a theme, an URL is created as an HREF that goes into your form
- When the user clicks on the link, the search form is generated automatically


```
HREF=:http://train7:7000/search97cgi/s97_cgi.exe?
Action=FormGen&ServerKey=Primary&Template=smplsrcp.hts">

I want to do a simple search that presents basic results.
```

Reviewing a FORMGEN Search Form

- This template has been customized from the one used by the Forms Wizard
- It shows how the search box is built and what commands cause a search to be performed by the Information Server

```
<!--- Template: $$Template --->
<HTML>
<HEAD><TITLE>Our Simple Query Form</TITLE></HEAD>
<BODY BGCOLOR=#FFFFFF TEXT=#000000 LINK=#000000 VLINK=#666666>
<CENTER><IMG SRC="/samples/cool.gif"><BR><FONT SIZE=5><FONT COLOR="#FF0000"><B>
<I>$$Company</I></B> </FONT></FONT> <HR> This is a Simple Query Form… <HR></CENTER>
<FORM METHOD="POST" ACTION="$$vtopicscriptname">
<INPUT TYPE=HIDDEN NAME="Action" VALUE="Search">
<INPUT TYPE=HIDDEN NAME="ServerKey" VALUE="$$ServerKey">
<INPUT TYPE=HIDDEN NAME="ResultTemplate" VALUE="smplrslp.hts">
<INPUT TYPE=HIDDEN NAME="Theme" VALUE="$$Theme">
<INPUT TYPE=HIDDEN NAME="Company" VALUE="$$Company">
<CENTER> <TABLE BORDER=0> <TR>
<TD COLSPAN=2><b>Enter words or phrases, separated by commas:</b></TD>
</TR><TR><TD COLSPAN=2><INPUT NAME="queryText" size=55 VALUE=""></TD>
</TR><TR><INPUT TYPE="image" SRC="/search97img/search.gif" NAME="SEARCH-97" BORDER=0>
</TD><TD ALIGN=RIGHT><INPUT TYPE="reset" VALUE=" Clear ">
</TD></TR></TABLE></CENTER></FORM>
</TD></TR></TABLE></CENTER></BODY></HTML>
```

Creating Dynamic Results

- All results processing is done on the fly
- You can add important functionality to your results list beyond the collection data


```
<INPUT TYPE=HIDDEN NAME="ResultTemplate" VALUE="smplrslp.hts">
```

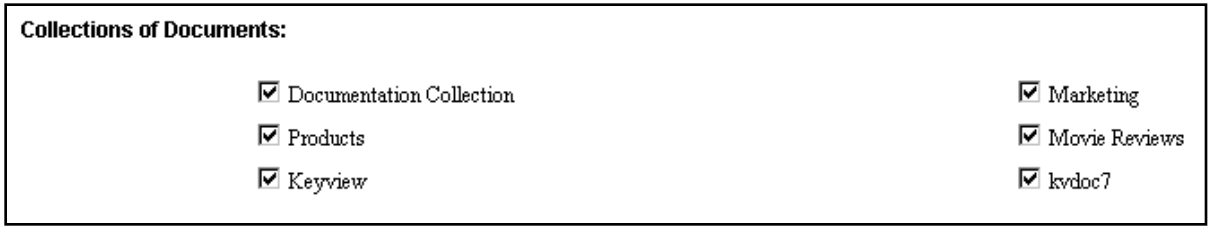
Reviewing a FORMGEN Results Form

- This template is the same as the one used in the Forms Wizard (but it is not a complete file)

```
<!--- Template: $$Template --->
<HTML><HEAD><TITLE>Information Server Simple Results List</TITLE></HEAD>
<BODY> <CENTER> <TABLE BORDER=1 WIDTH=520> <TR> <TD> <CENTER>
<IMG SRC="/search97img/rbanner.gif"> <FONT SIZE=5><B>$$Company</B></FONT>
<HR>Simple Results List   <HR> </CENTER>

<% If Count(Result.PageUrls) > 1 Then %> <CENTER>
<B>Results List Navigation:</B>
<% if PrevPageURL then %>
<A HREF="$$(PRevPageURL)&ServerKey=$$ServerKey&AdminImagePath=
<% printUrlEsc(AdminImagePath) %> &Theme=$$Theme&Company=<%PrintUrlEsc(Company)%>">
[Prev]</A> <% Endif %> <HR> <% EndIf %> <TABLE>

<TR><TH>Score</TH><TH ALIGN="left">Title</TH></TR>
<% Foreach Doc in Result.Documents %>
<TR><TD>$$Doc.Score</TD>  <TD>
<A HREF="<%$$Doc.URL_HTML%>hlnavigate=off&ViewTemplate=<% PrintUrlEsc("smplviep.hts") %>
&ServerKey=$$ServerKey&AdminImagePath=<% PrintUrlEsc(AdminImagePath) %>
&Theme=$$Theme&Company=<%PrintUrlEsc(Company)%>">
<% If exists(Doc.Title) Then %> $$Doc.Title
<% Else %> No Title
<% EndIf %></A>
</TD>  </TR> <% Endfor %> </TABLE>
...
```

Dynamic List of Collections

- Build a list of collections using all enabled collections (review basic0.hts)




```
<DD><H4><B>Collections of Documents:</BOLD></H4>
<% x = 0 %> <% coll_count = Count( Collections ) %> <% if coll_count > 0 %>
<CENTER> <TABLE BORDER=0 WIDTH=100%>

<% foreach coll in Collections %>
<%-- trim out collections not enabled or not in the default list --%>
<% if exists(coll.state) AND 0 = StrComp( coll.state,"Enabled" ) %>
<% if exists( coll.defaultList ) AND 0 = StrComp( coll.defaultList, "True" ) %>
 <% x = x + 1 %><% if x % 2 %> <TR> <TD WIDTH=25%></TD> <TD>
<INPUT TYPE="checkbox" NAME="collection" VALUE="$$coll.Alias" CHECKED> $$coll.Name</TD>
<% else %> <TD>
<INPUT TYPE="checkbox" NAME="collection" VALUE="$$coll.Alias" CHECKED>$$coll.Name</TD>
</TR>
<% endif %><% endif %><% endif %><% endfor %>
<% if x % 2 %> </TR> <% endif %> </TABLE>
</CENTER> <% else %>
<DD>No collections.
<% endif %>
```

Search Feedback

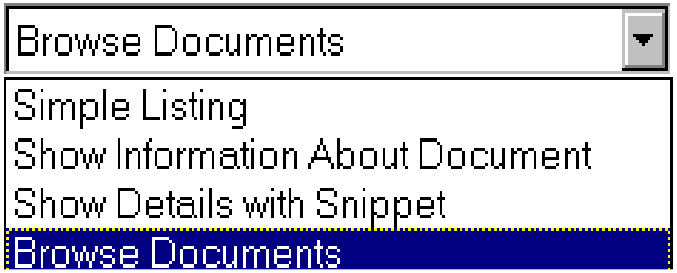
- Provide feedback to the user on results being returned

Your query matched 12 out of 1918 documents in our library. Documents 1 - 10 are shown below, with best matches first

```
Your query matched <FONT COLOR=FF0001> $$DocsFound out of
$$DocsSearched </FONT> documents in our library. Documents
$$DocsStart - $$DocsEnd are shown below, with best matches
first.
```

Choosing Results Style

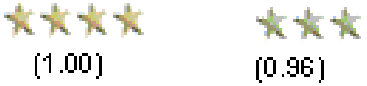
- Give users control over how results will be output




```
<BR><DD><H4><B>Choose How to Display Results:</H4>
<DD><SELECT NAME="ResultTemplate">
<OPTION SELECTED VALUE= "rstndard.hts">Simple Listing
<OPTION VALUE="rpower.hts">Show Information About Document
<OPTION VALUE= "rbasic.hts">Show Details with Snippet
<OPTION VALUE= "rbrowse.hts">Browse Documents
```

Images for Relevance Ranking

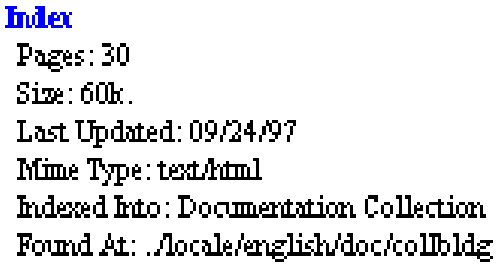
- Represent the document's relevance with an image rather than the score




```
<% foreach doc in result.documents %> <tr>
<td valign=top><right>
<img src="/samples/<% Val( Right( $$doc.Score,2 ) )
          /25%>star.gif"> <br>
<b> <center> <FONT FACE="arial"><FONT SIZE=1>
($$doc.Score)</right></FONT> </FONT></center></td>
```

Get Info

- Display document information directly from the collection

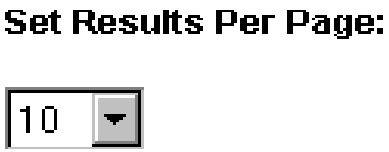



```
<a href="$$doc.URL_HTML"><strong>$$doc.Title</strong></a><br>
Pages: <% Val(doc.Size) / 2048 %><br>
Size: <% Val(doc.Size) / 1024 %>k.<br>
Last Updated:
    <%print(date($$doc._Modified,"$mm/$dd/$yy")) %><br>
Mime Type: $$doc.mime-type<br>
Indexed Into: $$CollectionName<br>
Found At: $$doc._ParentID<br>
```

Setting Results

- Allow the user to select how many results they want per page

```
<DD><H4><B>Set Results Per Page:</H4>
<DD><SELECT NAME="ResultCount">
<OPTION SELECTED VALUE="10" CHECKED>10
<OPTION SELECTED VALUE="20">20
<OPTION SELECTED VALUE="50">50
```



- When viewing documents, jump to the first highlight

```
<a href="$$(doc.URL_HTML)#h1h10">
<strong><font size =-1>
<% If exists(Doc.Title) then %> $$doc.Title>
<% Else %> Title Missing  <% EndIf %>
```

Navigation Links

- Turn on navigation from inside documents

And, check this box ☐ to display navigation links when viewing documents.

```
<TR><TD><A HREF="$$(SearchURL)&ServerKey=$$ServerKey
&AdminImagePath=<% PrintUrlEsc(AdminImagePath) %>
&Theme=$$Theme&Company=<%PrintUrlEsc(Company)%>">
[Go Back To Results] </A></TD>
<% if Exist(PrevDocURL) %><TD>
<A HREF="$$(PrevDocURL)&ServerKey=$$ServerKey
&AdminImagePath=<% PrintUrlEsc(AdminImagePath) %>
&Theme=$$Theme&Company=<%PrintUrlEsc(Company)%>">
[Previous Doc]</A></TD> <% endif %>
<% if Exist(NextDocURL) %>  <TD>
<A HREF="$$NextDocURL&&ServerKey=$$ServerKey
&AdminImagePath=<% PrintUrlEsc(AdminImagePath) %>
&Theme=$$Theme&Company=<%PrintUrlEsc(Company)%>">
[Next Doc] </A></TD> <% endif %> </TR>
```

[Go Back To Results] [Previous Doc] [Next Doc]

Using Cookies

- Create dynamic, personalized searchers using cookies
  - Verity ships a cookies.hts file that has core functionality
  - The cookies.hts file used for class has been greatly enhanced
- Use the Information Server's date functions

This server is in California. It is now Monday, January 05, 1998 at 02:38 am.
In New York it is Monday, January 05, 1998 05:38 am.
In London it is Monday, January 05, 1998 10:38 am.

```
This server is in California. It is now
<% Today = Now() %>
<% Print (Date(Today, "$Day, $Month $DD, $YYYY
at $HH:$MI $am"))%>. <BR>
In New York it is
<% Today = Now() %>
<% NY = Now() + 3*60*60 %>
<% Print (Date(NY, "$Day, $Month $DD, $YYYY
$HH:$MI $am" ))%>.<BR>
In London it is
<% Today = Now() %>
<% LON = Now() + 8*60*60 %>
<% Print (Date(LON, "$Day, $Month $DD,
$YYYY $HH:$MI $am"  ))%>.
```

Customizing Cookies

- Set variable's for data that is not in the server and then use what is captured from input, to customize a form

Natasha's Personal Searcher

```
<H1> <FONT COLOR="#0000FF" FONT SIZE=+3><I><% IsEmpty(Username)?
"In": Username + "</I>'s "%>
<A NAME="search">Personal Searcher</A></FONT> </FONT></H1>
```

- Carry forward information from one form to create another

You have chosen to sort on Score in Asc order. Only 10 documents are to be displayed on each result page, and results will be presented using the default.hts format style.

```
You have chosen to sort on $$SortField in $$SortOrder order. Only
$$ResultCount documents are to be displayed on each result page,
and results will be presented using the $$ResultTemplate format
style.
```

Viewing by Document Type

- Launch to different viewers based on mime type

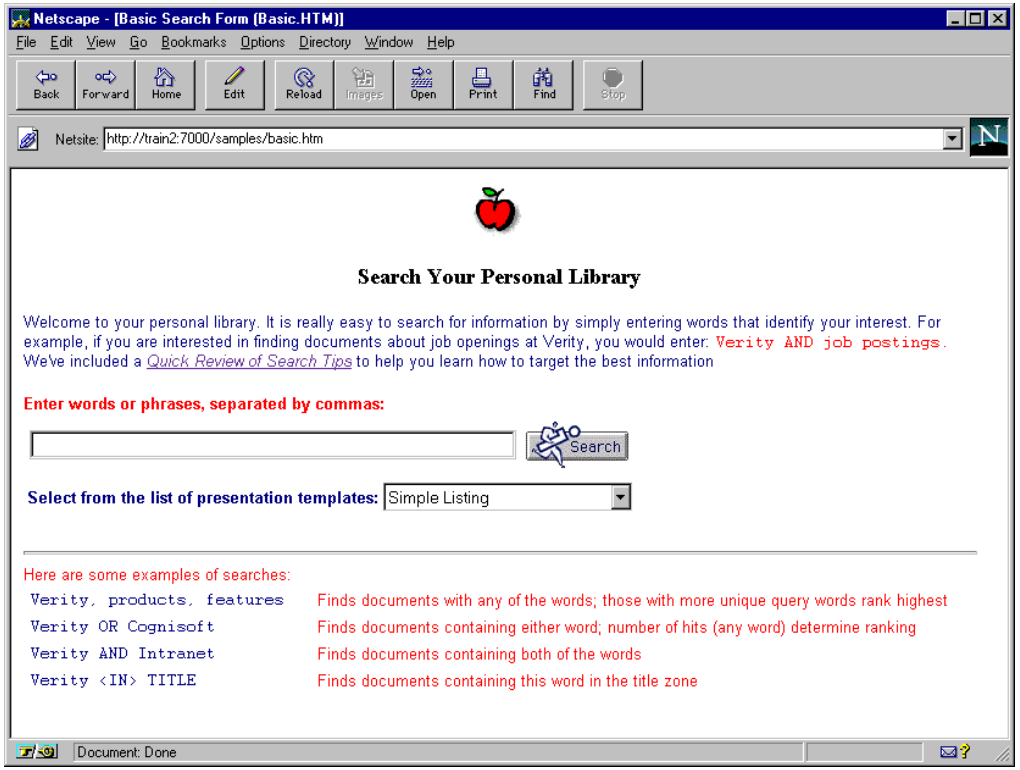
```
<table> <tr>
<td> Score </td>  <td> Title </td>  </tr>
<% foreach doc in result.Documents %>
<tr>
<td>$$doc.Score</td>

<% if doc.MIME-Type = "text/html" %>
          <td> <a href = "$$doc.URL_HTML">$$doc.Title </a> </td>

<% elseif doc.MIME-Type = "application/pdf" %>
          <td> <a href = "$$doc.URL_XML">$$doc.Title </a> </td>

<% else %>
          <td> <a href = "$$doc.URL">$$doc.Title </a> </td>

<% endif %>
</tr>
<% endfor %>
</table>
```

Practice Lab

- Please complete the exercises for Practice Lab #4-A in your student workbook
  - These exercises will give you a chance to create FORMGEN templates that are customized for your site

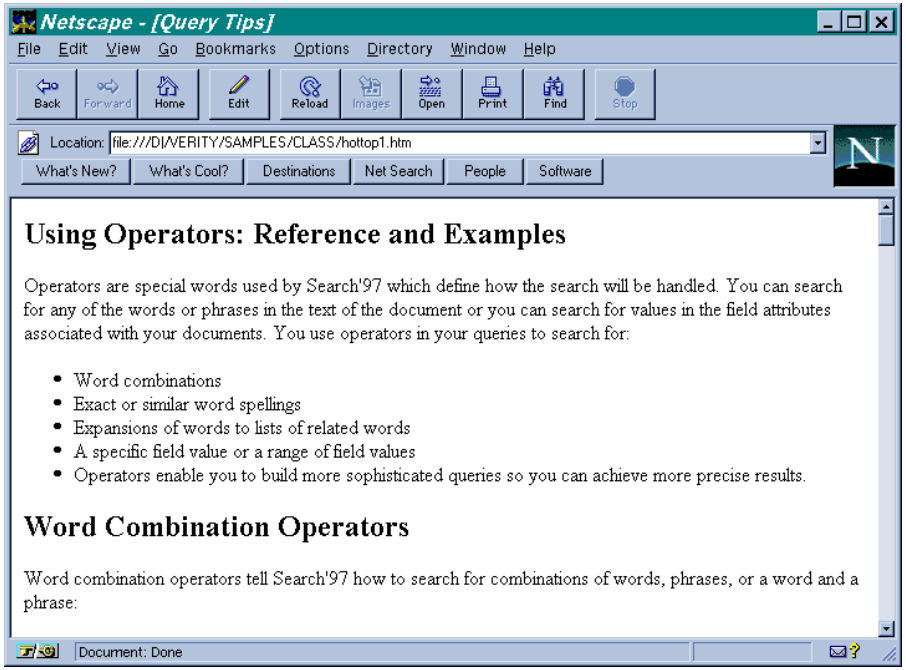
Building Search Forms

- You can copy these training search pages as a starting point, and then customize them to fit your needs
- Because results are still dynamic, you will need to make sure that customized versions reside in the template directory (specified in the inetsrch.ini file)
  - This search form is called **basic.htm**



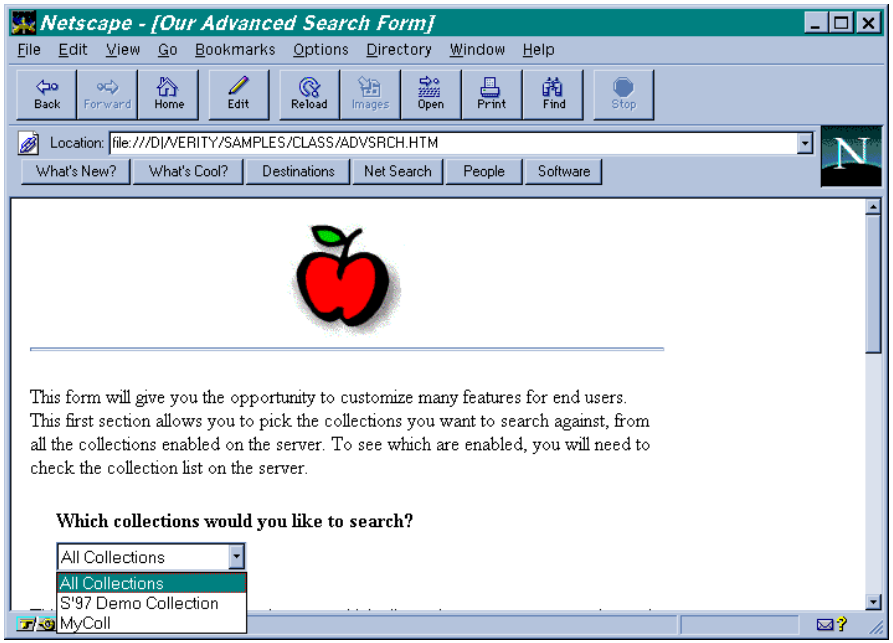
Helping Novices

- It's a good idea to include search tips for users if you expect to have regular "visitors" who have little experience with writing queries
- You are welcome to link to any of the information points on the Verity web site or embed the new Online Search Guide on your pages
  - This document is called **hottop1.htm**



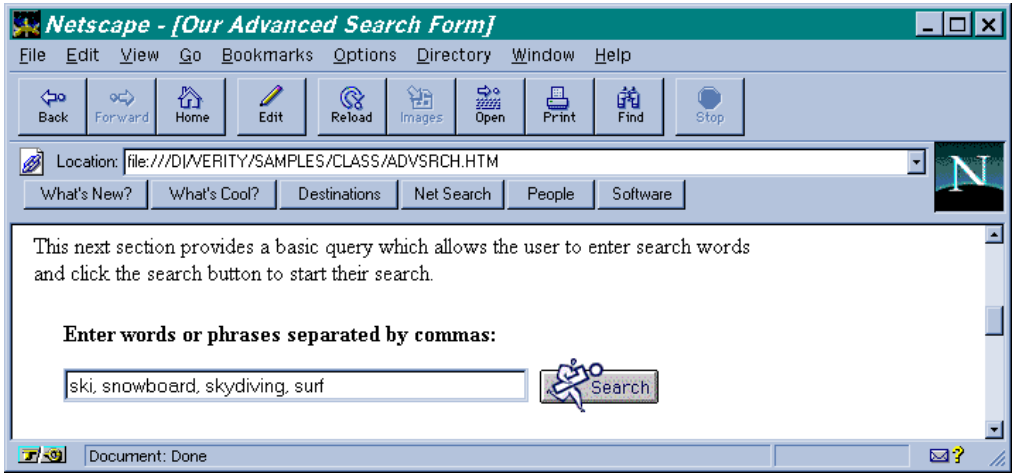
Providing Features

- There are several features you can add to improve searching for users
- One of the most important feature is to allow users to limit their searches to specific collections
  - For many is it more about what they WON'T see than what they WILL see



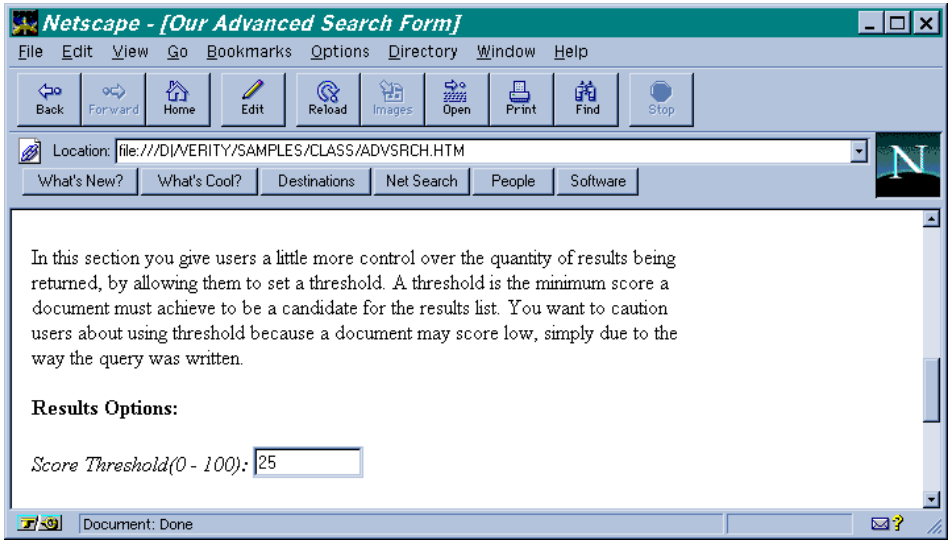
Query Box

- Over the years we have found the best way to describe how to enter a query as shown below
  - Many sites do fun or interesting things to help users understand how to write a meaningful query
  - The Verity web site has many customer demos and you might get some valuable ideas by perusing these sites



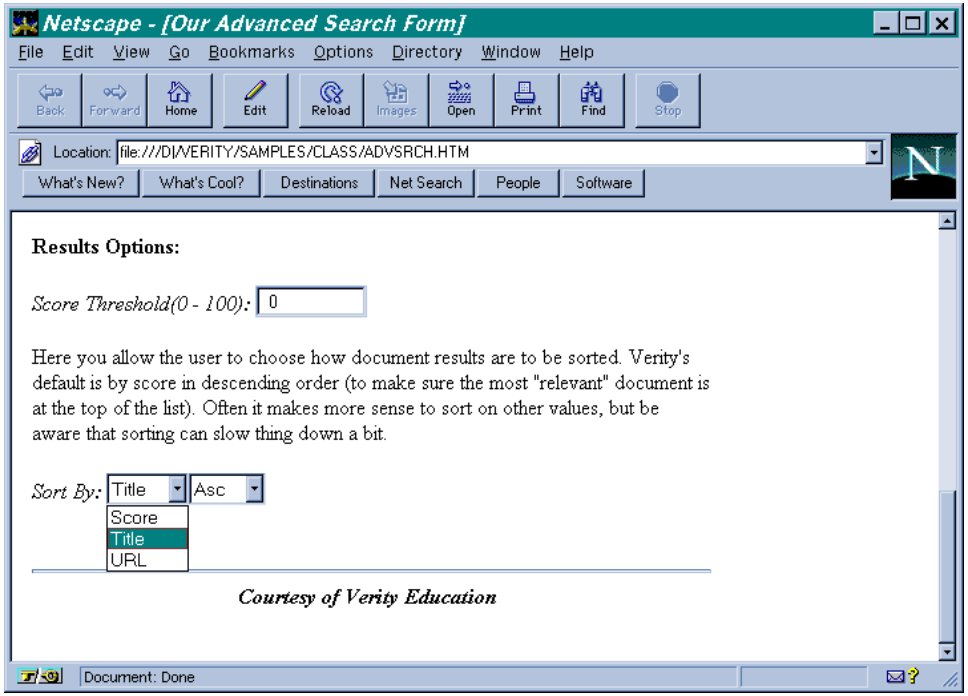
Setting a Threshold

- You can allow your users to limit their results to the “best” of the bunch
  - Threshold is good for avoiding documents that are minimally related to the subject
  - The query you choose can drastically influence the score on any given document, so use threshold with caution!



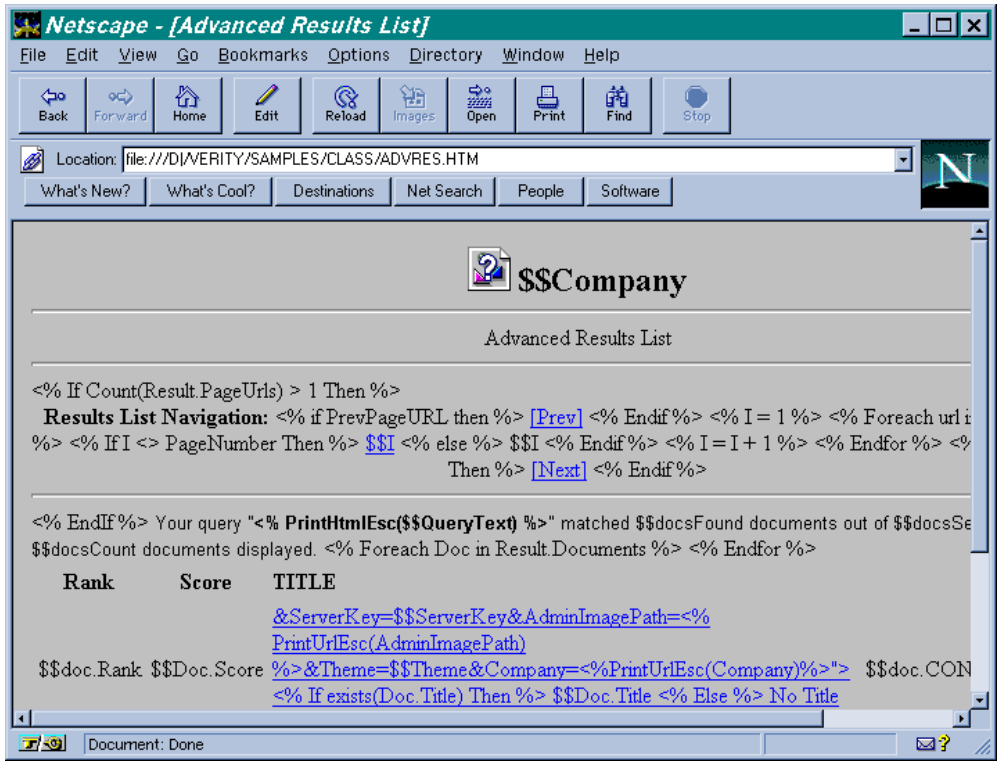
Sorting Results

- Keep in mind that the most important duty for the results list is to help users determine which documents they want to read
  - Users have a short attention span and only want to see the best information
  - Sorting takes a little more time, so based on the number of results, you may want to limit options



Copying the Right Files

- You need to be careful about how you copy files
  - Search forms (HTML) are not dynamic so they can reside anywhere you want
  - Results forms are dynamic and must be placed in the template directory named in your inetsrch.ini file



Exploring the SEARCHScript

Search Your Personal Library

Welcome to your personal library. It is really easy to search for information by simply entering words that identify your interest. For example, if you are interested in finding documents about job openings at Verity, you would enter: Verity AND job postings. We've included a *Quick Review of Search Tips* to help you learn how to target the best information.

```
<HTML>
<HEAD><TITLE>Basic Search Form (Basic.HTM)</TITLE></HEAD>
<BODY bgcolor=ffffff>
<form method="POST" action="/s97is.vts"> <INPUT TYPE="hidden" NAME="Action" VALUE="Search">
<center><img src="smapple.gif" height=40 width=42>
<H3>Search Your Personal Library</H3></center>

<FONT FACE="arial"> <FONT SIZE=-1> <FONT COLOR="#000080">
Welcome to your personal library.  It is really easy to search for information by simply
entering words that identify your interest.  For example, if you are interested in finding
documents about job openings at Verity, you would enter:  <FONT FACE="courier">
<FONT COLOR="#FF0001"> Verity AND job postings. </FONT> </FONT><BR>
We've included a <A HREF="hottop1.htm"> <I>Quick Review of Search Tips</I></A> to help
you learn how to target the best information.  <BR> </FONT></FONT></FONT>
```

Build the Search Box


```
<P>
<b><FONT FACE="arial"> <FONT SIZE=-1> <FONT COLOR="#FF0001">
Enter words or phrases, separated by commas:</b>
<table>
<tr>
    <td><INPUT NAME="queryText" size=55 VALUE=""></td>
    <td><INPUT TYPE="image" SRC="search.gif" NAME="SEARCH-97"
    ALT="Search" BORDER=0></td>
</tr>
</table>
```

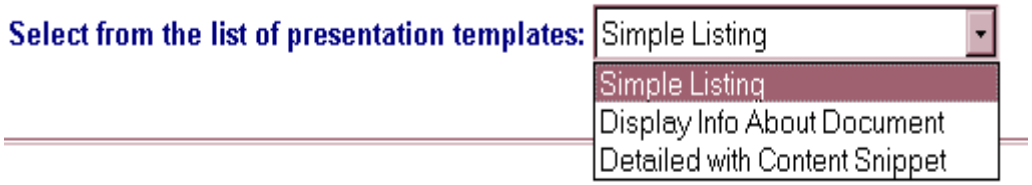
Help Them Along

Here are some examples of searches:

Verity, products, features	Finds documents with any of the words; those with more unique query words rank highest
Verity OR Cognisoft	Finds documents containing either word; number of hits (any word) determine ranking
Verity AND Intranet	Finds documents containing both of the words
Verity <IN> TITLE	Finds documents containing this word in the title zone

```
Here are some examples of searches: </FONT></FONT></FONT>
<table> <tr>

<td><FONT FACE="courier"> <FONT SIZE=-2> <FONT COLOR="#000080">
Verity, products, features </FONT></FONT></FONT> </td> <td> </td><td> </td>
<td><FONT FACE="arial"> <FONT SIZE=-1> <FONT COLOR="#FF0001">

Finds documents with any of the words; those with more unique query words rank
highest </FONT></FONT></FONT> </td> </tr> <tr> <td><FONT FACE="courier">
<FONT SIZE=-2> <FONT COLOR="#000080">

Verity OR Cognisoft </FONT></FONT></FONT> </td> <td> </td><td> </td>
<td><FONT FACE="arial"> <FONT SIZE=-1> <FONT COLOR="#FF0001">

Finds documents containing either word; number of hits (any word) determine ranking
</FONT></FONT></FONT> </td>
```

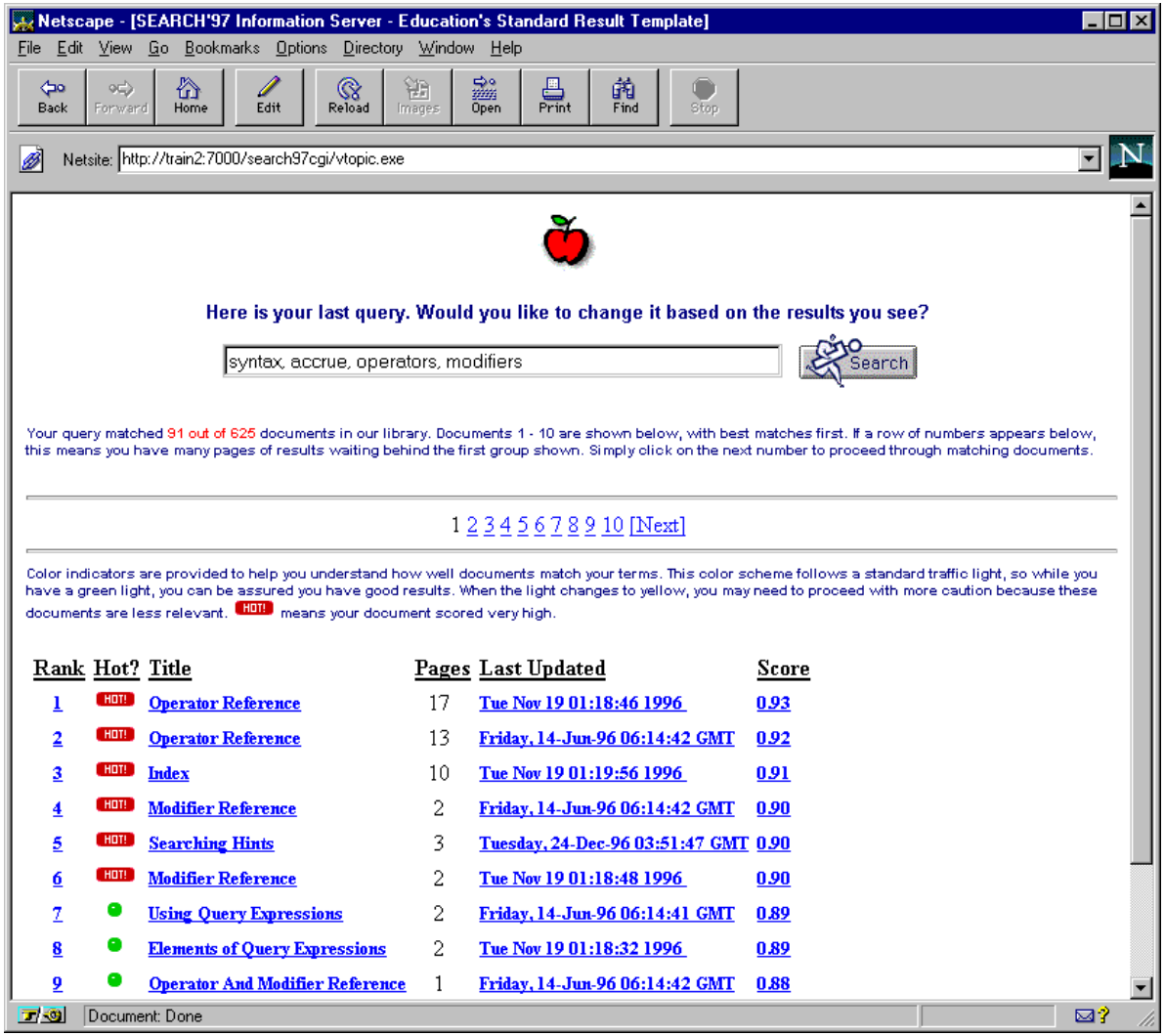
Choosing Result Formatting




```
<table border=0>
<tr>
<td>
</FONT></FONT></FONT>
<FONT FACE="arial"> <FONT SIZE=-1> <FONT COLOR="#000080">

<b>Select from the list of presentation templates:</b></td>
<td>
<select name="ResultTemplate">
<option value="stndard.hts" Selected>Simple Listing
<option value="power.hts">Display Info About Document
<option value="rbasic.hts">Detailed with Content Snippet
</select>
</td>
</tr>
</table>
</FONT></FONT></FONT>
<br>
```

Simple Listing

This template is called **rstndard.hts**



Info About Document

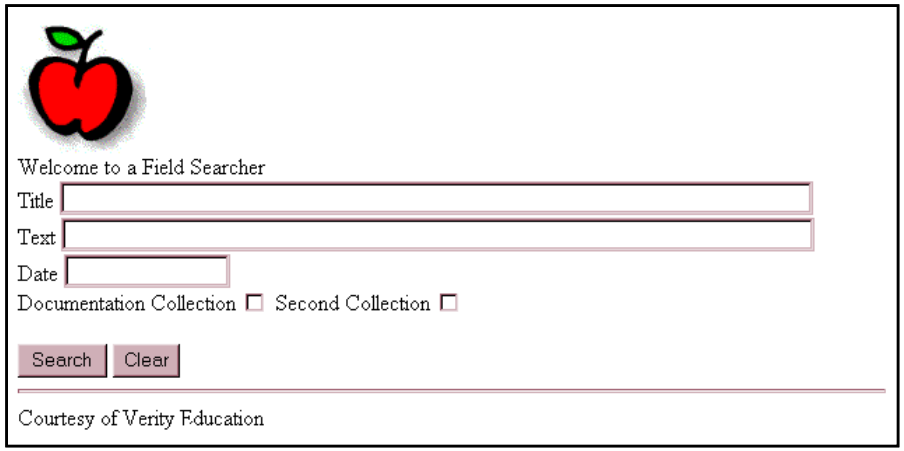
This template is called

rpower.hts

Content Details with Snippet

This template is called **rbasic.hts**

Create a Field Searcher



This search form is called **basic2.htm**

```
<% if IsEmpty(QueryText[1]) then %>
<% QueryText[1] = "" %>
<% else %>
<% QueryText[1] = "title <contains> " +
   QueryText[1] %>
<% endif %>

<% if IsEmpty(QueryText[1])
   || IsEmpty(QueryText[2]) then %>
<% QueryConnect[1] = "" %>
<% else %>
<% QueryConnect[1] = "OR" %>
<% endif %>

<% if IsEmpty(QueryText[3]) then %>
<% QueryConnect[3] = "" %>
<% else %>
<% QueryText[3] = "_index_date " +
   QueryText[3] %>
<% QueryConnect[2] = "AND" %>
<% endif %>
```

Building a Field Searcher

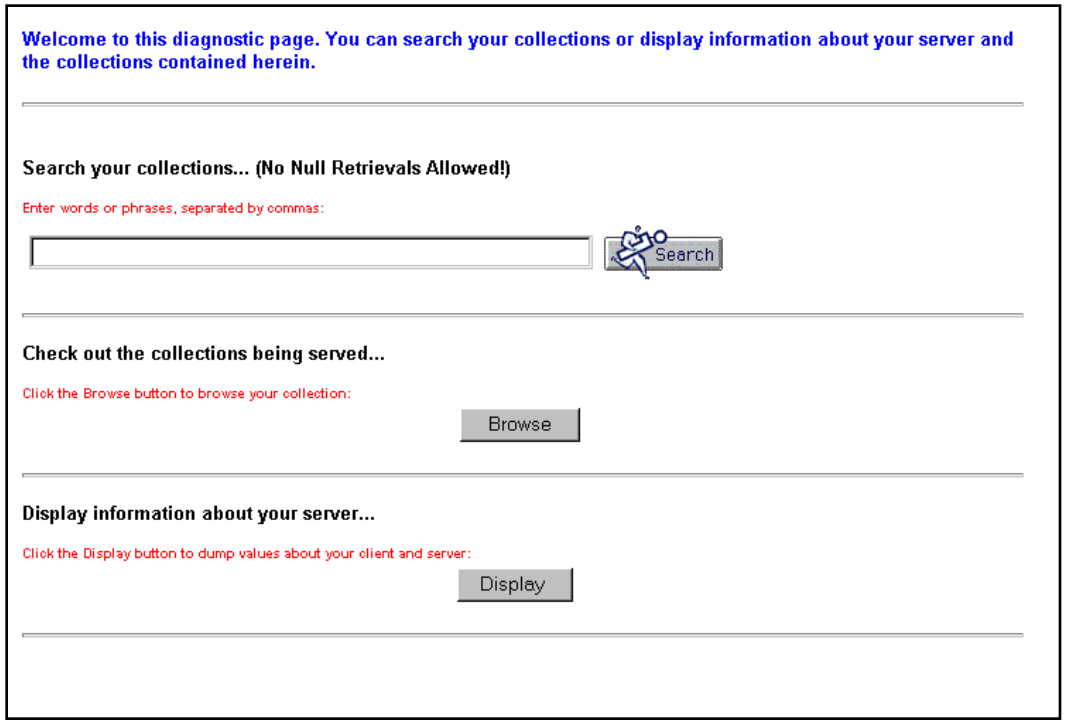
```
<HTML
<HEAD><TITLE>My Basic Search Form</TITLE></HEAD> <BODY bgcolor=ffffff>
<form method="POST"   ACTION= "/s97is.vts">
<INPUT TYPE="hidden"  NAME="SearchPage"     VALUE="mybasic.htm">
<INPUT TYPE="hidden"  NAME="Action"         VALUE="FilterSearch">
<INPUT TYPE="hidden"  NAME="ResultTemplate" VALUE="myresults.hts">
<INPUT TYPE="hidden"  NAME="filter"         VALUE="myfilter.hts">

<img src="apple.gif">
<br>Welcome to a Field Searcher <br>

Title <INPUT TYPE="text"    NAME="QueryText[1]"   SIZE=75>
      <INPUT TYPE="hidden"  NAME="QueryWeight[1]" VALUE="required"> <br>

Text  <INPUT TYPE="text"    NAME="QueryText[2]"   SIZE=75>
      <INPUT TYPE="hidden"  NAME="QueryWeight[2]" VALUE="required"> <br>

Date  <INPUT TYPE="text"    NAME="QueryText[3]"   SIZE=15>
      <INPUT TYPE="hidden"  NAME="QueryWeight[3]" VALUE="required"> <br>

Documentation Collection <INPUT TYPE=checkbox  NAME="collection" VALUE="C1”> <br>
Second Collection  <INPUT TYPE=checkbox  NAME="collection” VALUE="Coll2”> <br> <p>
<INPUT TYPE="submit"  VALUE="Search">
<INPUT TYPE="reset"   VALUE="Clear">
</HTML>
```

Browse Document Contents

- We've created a handy set of forms for Administrators

This search form is called

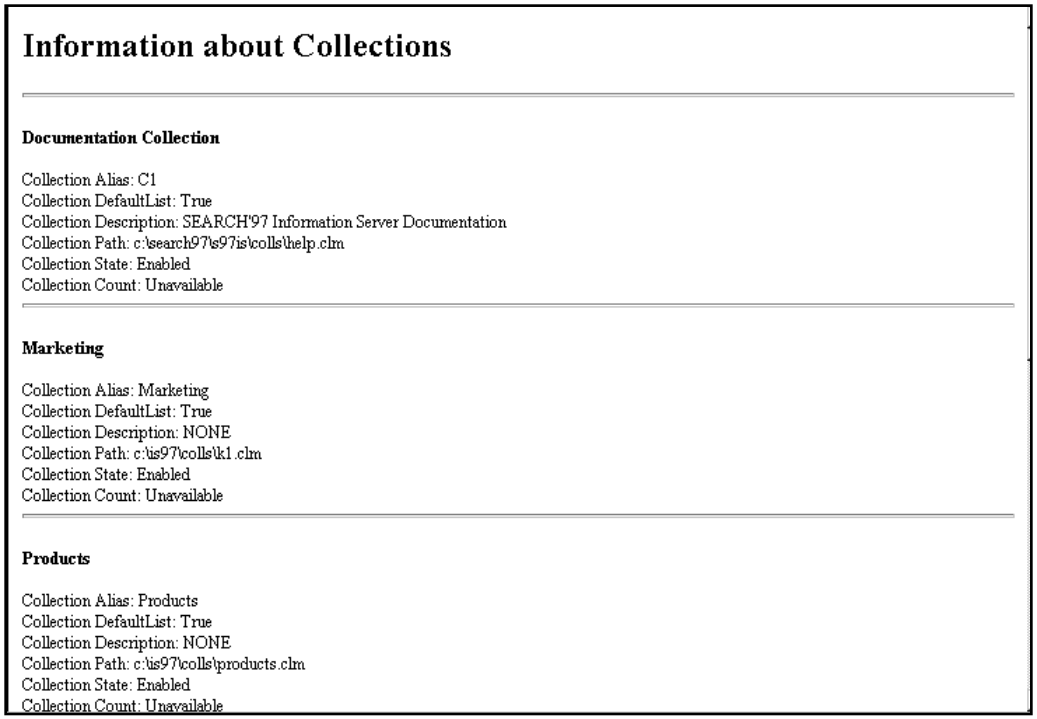
basic1.htm



Administrator Forms

- This one shows “collective” collection data

  This template is called **browse.hts**



Information about Collections

Documentation Collection

Collection Alias: C1
Collection DefaultList: True
Collection Description: SEARCH'97 Information Server Documentation
Collection Path: c:\search97\s97is\colls\help.clm
Collection State: Enabled
Collection Count: Unavailable

Marketing

Collection Alias: Marketing
Collection DefaultList: True
Collection Description: NONE
Collection Path: c:\is97\colls\k1.clm
Collection State: Enabled
Collection Count: Unavailable

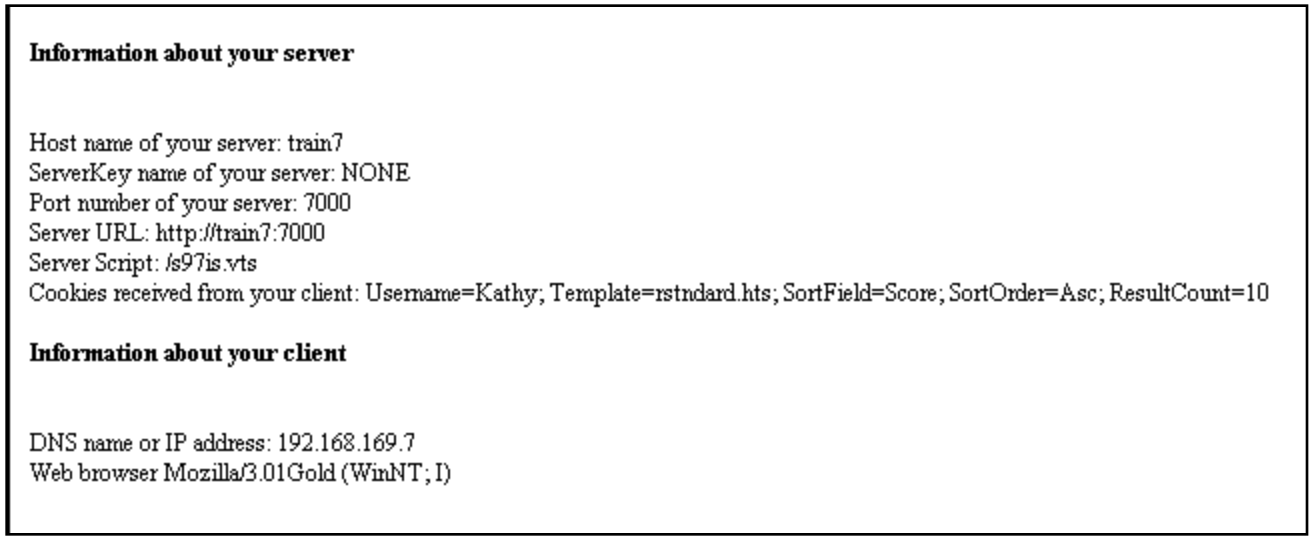
Products

Collection Alias: Products
Collection DefaultList: True
Collection Description: NONE
Collection Path: c:\is97\colls\products.clm
Collection State: Enabled
Collection Count: Unavailable

Administrator Forms

- This one shows information about your server

This template is called **webserver.hts**



Information about your server

Host name of your server: train7
ServerKey name of your server: NONE
Port number of your server: 7000
Server URL: http://train7:7000
Server Script: /s97is.vts
Cookies received from your client: Username=Kathy; Template=rstndard.hts; SortField=Score; SortOrder=Asc; ResultCount=10

Information about your client

DNS name or IP address: 192.168.169.7
Web browser Mozilla/3.01Gold (WinNT; I)

Practice Lab

- Please complete the exercises for Practice Lab #4-B in your student workbook
  - These exercises will give you a chance to create custom search forms and result layout you can use at your site
    - Be sure to download your artwork if possible
    - If you indexed your site, be sure to use your collections for testing

SEARCHScript Workshop

- This workshop will give you the opportunity to build a complete site
  - Collections of your documents
  - Search forms for your users
  - Result templates to support various user types
- Create and document, complete search and result solutions for a novice user and if time permits, for an advanced searcher
- At the end of the lab, present your forms to the class
  - Describe key design features and explain why you chose to use them
  - Share your forms and templates
  - Identify surprises
    - What you found really easy
    - What you found harder to accomplish - and why
    - How you solved particular problems

Your Mission...

- You have two primary tasks to complete
- Each task may involve a series of forms
  - Create a basic search and result “package” that is oriented to novices visiting your server
    - Present basic options that fit the 10/90 rule: the ten percent used - ninety percent of the time
    - Pay attention to aesthetics, take advantage of graphical representations on score, and helpful information on the results list
  - Create an advanced search and result “package” that incorporates the most important features and options for a knowledge worker
    - Provide choices for functionality throughout
    - Learn more about the information contained in the index, to present valuable information to users
- There are no pre-defined answers but you can use all of our samples to get you started (but we suggest you make copies)

Ideas for Your Solutions

- Allow users to select collections and provide a link to a description page that tells them about the collections - for more advanced functionality build a “get info” the users can create dynamically using information stored in the index
- Allow users to specify how they want results presented (from a drop-down list of choices) and provide information they can review about the differences in the choices
- Allow users to select their query mode and to specify how results are to be sorted. Follow through on your results list to remind them what modes they are using.
- Provide results that include rank, score, title, summary and any two other fields from the style.ddd or style.ufl. Show last modified date & size of the document.
- Show only ten results on the page but allow user to select pages of ten results each and let them up this number to 20 if they choose

Ideas for Your Solutions

- Set-up different URL's for results with a click on score going to the highlighted document and a click on rank going to the original document.
- At the bottom of document display, have the Web page display PrevDoc and NextDoc for navigating the results list
- Show information about results including the page number, document count, total documents found and docs searched.
- Build a "Get Info" page about the documents, displaying 10 fields from the index.

Ideas for Your Solutions

- Give the user the ability select from these display options:

URL	Pass the original document to the browser without highlights
URL_HTML	Pass a server-generated HTML document to the browser with highlights
URL_XML	Pass a filtered document to the browser with highlights (PDF or WYSIWYG)

- Add navigation with some of these features:

HTMLHeadPrint	Heading of current document
HTMLBodyPrint	Body Text of current document
NextDocURL	URL of next document on list
PrevDocURL	URL of previous document on list

Ideas for Your Solutions

- Add a date range option, selectable by the user and define a filtersearch to process the values entered
- Build a field-oriented search form
- Use various collection fields on your results list
- Generate search forms on the fly with formgen
- Try a conditional or iterative statement on your advanced search form
- Change the presentation of date information using a function
- Practice with navigation in the viewer (see the Document View Template Properties)
- Add meta-data tags to your documents and build a collection that captures the new fields
- Use math features to calculate the number of pages based on the document size

Practice Lab

- Please complete the exercises for Practice Lab #5 in your student workbook
  - These exercises will give you a chance to
    - Create your own search forms using basic SEARCHScript language
    - Choose how results will be displayed
    - Add features to help users refine searches
    - Have fun and share ideas with other students!