BaysideSearch
Install and Usage Manual

BaysideNetworks.com, Inc.

March 23, 2009

Bayside Networks
Information Technology Service and Support
http://baysidenetworks.com

3655 Nobel Drive Suite 640
San Diego CA 92122-1052

858-654-4080 Phone
877-654-4080 Toll-free

Contents

Chapter 1

Site Information

Company Name:	
BaysideSearch URL:	

Chapter 2

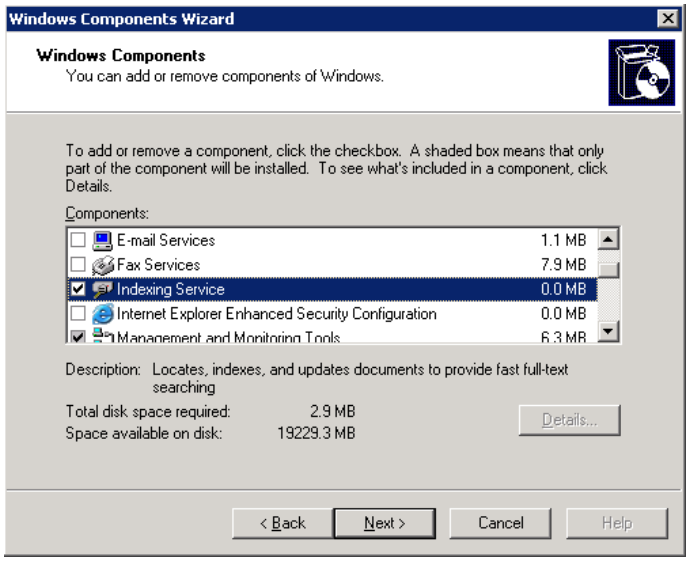
BaysideSearch

2.1 Installing BaysideSearch

2.1.1 Indexing Service

Install

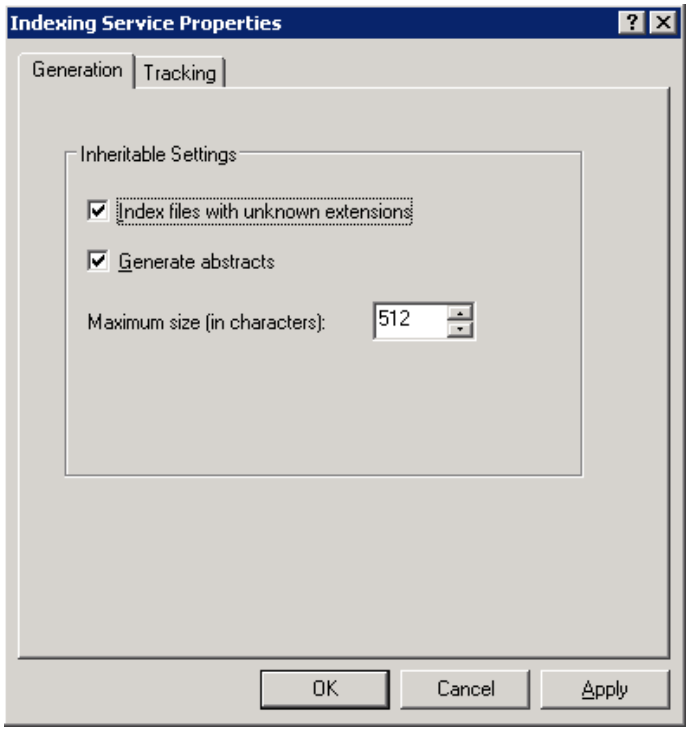
The indexing backed used by BaysideSearch is Windows Indexing Service. You will have to install this component if it is not already installed on your server.



Setup

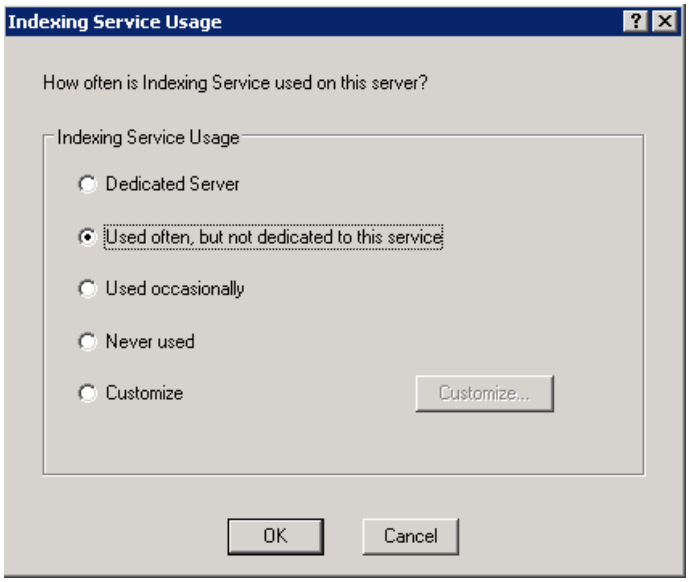
Now that the indexing service is installed you will probably want to change a few of the default settings. Using the Management Console right click on the Indexing Service (found under 'Services and Applications'). You will want to check the 'Generate abstracts' box and bump the 'Maximum size' up a bit to make the abstracts

more readable. Enabling the Abstracts will allow BaysideSearch to also display the abstract in the results page (just like your favorite search engine).



Tune Performance

Depending on your environment you will also want to adjust the Performance settings of the Indexing Service. You should stop the Indexing Service before adjusting the Tuning. To stop the Indexing Service simply right click on the Indexing Service (In the Management Console) and select 'Stop'. Once the Service is Stopped you can enter the 'Tune Performance' by right clicking on the 'Indexing Service' and select 'Tune Performance' under the 'All Tasks' section. To speed up creating of the initial indexing (after hours/weekend - since this will cause the server to slowdown) set the 'Indexing Service Usage' to 'Dedicated Server'. After the initial index has been built return to this screen (stopping the Service first) and change it to 'Used often, but not dedicated to this service' setting.

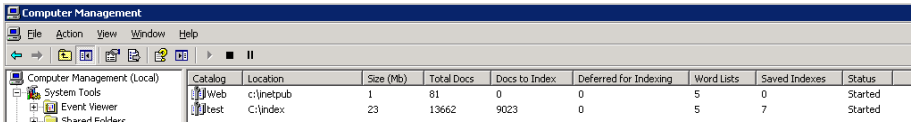


Adding Indexing Catalogs

The BaysideSearch program needs a 'Catalog' name to point to. You will need to create one or more 'Catalogs'. Select 'New' from the menu that appears when you right click on the 'Indexing Service' in the Management Console. You need to give each 'Catalog' a unique name (keep them simple, this name will be needed by the BaysideSearch software). The Location is the directory that will store the 'Catalog Index' files. You will need to add 'directory's' for the Service to index for this Catalog. Right-click on the newly created Catalog and select 'New ¿ Directory'. The 'Path' is the file system path for the file to be indexed. The Alias should be set to the UNC for accessing the Path. You can exclude subdirectories from the index by adding another 'Directory' and selecting 'No' in the 'Include in Index?' selector.

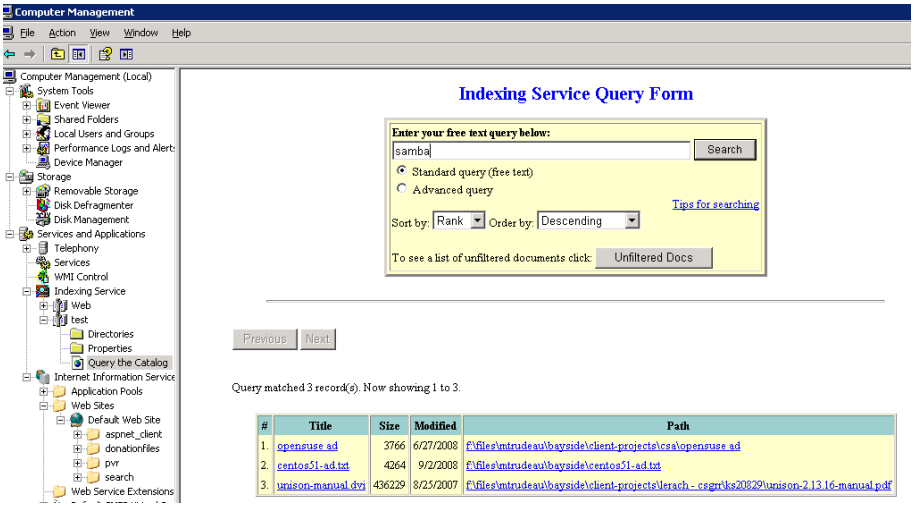
Reviewing Status

When you click on the 'Indexing Service' in the Management Console you can view the current status of the service. When the 'Docs to Index' column is 0 then the Initial Index is complete.



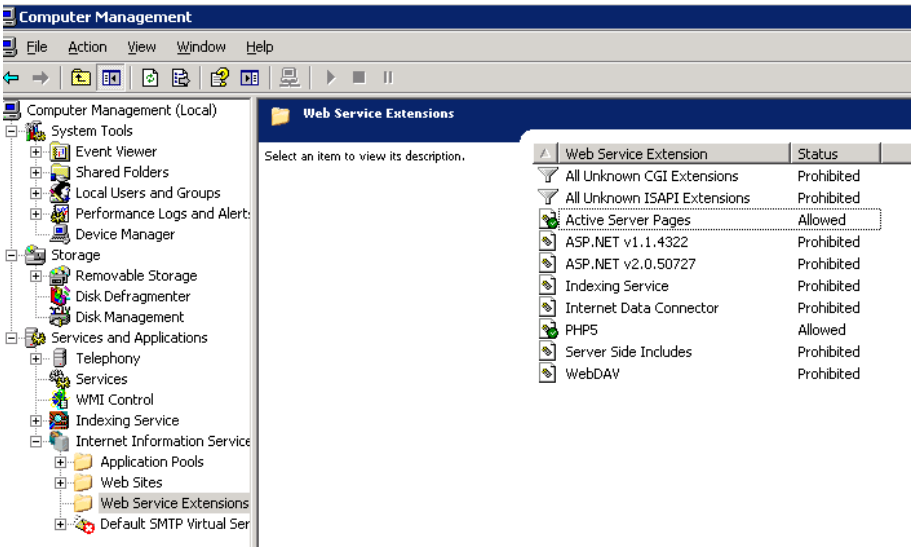
Test Query

when the Initial Index is complete, you can test it by clicking on the 'Query the Catalog' item listed under the created catalog (Management Console).



2.1.2 Active Server Pages

BaysideSearch is written ASP (Active Server pages) so the 'Active Server pages' Extension needs to be set to 'Allowed'. On the IIS server (Management Console) you need to go to the 'Web Service Extensions' folder and select the 'Active Service Pages' entry and set it to 'Allow'.



2.1.3 BaysideSearch

Create a new IIS 'Web Site' or a new sub folder to run the Software from. Copy the two files from the software archive (default.asp and settings.inc) to 'Web Site's' folder.

Company Name

On line 312 in the default.asp file you can modify 'Your Company Name' to the name you wish to display on the web page.

```
response.write "<font face=arial size=5 color=""#000099"">" & _
     "<b>Your Company Name</b></font><br><br>" & nl
```

Adding Catalogs

You will need to modify the settings.inc file to point BaysideSearch to your Index Catalogs. The following section defines how many (base 0) entries that you will have to choose from. The default is 0, one entry. If you wanted to add an additional Catalog you would need to increment the 0 (zero) by the number of entries that you will be adding.

```
dim value(0)
dim label(0)
dim links(0)
```

For each entry you will have one of the following sections. When you add new Catalogs you will need to duplicate this section and increment the 0 (zero) for each entry.

```
value(0) = "test"
label(0) = "Test Catalog"
links(0) = "query://SERVERNAME/test"
```

Here is a settings file example that has multiple Catalogs and a 'All' Selection that allows for searching across multiple catalogs.

```
dim value(2)
dim label(2)
dim links(2)

value(0) = "all"
label(0) = "All Catalogs"
links(0) = "query://SERVERNAME/test,query://SERVERNAME/test2"

value(1) = "test"
label(1) = "Test Catalog"
links(1) = "query://SERVERNAME/test"

value(2) = "test2"
label(2) = "Test Catalog"
links(2) = "query://SERVERNAME/test2"
```

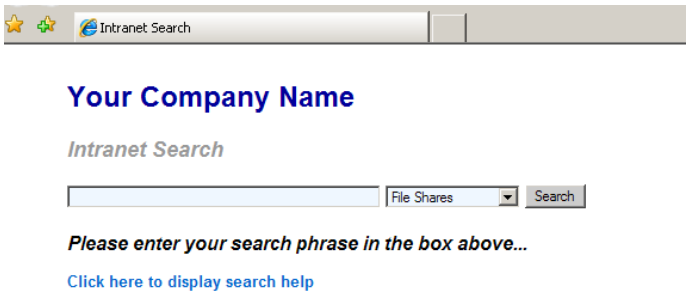
2.2 Using BaysideSearch

2.2.1 Accessing

Using the your web browser go to the URL provided in the Site Information section [[BaysideSearch URL]]. If you are within the office you should automatically be logged in to BaysideSearch, if on an external computer

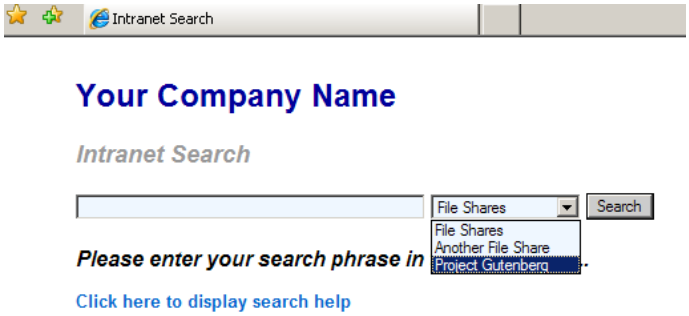
you will be prompted for you user id and password.

After logging on you will be presented with the following initial screen.



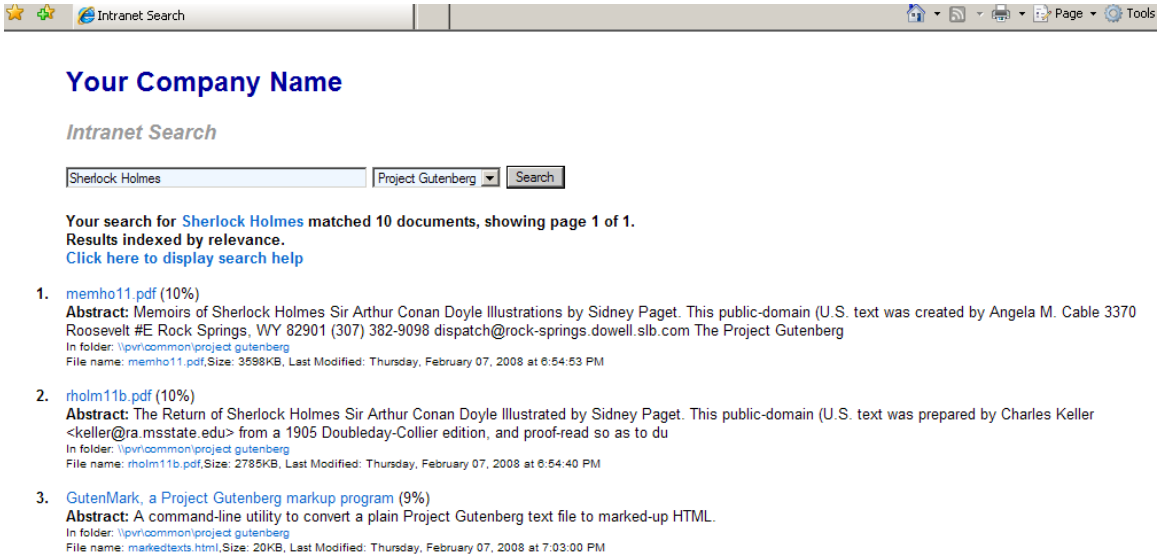
2.2.2 Choose Index

If your company is setup with the ability to limit the search results by functional area or other segregation, you will need to next select which index you wish to search.



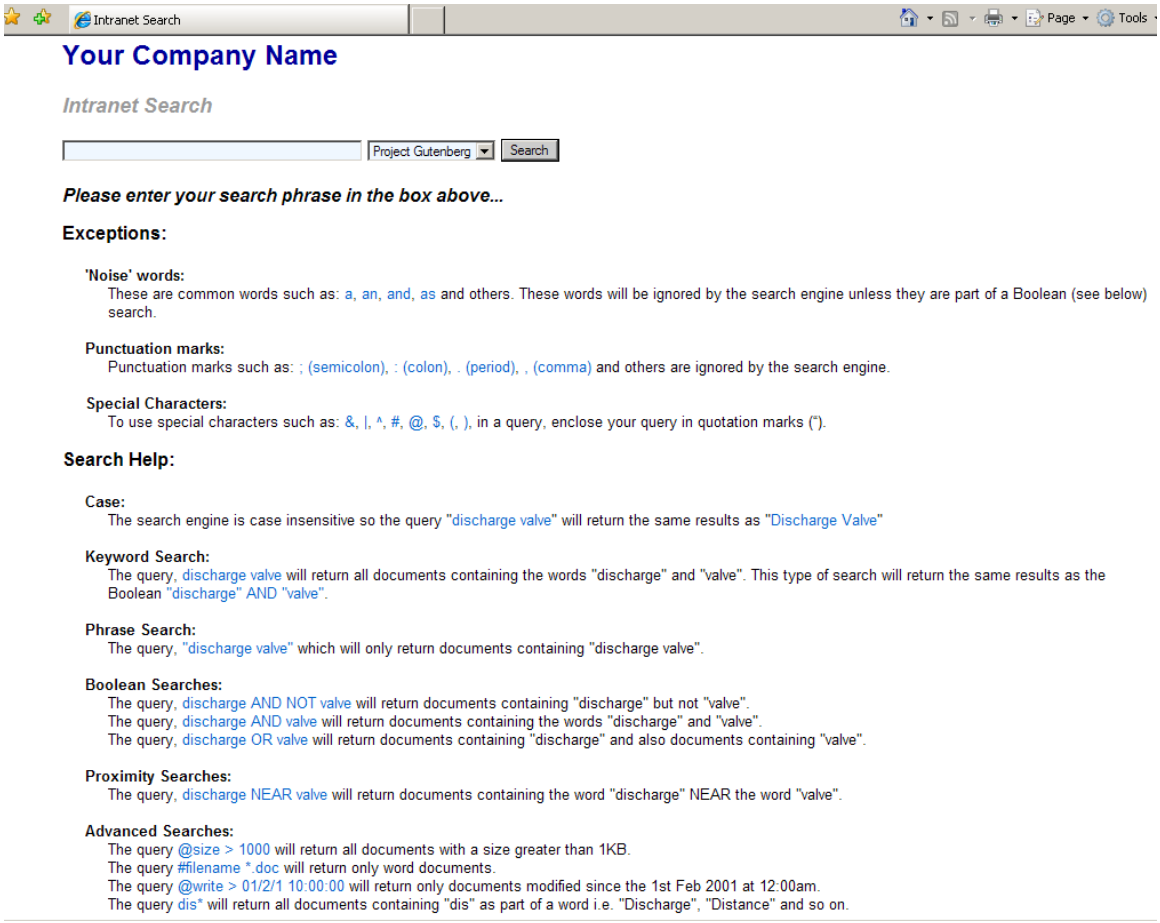
2.2.3 Perform Search & Results

Enter your search parameters and click **Search**. If your search returns any results you will be presented with the following results screen.



The blue text within the results are click-able links to the file and the directory that the file is located in.

If your search returns to many results you can limit the search by adding additional search parameters. There is a on-line query syntax help guide accessed by clicking on **Click here to display search help**. The following help will appear below. Read chapter 3 of this document for more examples and a in-depth explanation of the query syntax.


Intranet Search
Page
Tools
Your Company Name
Intranet Search
Project Gutenberg
Search
Please enter your search phrase in the box above...
Exceptions:
'Noise' words:
These are common words such as: a, an, and, as and others. These words will be ignored by the search engine unless they are part of a Boolean (see below) search.
Punctuation marks:
Punctuation marks such as: ; (semicolon), : (colon), . (period), , (comma) and others are ignored by the search engine.
Special Characters:
To use special characters such as: &, |, ^, #, @, $, (,), in a query, enclose your query in quotation marks (").
Search Help:
Case:
The search engine is case insensitive so the query "discharge valve" will return the same results as "Discharge Valve"
Keyword Search:
The query, discharge valve will return all documents containing the words "discharge" and "valve". This type of search will return the same results as the Boolean "discharge" AND "valve".
Phrase Search:
The query, "discharge valve" which will only return documents containing "discharge valve".
Boolean Searches:
The query, discharge AND NOT valve will return documents containing "discharge" but not "valve".
The query, discharge AND valve will return documents containing the words "discharge" and "valve".
The query, discharge OR valve will return documents containing "discharge" and also documents containing "valve".
Proximity Searches:
The query, discharge NEAR valve will return documents containing the word "discharge" NEAR the word "valve".
Advanced Searches:
The query @size > 1000 will return all documents with a size greater than 1KB.
The query #filename *.doc will return only word documents.
The query @write > 01/2/1 10:00:00 will return only documents modified since the 1st Feb 2001 at 12:00am.
The query dis* will return all documents containing "dis" as part of a word i.e. "Discharge", "Distance" and so on.

Chapter 3

Query Language

Searches produce a list of files that contain the word or phrase no matter where they appear in the text. This list gives the rules for formulating queries:

- Consecutive words are treated as a phrase; they must appear in the same order within a matching document.
- Queries are case-insensitive, so you can type your query in uppercase or lowercase.
- You can search for any word except for those in the exception list (for English, this includes a, an, and, as, and other common words), which are ignored during a search.
- Words in the exception list are treated as placeholders in phrase and proximity queries. For example, if you searched for Word for Windows, the results could give you Word for Windows and Word and Windows, because for is a noise word and appears in the exception list.
- Punctuation marks such as the period (.), colon (:), semicolon (;), and comma (,) are ignored during a search.
- To use specially treated characters such as &, —, ^, #, @, $, (,), in a query, enclose your query in quotation marks ().
- To search for a word or phrase containing quotation marks, enclose the entire phrase in quotation marks and then double the quotation marks around the word or words you want to surround with quotes. For example, World-Wide Web or Web searches for World-Wide Web or Web.
- You can insert Boolean operators (**AND**, **OR**, and **NOT**) and the proximity operator (**NEAR**) to specify additional search information.
- The wildcard character (*) can match words with a given prefix. The query esc* matches the terms ESC, escape, and so on.
- Free-text queries can be specified without regard to query syntax.
- Vector space queries can be specified.
- ActiveX (OLE) and file attribute property value queries can be issued.

3.0.4 Boolean and Proximity Operators

Boolean and proximity operators can create a more precise query.

To Search For	Example	Results
Both terms in the same page	access and basic Or access & basic	Pages with both the words access and basic
Either term in a page	cgi or isapi Or cgi — isapi	Pages with the words cgi or isapi
The first term without the second term	access and not basic Or access & ! basic	Pages with the word access but not basic
Pages not matching a property value	not @size = 100 Or ! @size = 100	Pages that are not 100 bytes
Both terms in the same page, close together	excel near project Or excel p̃roject	Pages with the word excel near the word project

Hints:

- You can add parentheses to nest expressions within a query. The expressions in parentheses are evaluated before the rest of the query.
- Use double quotes () to indicate that a Boolean or **NEAR** operator keyword should be ignored in your query. For example, Abbott and Costello will match pages with the phrase, not pages that match the Boolean expression. In addition to being an operator, the word and is a noise word in English.
- The **NEAR** operator is similar to the **AND** operator in that **NEAR** returns a match if both words being searched for are in the same page. However, the **NEAR** operator differs from **AND** because the rank assigned by **NEAR** depends on the proximity of words. That is, the rank of a page with the searched-for words closer together is greater than or equal to the rank of a page where the words are farther apart. If the searched-for words are more than 50 words apart, they are not considered near enough, and the page is assigned a rank of zero.
- The **NOT** operator can be used only after an **AND** operator in content queries; it can be used only to exclude pages that match a previous content restriction. For property value queries, the **NOT** operator can be used apart from the **AND** operator.
- The **AND** operator has a higher precedence than **OR**. For example, the first three queries are equal, but the fourth is not:
  a AND b OR c
  c OR a AND b
  c OR (a AND b)
  (c OR a) AND b

Note: The NEAR operator can be applied only to words or phrases.

3.0.5 Wildcards

Wildcard operators help you find pages containing words similar to a given word.

To Search For	Example	Results
Words with the same prefix	comput*	Pages with words that have the prefix comput, such as computer, computing, and so on
Words based on the same stem	fly**	Pages with words based on the same stem as fly, such as flying, flown, flew, and so on

3.0.6 Free-Text Queries

The query engine finds pages that best match the words and phrases in a free-text query. This is done by automatically finding pages that match the meaning, not the exact wording, of the query. Boolean, proximity, and wildcard operators are ignored within a free-text query. Free-text queries are prefixed with $contents.

To Search For	Example	Results
Files that match free-text	$contents how do I print in Microsoft Excel?	Pages that mention printing and Microsoft Excel.

3.0.7 Vector Space Queries

The query engine supports vector space queries. Vector queries return pages that match a list of words and phrases. The rank of each page indicates how well the page matched the query.

To Search For	Example	Results
Pages that contain specific words	light, bulb	Files with words that best match the words being searched for
Pages that contain weighted prefixes, words, and phrases	invent*, light[50], bulb[10], ”light bulb”[400]	Files that contain words prefixed by invent, the words light, bulb, and the phrase light bulb (the terms are weighted)

- Components in vector queries are separated by commas.
- Components in vector queries can be weighted by using the [weight] syntax.
- Pages returned by vector queries do not necessarily match every term in the query.
- Vector queries work best when the results are sorted by rank.

3.0.8 Property Value Queries

With property value queries, you can find files that have property values that match a given criteria. The properties over which you can query include basic file information like file name and file size, and ActiveX properties including the document summary (information) that is stored in files created by ActiveX-aware applications.

There are two types of property queries:

- Relational property queries consist of an at character (@), a property name, a relational operator, and a property value. For example, to find all of the files larger than one million bytes, issue the query @size $>$ 1000000.

- Regular expression property queries consist of a number sign (#), a property name, and a regular expression for the property value. For example, to find to find all of the video (.avi) files, issue the query #filename *.avi. Regular expressions will never match the special properties contents (#contents) and all (#all). Properties that are not retrievable at query time cannot be used in # queries. these include HTML META properties not stored in the property cache.

Property Names

Property names are preceded by either the at (@) or number sign (#) character. Use @ for relational queries, and # for regular expression queries.

If no property name is specified, @contents is assumed.

Properties available for all files include:

Property Name	Description
All	Matches words, phrases, and any property
Contents	Words and phrases in the file
Filename	Name of the file
Size	File size
Write	Last time the file was modified

ActiveX property values can also be used in queries. Web sites with files created by most ActiveX-aware applications can be queried for these properties:

Property Name	Description
DocTitle	Title of the document
DocSubject	Subject of the document
DocAuthor	The documents author
DocKeywords	Keywords for the document
DocComments	Comments about the document

For a complete list of property names, see the List of Property Names later on this page.

Relational Operators

Relational operators are used in relational property queries.

To Search For	Example	Results
Property values in relation to a fixed value	@size < 100 @size <= 100 @size = 100 @size != 100 @size >= 100 @size > 100	Files whose size matches the query
Property values with all of a set of bits on	@attrib ^a 0x820	Compressed files with the archive bit on
Property values with some of a set of bits on	@attrib ^s 0x20	Files with the archive bit on

Property Values

To Search For	Example	Results
A specific value	@DocAuthor = Bill Barnes	Files authored by Bill Barnes
Values beginning with a prefix	#DocAuthor George*	Files whose author property begins with George
Files with any of a set of extensions	#filename *.—(exe—,dll—,sys—)	Files with .exe, .dll, or .sys extensions
Files modified after a certain date	@write > 96/2/14 10:00:00	Files modified after February 14, 1996 at 10:00 GMT
Files modified after a relative date	@write > -1d2h	Files modified in the last 26 hours
Vectors matching a vector	@vectorprop = { 10, 15, 20 }	ActiveX documents with a vectorprop value of { 10, 15, 20 }
Vectors where each value matches a criteria	@vectorprop >^a 15	ActiveX documents with a vectorprop value in which all values in the vector are greater than 15
Vectors where at least one value matches a criteria	@vectorprop =^s 15	ActiveX documents with a vectorprop value in which at least one value is 15

- Be sure to use the pound (#) character before the property name when using a regular expression in a property value, and an at (@) character otherwise. The equal (=) relational operator is assumed for regular-expression queries.

- File name (#filename) is the only property that efficiently supports regular expressions with wildcards to the left of text.

- Date and time values are of the form yyyy/mm/dd hh:mm:ss or yyyy-mm-dd hh:mm:ss. The first two characters of the year and the entire time can be omitted. If you omit the first two characters of the year, then 29 or less is interpreted as the year 2000, and 30 or greater is interpreted as the year 1900. All dates and times are in Greenwich Mean Time (GMT).

- Dates and times relative to the current time can be expressed with a minus (-) character followed by zero or by more integer unit and time unit pairs. Time units are expressed as: (y) for years, (m) for months, (w) for weeks, (d) for days, (h) for hours, (n) for minutes, and (s) for seconds. A three-digit millisecond value can be optionally specified after the seconds value in date expressions. For example, 1997/12/8 10:10:03:452

- Currency values are of the form x.y, where x is the whole value amount and y is the fractional amount. There is no assumption about units.

- Boolean values are (t) or (true) for **TRUE** and (f) or (false) for **FALSE**.

- Vectors (VT_VECTOR) are expressed as an opening brace ({), followed by a comma-separated list of values, then a closing brace (}).

- Single-value expressions that are compared against vectors are expressed as a relational operator, then a (^a) for all of or a (^s) for some of.

- Numeric values can be in decimal or hexadecimal (preceded by 0x).

- The contents property does not support relational operators. If a relational operator is specified, no results will be found. For example, @contents Microsoft will find documents containing Microsoft, but @contents=Microsoft will find none.

3.0.9 Regular Expressions

Regular expressions in property queries are defined as follows:

- Any character except asterisk (*), period (.), question mark (?), and vertical bar (—) defaults to matching just itself.

- Regular expressions can be enclosed in matching quotes (), and must be enclosed in quotes if they contain a space ( ) or closing parenthesis ()).

- The characters *, ., and ? behave as they behave in Windows; they match any number of characters, match (.) or end of string, and match any one character, respectively.

- The character — is an escape character. After —, the following characters have special meaning:
  (opens a group. Must be followed by a matching).
) closes a group. Must be preceded by a matching (.
  [opens a character class. Must be followed by a matching (un-escaped)].
  { opens a counted match. Must be followed by a matching }.
  } closes a counted match. Must be preceded by a matching {.
  , separates OR clauses.
  matches zero or more occurrences of the preceding expression.
  ? matches zero or one occurrences of the preceding expression.
  + matches one or more occurrences of the preceding expression.
  Anything else, including —, matches itself.

- Between square brackets ([]) the following characters have special meaning:
  ^ matches everything but following classes. Must be the first character.
  ] matches]. May only be preceded by ^, otherwise it closes the class.
  - range operator. Preceded and followed by normal characters.
  Anything else matches itself (or begins or ends a range at itself).

- Between curly braces ({}) the following syntax applies:
  —{m—} matches exactly m occurrences of the preceding expression. ($0 < m < 256$).
  —{m,—} matches at least m occurrences of the preceding expression. ($1 < m < 256$).
  —{m,n—} matches between m and n occurrences of the preceding expression, inclusive. ($0 < m < 256$, $0 < n < 256$).

- To match *, ., and ?, enclose them in brackets (for example, —[*]sample will match *sample).

3.0.10 Query Examples

Example	Results
@size > 1000000	Pages larger than one million bytes
@write > 95/12/23	Pages modified after the date
Apple tree	Pages with the phrase apple tree
”apple tree”	Same as above
@contents apple tree	Same as above
Microsoft and @size > 1000000	Pages with the word Microsoft that are larger than one million bytes
”microsoft and @size > 1000000”	Pages with the phrase specified (not the same as above)
#filename *.avi	Video files (the # prefix is used because the query contains a regular expression)
@attrib ^s 32	Pages with the archive attribute bit on
@docauthor = John Smith	Pages with the given author
$contents why is the sky blue?	Pages that match the query
@size < 100 & #filename *.gif	Graphics Interchange Format (GIF) files less than 100 bytes in size

3.0.11 List of Property Names

These properties are always available for queries. Additional properties may also be available depending on the configuration of the Web server.

Friendly Name	Datatype	Property
A_HRef	DBTYPE_WSTR — DBTYPE_BYREF	Text of HTML HREF. This property name was created for Microsoft Site Server and corresponds with the Indexing Service property name HtmlHRef. *Can be queried but not retrieved.*
Access	VT_FILETIME	Last time file was accessed.
All	(not applicable)	Searches every property for a string. *Can be queried but not retrieved.*
AllocSize	DBTYPE_I8	Size of disk allocation for file.
Attrib	DBTYPE_UI4	File attributes. Documented in Win32 SDK.
ClassId	DBTYPE_GUID	Class ID of object, for example, WordPerfect, Word, and so on.
Characterization	DBTYPE_WSTR — DBTYPE_BYREF	Characterization, or abstract, of document. Computed by Indexing Service.
Contents	(not applicable)	Main contents of file. *Can be queried but not retrieved.*
Create	VT_FILETIME	Time file was created.
Directory	DBTYPE_WSTR — DBTYPE_BYREF	Physical path to the file, not including the file name.
DocAppName	DBTYPE_WSTR — DBTYPE_BYREF	Name of application that created the file.
DocAuthor	DBTYPE_WSTR — DBTYPE_BYREF	Author of document.
DocByteCount	DBTYPE_I4	Number of bytes in a document.
DocCategory	DBTYPE_STR — DBTYPE_BYREF	Type of document such as a memo, schedule, or whitepaper.
DocCharCount	DBTYPE_I4	Number of characters in document.
DocComments	DBTYPE_WSTR — DBTYPE_BYREF	Comments about document.
DocCompany	DBTYPE_STR — DBTYPE_BYREF	Name of the company for which the document was written.
DocCreatedTm	VT_FILETIME	Time document was created.
DocEditTime	VT_FILETIME	Total time spent editing document.
DocHiddenCount	DBTYPE_I4	Number of hidden slides in a Microsoft PowerPoint document.
DocKeywords	DBTYPE_WSTR — DBTYPE_BYREF	Document keywords.
DocLastAuthor	DBTYPE_WSTR — DBTYPE_BYREF	Most recent user who edited document.
DocLastPrinted	VT_FILETIME	Time document was last printed.
DocLastSavedTm	VT_FILETIME	Time document was last saved.
DocLineCount	DBTYPE_I4	Number of lines contained in a document.
DocManager	DBTYPE_STR — DBTYPE_BYREF	Name of the manager of the document's author.
DocNoteCount	DBTYPE_I4	Number of pages with notes in a PowerPoint document.
DocPageCount	DBTYPE_I4	Number of pages in document.
DocParaCount	DBTYPE_I4	Number of paragraphs in a document.
DocPartTitles	DBTYPE_STR — DBTYPE_VECTOR	Names of document parts. For example, in Excel part titles are the names of spread sheets, in PowerPoint slide titles, and in Word for Windows the names of the documents in the master document.
DocPresentationTarget	DBTYPE_STR—DBTYPE_BYREF	Target format (35mm, printer, video, and so on) for a presentation in PowerPoint.
DocRevNumber	DBTYPE_WSTR — DBTYPE_BYREF	Current version number of document.
DocSlideCount	DBTYPE_I4	Number of slides in a PowerPoint document.

Friendly Name	Datatype	Property
DocSubject	DBTYPE_WSTR — DBTYPE_BYREF	Subject of document.
DocTemplate	DBTYPE_WSTR — DBTYPE_BYREF	Name of template for document.
DocTitle	DBTYPE_WSTR — DBTYPE_BYREF	Title of document.
DocWordCount	DBTYPE_I4	Number of words in document.
FileIndex	DBTYPE_I8	Unique ID of file.
FileName	DBTYPE_WSTR — DBTYPE_BYREF	Name of file.
HitCount	DBTYPE_I4	Number of hits (words matching query) in file.
HtmlHRef	DBTYPE_WSTR — DBTYPE_BYREF	Text of HTML HREF. *Can be queried but not retrieved.*
HtmlHeading1	DBTYPE_WSTR — DBTYPE_BYREF	Text of HTML document in style H1. *Can be queried but not retrieved.*
HtmlHeading2	DBTYPE_WSTR — DBTYPE_BYREF	Text of HTML document in style H2. *Can be queried but not retrieved.*
HtmlHeading3	DBTYPE_WSTR — DBTYPE_BYREF	Text of HTML document in style H3. *Can be queried but not retrieved.*
HtmlHeading4	DBTYPE_WSTR — DBTYPE_BYREF	Text of HTML document in style H4. *Can be queried but not retrieved.*
HtmlHeading5	DBTYPE_WSTR — DBTYPE_BYREF	Text of HTML document in style H5. *Can be queried but not retrieved.*
HtmlHeading6	DBTYPE_WSTR — DBTYPE_BYREF	Text of HTML document in style H6. *Can be queried but not retrieved.*
Img_Alt	DBTYPE_WSTR — DBTYPE_BYREF	Alternate text for <IMG> tags. *Can be queried but not retrieved.*
Path	DBTYPE_WSTR — DBTYPE_BYREF	Full physical path to file, including file name.
Rank	DBTYPE_I4	Rank of row. Ranges from 0 to 1000. Larger numbers indicate better matches.
RankVector	DBTYPE_I4 — DBTYPE_VECTOR	Ranks of individual components of a vector query.
ShortFileName	DBTYPE_WSTR — DBTYPE_BYREF	Short (8.3) file name.
Size	DBTYPE_I8	Size of file, in bytes.
USN	DBTYPE_I8	Update Sequence Number. NTFS drives only.
VPath	DBTYPE_WSTR — DBTYPE_BYREF	Full virtual path to file, including file name. If more than one possible path, then the best match for the specific query is chosen.
WorkId	DBTYPE_I4	Internal ID for file. Used within Indexing Service.
Write	VT_FILETIME	Last time file was written.