

Ransomware Detection powered by CyberSense®

v8.13.0

Qshell API Guide

Describes the Application Programming Interface (API) that allows users to programmatically control operations on data that has been indexed by the Ransomware Detection software.

© 2023, 2025 Hitachi Vantara. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including copying and recording, or stored in a database or retrieval system for commercial purposes without the express written permission of Hitachi, Ltd., or Hitachi Vantara LLC (collectively "Hitachi"). Licensee may make copies of the Materials provided that any such copy is: (i) created as an essential step in utilization of the Software as licensed and is used in no other manner; or (ii) used for archival purposes. Licensee may not make any other copies of the Materials. "Materials" mean text, data, photographs, graphics, audio, video and documents.

Hitachi reserves the right to make changes to this Material at any time without notice and assumes no responsibility for its use. The Materials contain the most current information available at the time of publication.

Some of the features described in the Materials might not be currently available. Refer to the most recent product announcement for information about feature and product availability, or contact Hitachi Vantara LLC at https://support.hitachivantara.com/en_us/contact-us.html.

Notice: Hitachi products and services can be ordered only under the terms and conditions of the applicable Hitachi agreements. The use of Hitachi products is governed by the terms of your agreements with Hitachi Vantara LLC.

By using this software, you agree that you are responsible for:

- 1) Acquiring the relevant consents as may be required under local privacy laws or otherwise from authorized employees and other individuals; and
- 2) Verifying that your data continues to be held, retrieved, deleted, or otherwise processed in accordance with relevant laws.

Notice on Export Controls. The technical data and technology inherent in this Document may be subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Reader agrees to comply strictly with all such regulations and acknowledges that Reader has the responsibility to obtain licenses to export, re-export, or import the Document and any Compliant Products.

Hitachi and Lumada are trademarks or registered trademarks of Hitachi, Ltd., in the United States and other countries.

AIX, DS6000, DS8000, Enterprise Storage Server, eServer, FICON, FlashCopy, GDPS, HyperSwap, IBM, OS/390, PowerHA, PowerPC, S/390, System z9, System z10, Tivoli, z/OS, z9, z10, z13, z14, z15, z16, z/VM, and z/VSE are registered trademarks or trademarks of International Business Machines Corporation.

Active Directory, ActiveX, Bing, Excel, Hyper-V, Internet Explorer, the Internet Explorer logo, Microsoft, Microsoft Edge, the Microsoft corporate logo, the Microsoft Edge logo, MS-DOS, Outlook, PowerPoint, SharePoint, Silverlight, SmartScreen, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, the Windows logo, Windows Azure, Windows PowerShell, Windows Server, the Windows start button, and Windows Vista are registered trademarks or trademarks of Microsoft Corporation. Microsoft product screen shots are reprinted with permission from Microsoft Corporation.

All other trademarks, service marks, and company names in this document or website are properties of their respective owners.

Copyright and license information for third-party and open source software used in Hitachi Vantara products can be found in the product documentation, at <https://www.hitachivantara.com/en-us/company/legal.html>.

CyberSense® is a registered trademark of Index Engines, Inc.

Contents

Chapter 1. Introduction.....	5
The qshell Command Line Interface.....	7
Usage.....	7
Entering and exiting qshell.....	7
Shell built-in features.....	7
qshell Commands.....	10
ADD.....	10
BACKUPSETS.....	10
ESSENCE.....	10
EXIT.....	11
HELP.....	11
LISTTAGS.....	12
QUERY.....	12
REBUILDTAGS.....	12
REMOVE.....	12
RESET.....	13
SAVE.....	14
SEGMENTS.....	14
SET.....	14
SHOW.....	30
SOURCE.....	31
TREE.....	31
Chapter 2. C Library Interface (This section is in progress).....	33
Appendix A. CSV Properties and Fields.....	35
CSV fields.....	36
Appendix B. User Expression Fields.....	43

Chapter 1. Introduction

The Application Programming Interface (API) described in this document allows users to programmatically control operations on data that has been indexed by the Index Engines software. This API does not provide any functionality for controlling operations during the indexing of data.

The API comes in two formats: a set of shell commands that run on the Index Engines server, and a C-Language programming interface. Both formats are described in this document. In addition, the API can be run interactively to facilitate learning the API and debugging user software.



Note: Phase 2 deliverables are noted as such throughout this document.

This API provides the ability to run queries that:

- Produce results in a CSV file format
- Produce reports in a CSV file format
- Tag data in the index
- Remove tags from data in the index
- Modify the owner of a file within the index
- Extract data to a destination (phase 2 delivery)
- Copy (phase 2 delivery)
- Delete from source
- Delete from index (phase 2 delivery)
- Archive data (phase 2 delivery)

The functionality listed above supports many workflows. Some typical workflows include the following:

- Query for a number of document and then send the reconstruction of the text in the document to another tool for further analysis or analytics.
- Query for a number of documents and then extract the documents with a load file that can be used to load the documents into a review tool that supports the Concordance load file specification.
- Query for documents that are required for legal hold and move them into the archive.
- Query for a number of documents and then produce reports about this set of documents.

The API provides the ability to determine which metadata fields are included in the query results CSV file.

- The *Index Engines Metadata Field Definitions* document provides details about the metadata fields.
- The *Index Engines Query Operators Guide* provides a listing and details about the operators that can be used in the query.
- See the *Index Engines Search Guide* for information regarding the use of tags.

This API also provides the ability to produce customized histogram reports that are not available through the Index Engines Graphical User Interface (GUI). The customized reports can contain multiple axes where the range, binning, and scaling of the axes (linear, log, etc.), can all be specified. The report can also contain multiple metadata fields where for each field, a function is applied to the field values (sum, average, standard deviation, maximum, or minimum) for all results in a given bin to produce the histogram value for that bin.

The qshell Command Line Interface

Usage

In a CLI window on your Index Engines server, enter the following to display the help options of the qshell API:

```
$ qshell -h
Usage: qshell [options] [-c <command> | <file>...]

Commands can be run non-interactively, using:
  -c <command> .... run command directly
  <file> ..... read commands to run from file
Alternatively, commands can be passed into qshell via a
command-line pipe (e.g., echo "cmd" | qshell).
If no command or file is provided, qshell will run in interactive
mode, allowing commands to be entered at the qshell prompt.
To see a list of supported commands, enter: qshell -c help
Options:
  -s <server> .... name of the server running the Index Engines software
  -p <port> ..... port to use
  -D ..... do not read .qshrc file on startup
  -V ..... version info, build date & time
```

When running in interactive mode, and unless the `-D` option has been specified, qshell reads and executes commands from `~/ .qshrc`, if that file exists, allowing a customized configuration to be specified.

Entering and exiting qshell

The interactive qshell is entered by typing `qshell` (possibly followed by one or more options) at the Linux command prompt:

```
$ qshell
(qshell)
```

To exit the qshell, type `exit`:

```
(qshell) exit
$
```

Shell built-in features

The qshell API supports common shell features such as I/O redirection (via `|` and `>`) and shell variables.

Read-only qshell variables

The following variables are maintained by the qshell as read-only:

QID

ID associated with the most recently executed query

QHASH

Hash code associated with the most recently executed query

CSVFILE

Path to most recently created CSV output file on the server

NFOUND

Total number of results found by the most recently executed query

Additional variables can be defined in the `VAR=value` format, and referenced with `$VAR`.

echo

The `echo` command can be used to display its arguments, including any referenced variables, to standard out (stdout).

Flow control

The qshell supports several bash-like flow control constructs. Conditional expressions are bracketed test expressions as described for the `bash(1)` `test` built-in command, but can alternatively take the value of either `true` or `false`. The `break` command is available to immediately exit the lowest level enclosing loop. The `continue` command is available to immediately begin the next iteration of the lowest level enclosing loop.

The for construct

```
for VARIABLE in <value-list>
do
commands ...
done
```

The if construct

```
if [ condition1 ]
then
commands ...
elif [ condition2 ]
commands ...
else
commands ...
fi
```

The `elif` and `else` clauses are optional; `elif` may also be specified multiple times.

The while construct

```
while [ condition ]  
do  
  commands ...  
done
```

qshell Commands

ADD

For any option described under [set \(on page 14\)](#) that takes a comma-separated list of values, the add command can be used to add another value to the set of values already specified. See also [set \(on page 14\)](#) and [remove \(on page 12\)](#).

Usage is:

```
add <option> <value>
```

Options are:

Option	Argument
amask	{bcc body cc container default filename ocr sender subject to}
dedup	{content filename msgid msgidelsecont mtime occurrence path size supplemental family owner}
locations	[-f <file> <subtree> [<subtree>...]]
logflags	{no_cache no_log}
prodsig	{content filename msgid msgidelsecont mtime occurrence path size supplemental family owner}
queryflags	{archived content dontexclude excludepc federate files hidearchivable hideextractable ignorededup inversededup locked nobackups onlyarchivable onlydeleted onlyextractable persist produced unlocked wildcardscan withdeleted}
ranks	<rank-number> [<rank-number> ...]

BACKUPSETS

List the backupsets in a segment from the query server. Usage is:

```
backupsets [-n node] <segment>
```

ESSENCE

Show the essence for one result of a query. Usage is:

```
essence <rank> [<query term> ... ]
```

The `<query term>` field can be omitted (and will otherwise be ignored) if `qid` and `qhash` have been defined (see [set qid \(on page 27\)](#) and [set qhash \(on page 27\)](#)).

EXIT

Exit the qshell. Usage is:

```
exit
```

HELP

Display help text. Usage is:

```
(qshell) help
```

Value	Description
add	Add a value to an option setting
essence	Show essence for one result of a query
exit	Exit the interactive qshell
help	Show this message
listtags	List the tags known to the query server
query	Query the server and show results
rebuildtags	Rebuild tag databases
remove	Remove a value from an option setting
reset	Close and reopen connection to server
save	Save the current configuration setting to a file
segments	List the index segments from the query server
set	Set option settings
show	Show option settings
source	Read and execute commands from a file
tag	Set or clear tags on documents
times	Show the oldest and newest backup times

Value	Description
tree	Show servers, shares and directories

LISTTAGS

List the tags known to the query server. Usage is:

```
listtags
```

QUERY

Query the server and show results. Usage is:

```
query -f <query file> | <query term> ...
```

The structure of a query term is defined in detail in the *Index Engines Query Operators Guide*.

REBUILDTAGS

Rebuild the tags database. Usage is:

```
rebuildtags [dbid ... ]
```

REMOVE

For any option described under [set \(on page 14\)](#) that takes a comma-separated list of values, the `remove` command can be used to remove a value from the set of values currently specified. If no values are specified, all values associated with the option will be removed. See also [add \(on page 10\)](#) and [set \(on page 14\)](#).

Usage is:

```
remove <option> [<value>]
```

The options are:

Option	Value
amask	{bcc body cc container default filename ocr sender subject to}
deadman	(omit args)

Option	Value
dedup	{content filename msgid msgidelsecont mtime occurrence path size supplemental family owner}
extractargs	(omit args)
formatstr	(omit args)
locations	[-f <file> <subtree> [<subtree>...]]
logflags	{no_cache no_log}
offset	(omit args)
prodsig	{content filename msgid msgidelsecont mtime occurrence path size supplemental family owner}
progress	(omit args)
qhash	(omit args)
qid	(omit args)
queryflags	{archived content dontexclude excludepc federate files hidearchivable hideextractable ignorededup inversededup locked nobackups onlyarchivable onlydeleted onlyextractable persist produced unlocked wildcardscan withdeleted}
ranks	<rank-number> [<rank-number> ...]
setowner	(omit args)
timeout	(omit args)

RESET

Close and reopen connection to the server. This will also reset the configuration settings to the system defaults, or as specified in the `~/ .qshrc` file.

Usage is:

```
reset [qshell options]
```

The `reset` command accepts the same options as the `qshell` command invocation.



Note: A binary that is written to the qshell library API can specify its own `getopt(3)` function to use when `reset` is called. Typically, this will be the same `getopt()` function called to process command line options to the binary itself.

SAVE

Saves the current configuration in the specified file. Usage is:

```
save [-f] <file>
```

If the specified file already exists, the user will be prompted to overwrite the file when running in interactive mode. Use the `-f` option to force the overwrite an existing file without being prompted.

SEGMENTS

List the index segments from the query server. Usage is:

```
segments
```

SET

Configures the query options. `set` will overwrite any previously specified value for an option. See also [add \(on page 10\)](#) and [remove \(on page 12\)](#).

Usage is:

```
set <option> <value>
```

The options are:

Option	Value
amask	{none bcc body cc container default filename ocr sender subject to}
attachmode	{all atomicmsg noattach sepattach}
colorscheme	{bold red underbar}
context	{off on}
deadman	<milliseconds>
dedup	{none content filename msgid msgidelsecont mtime occurrence path size supplemental family owner}
depth	<tree-depth>
extractargs	[-u user] [-l output-dir] [-d description]

Option	Value
format	{clearowner csvfile csvlines extract microqfstat qfstat reports reportscsv setowner tagonly}
formatstr	<format-string>
howmany	<num-results>
ids	{off on}
keywordstyle	<style-value>
knownftypemode	{all nonsrl}
locations	[-f <file> <subtree> [<subtree>...]]
logflags	{none no_cache no_log}
ntocache	<number-to-cache>
offset	<offset>
orderby	{unsorted [+]?{atime mtime name owner relevance size}}
prodsig	{none content filename msgid msgidelsecont mtime occurrence path size supplemental family owner}
production	<production-name>
progress	<milliseconds>
qhash	<hash-value>
qid	<query-id>
queryflags	{none archived content dontexcludep excludepc federate files hidearchivable hideextractable ignorededup inversededup locked nobackups onlyarchivable onlydeleted onlyextractable persist produced unlocked wildcardscan withdeleted}
quietoutput	{off on}
ranks	<rank-number> [<rank-number>...]
regexlang	{basic extended wildcards}
setowner	-o <owner> -d <dispname> -t <type> -a <apptype> -s <server>
showtags	{off on}
timeout	<milliseconds>
verbose	{off on}
verbosity	<degree>
view	{responsive parent family}

set amask <attribute-list>

Sets the file attribute mask for the query. The attribute list is specified as a comma-separated list of attributes, as defined below. The default value contains all attributes except for `container`. Supported options are:

Option	Description
<code>bcc</code>	Include the BCC field in the query
<code>body</code>	Query the main message BODY
<code>cc</code>	Include the CC field in the query
<code>container</code>	Include the CONTAINER_PATH in the query
<code>default</code>	Includes BCC , BODY , CC , FILENAME , OCR , SENDER , SUBJECT , and TO .
<code>filename</code>	Include the FILENAME in the query
<code>ocr</code>	Include OCR data in the query
<code>sender</code>	Include the SENDER /author field in the query
<code>subject</code>	Include the SUBJECT /title field in the query
<code>to</code>	Include the TO field in the query

set attachmode <value>

Describes how attachments are handled during a query. The default value is `sepattach`. Supported values are:

Value	Description
<code>all</code>	Results constitute the union of a <code>together</code> and <code>sepattach</code> search.
<code>together</code>	Search the combination of the message body and all attachments and their contents as a single entity.
<code>noattach</code>	Do not search any attachments.
<code>sepattach</code>	Search attachments as separate entities.

set colorscheme <value>

Define the color scheme to use to highlight a query operand. The default value is `underscore`. Supported values are:

Value	Description
<code>bold</code>	Highlight the operand by printing in bold.
<code>red</code>	Highlight the operand by printing in red.

Value	Description
underscore	Highlight the <code>_operand_</code> by printing an underscore before and after.

set context { off | on }

When `context` is enabled, additional context is provided for each query result. The default value is `off`.

set deadman <milliseconds>

Sets the timeout, in milliseconds, after which the query will be canceled by the server if there is no open connection from the client to poll for results or progress. The default is `-1` (disabled).

See also [set progress \(on page 26\)](#) and [set timeout \(on page 29\)](#).

set dedup <flag-list>

Sets a default deduplication mode to be used when the query doesn't explicitly provide one. Refer to the *Index Engines Query Operators Guide* for a list and description of the supported `<flag-list>` values. Multiple values can be specified as a comma-separated list. When multiple values are specified, *all* must hold true in order for the matching result to be deduplicated.

set depth <tree-depth>

Specifies the maximum depth for the tree display. The default value is `10`.

set extractargs [-u <user>] [-l <output-dir>] [-d <description>]

Specifies parameters for the `extract` format option. More information will be provided with phase 2.

set format <value>

Specifies what action should be taken and as a result, how output should be formatted. Unless otherwise specified, output is written to `stdout`. The default value is `results`.

Supported values are:

Value	Description
results	Returns a set of metadata for the results
microresults	Returns a much smaller set of metadata for the results
csvlines	For shell commands, produces results to <code>stdout</code> in human-readable format; for a library, produces metadata in text format

Value	Description
<code>bincsvlines</code>	For shell commands, produces the same output as <code>csvlines</code> ; for a library, produces metadata in a binary format
<code>csvfile</code>	Produces results in a CSV format output file on the server
<code>sqlite</code>	Produces results in SQLite formatted output file on the server
<code>tableau</code>	Produces the results in a Tableau Data Extract (TDE) formatted output file on the server
<code>reports</code>	For shell commands, produces the Custom Reports in human-readable format on stdout; for a library, produces the Custom Reports in a binary format
<code>reportscsv</code>	Produces Custom Reports in a CSV formatted output file on the server
<code>reportssqlite</code>	Produces Custom Reports in an SQLite formatted output file on the server
<code>reportstableau</code>	Produces Custom Reports in a Tableau Data Extract (TDE) formatted output file on the server
<code>tagonly</code>	Sets/removes tags
<code>setowner</code>	Overrides indexed owner with configured setowner (on page 28) values
<code>resetowner</code>	Returns to indexed owner
<code>progress</code>	Returns the progress of the running query; must be used with <code>qid</code> and <code>qhash</code>

set formatstr <format-string>

Provides details about how to interpret the format. For example, in the CSV file format, the `formatstr` can be used to select fields. Each format value defined in [set format \(on page 17\)](#) has its own set of format strings for this purpose.

Format Value	Supported format strings	Description
<code>CSVFILE</code>	<code>header=<val></code>	Whether to print header
<code>CSVLINES</code>		(0 = off)
<code>BINCSVLINES</code>	<code>tformat=<time_format></code>	A time format string, as specified on the <code>strftime(3)</code> manual page.
<code>TABLEAU</code>		
<code>SQLITE</code>	<code>properties=<property_val></code> <code>[,<property_val> ...]</code>	See Appendix A (on page 35)

Format Value	Supported format strings	Description
	encoding={UTF8 UTF-8 ASCII 8859-1 ISO88591 ISO-8859-1}	Character encoding
	eol=<eol-chars>	Set of end-of-line characters
RESULTS	keywords [*]	Include keyword with results
[*] indicates a non-default option; that is, everything is default EXCEPT for these: keywords, activity, subroot, inputsig, linkage.	nothingbut	Turns off all previously specified format options, including defaults
	path	File location
	licsizes	Licensing Sizes
	exitmsg	Exit messages/errors
	sig	Signatures
	xattr	File attributes
	mailkey	MS Exchange internal identifiers
	binfo	Backup Information
	name	File name
	discard	Flagged as deleted
	owner	File owner
	tags	Tags
	archive	Archived
	custodian	User
	activity [*]	Audit logs
	subroot [*]	If container, include contents
	ssets	Signature sets
	deacttime	Deactivation time
	inputsig	Signature of bytes for this document in the parent container before any decoding or decompression
	linkage [*]	List of attachments
REPORTS	[
REPORTSCSV	{	
	header: int,	Whether to print header

Format Value	Supported format strings	Description
REPORTSTABLEAU	cache_only: int,	Cache results, but do not return
REPORTSSQLITE	return_value: int,	0: return all values 1: return only first value 2: return second value 3: return third value; etc....
Expressed as a JSON-obj-string.	min_value_fraction: int,	Only return values that have at least this amount in parts per million of the parent
	min_value_depth: int,	Axis number at which to apply min_value_fraction
	max_depth: int,	Maximum number of axes to return
	encoding: string,	Character encoding (see above)
	eol: string,	Set of end-of-line characters
	tformat: string,	A time format string (see above)
	report_title: string,	Report title
	report_values: [Array of report values (must define at least one)
	{	
	property: int,	See Appendix A (on page 35)
	sum_function: int,	0: sum 1: average 2: minimum 3: maximum 4: std deviation

Format Value	Supported format strings	Description
	<code>user_expression: string,</code>	See Appendix B (on page 43)
	<code>label: string,</code>	Optional report value label
	<code>},</code>	
	<code>. . .</code>	
	<code>],</code>	
	<code>axes: [</code>	Array of axes (must define at least one)
	<code>{</code>	
	<code>sort: int,</code>	<p>0: Don't sort</p> <p>1: Sort by group</p> <p>2: Sort by first value</p> <p>3: Sort by second value</p> <p>4: Sort by third value</p> <p>etc. ...</p> <p>Corresponding negative values indicate a reverse sort. Each axis can have its own sorting criteria. Secondary sorts to break ties are not yet supported.</p>
	<code>property: int,</code>	See Appendix A (on page 35)
	<code>limit: int,</code>	How many bins to return (0 = all)
	<code>offset: int,</code>	Starting bin to return
	<code>bin_offset: double,</code>	Lowest value in first bin
	<code>nbins: int,</code>	Number of histogram bins
	<code>logscale: int,</code>	<p>0: linear</p> <p>10: base 10</p> <p>1024: KB, MB, GB, ...</p>

Format Value	Supported format strings	Description
		86400: days, weeks, months, years...
	<code>axistype: int,</code>	Ignored
	<code>width: double,</code>	Bin width, if nbins not specified
	<code>range: {</code>	
	<code>from: string,</code>	Start of range
	<code>to: string,</code>	End of range
	<code>exactly: string,</code>	Single element (no range)
	<code>},</code>	
	<code>widthunit: int,</code>	Type of unit to interpret "width": 0: default (Bytes or seconds) 1: KB 2: MB 3: GB 4: TB 5: PB 6: EB 0: seconds 16: minutes 17: hours 18: days 19: weeks 20: months 21: years
	<code>rangeunit: int,</code>	Type of unit to interpret "range": (Same units as listed above for "width")

Format Value	Supported format strings	Description
	flags: int,	Future use
	user_expression: string,	See Appendix B (on page 43)
	user_detail_expression: string,	Expression evaluated and returned for first result; see Appendix B (on page 43)
	label string,	Optional axis label
	},	
	. . .	
],	
	}	
]	
TAGONLY	F	Propagate tag operation to Family
	A	Apply tag operation to all archivable parts
	D<dedup>	Apply tag operation to all duplicates (as specified by <dedup>).
	+<tag>	Set tag
	-<tag>	Unset tag
	/ <i><regexp></i>	Unset tags that match a wildcard regular expression <regexp>
		<p>RE: TAGS, PLEASE NOTE:</p> <ul style="list-style-type: none"> • One of the following is required: +, -, or / . • F, A, and D are optional. • A comma-separated list of tag operations is permitted if each operation is surrounded by double quotes. Literal quotation marks are escaped by doubling. • <dedup> must be specified using an uppercase hexadecimal number and be terminated by a non-hex character. <dedup> defaults to content and path if omitted.

Format Value	Supported format strings	Description
		<ul style="list-style-type: none"> The system recognizes the pre-defined tags listed below. These literals can be used wherever <tag> is indicated above:
	/retain	Retain files tagged for deletion on merge
	/discard	Tag for deletion; will no longer show up in queries
	/collect	Tag for archiving
MICRORESULTS	Not applicable	
PROGRESS	Not applicable	

set howmany <num-results>

Defines the number of query results to return or the number of results upon which to perform an action, beginning after the offset specified by `set offset`. The default value is 10.

Setting `howmany` to -1 removes the limit from this setting. It returns or acts on all the results, starting at the offset and up to the number cached, as controlled by `ntocache`.



Important:

When using a `howmany` of -1 with [set ntolcache \(on page 25\)](#) also at -1, if the query format is either `results`, `csvlines` or `bincsvlines`, it's possible to exceed the limits allowed by memory and/or the XDR RPC protocol if the query matches a large number of documents.

When using a value of `howmany` that's greater than the value of `ntocache`, the query still only returns at most `ntocache` results. (Here -1 is considered to be greater than any positive value.)

Specifying `howmany` clears any previously defined `ranks`.

See also [set offset \(on page 26\)](#), [set ranks \(on page 28\)](#), and [set ntolcache \(on page 25\)](#).

set ids { off | on }

When `ids` is enabled, the result id (`resid`) associated with each query result is included in the output. The default value is `off`. This only affects `qfstat`.

set keywordstyle <style-value>

The `keywordstyle` option modifies how keywords are presented to the user when the `qfstat` format is specified. The default value is 0. `keywords` must be specified in the format string.

See [set formatstr \(on page 18\)](#) in order for this option to be effective.

Currently the supported values are 0 and 1.

set knownftypemode <ftype>

Describes which known file types to include in the query. The default value is `all`. Supported values are:

Value	Description
<code>all</code>	Include all known file types in query.
<code>no_nsrl</code>	Do not include NIST/NSRL files in the query.

set locations [-f <filename> | <subtree> [<subtree> ...]]

Limits query results to the specified subtrees. It is undefined by default (entire tree is queried).

set logflags <flag-list>

Use to optionally disable logging and/or caching on the server; it is specified as a comma-separated list of flags. The default value is `unset`. Supported values are:

Value	Description
<code>no_cache</code>	Disable caching; results will not be cached on the server, so a query run with the same qid (on page 27) / qhash (on page 27) will need to run from scratch
<code>no_log</code>	Disable logging; the query will not appear in the <code>qserv.log</code> on the server

set ntocache <number-to-cache>

Defines the number of query results to cache on the server. The default value is 10.

`ntocache` must be large enough to cover the highest ranked file that is returned, based on the `offset`, `howmany`, and `ranks` settings. That is, it must be at least as large as `offset + howmany` or the highest rank specified by `ranks`. A value of `-1` will cache all results on the server.

See also [set offset \(on page 26\)](#), [set howmany \(on page 24\)](#), and [set ranks \(on page 28\)](#).

set offset <offset>

Defines the offset into the query results at which to start reporting matches. This is a 0-based offset, with a default value of 0 (i.e., begin with first result (rank = 1)).

Specifying `offset` will clear any previously defined `ranks`.

See also [set howmany \(on page 24\)](#), [set ranks \(on page 28\)](#), and [set ntocache \(on page 25\)](#).

set orderby <value>

Defines how query results should be ordered. The default value is `relevance`.

Supported values are:

Value	Description
<code>atime</code>	Order by access time (most recent first)
<code>mtime</code>	Order by modification time (most recent first)
<code>name</code>	Order by name (alphabetic)
<code>unsorted</code>	Do not order results
<code>owner</code>	Order by owner (alphabetic)
<code>relevance</code>	Order by relevance (most relevant first)
<code>size</code>	Order by size (largest to smallest)

To reverse the ordering, prepend a "-" (hyphen) to the value (e.g., to sort by name in descending order, use `set orderby -name`).

set prodsig <flag-list>

Specifies the deduplication flags used to generate production signatures. The default setting is `none`. See also [set dedup \(on page 17\)](#).

See the *Index Engines User Guide* for a definition of a production.

set production <production-name>

Defines the production name to use. It is undefined by default.

set progress <milliseconds>

Sets the progress update time in milliseconds. The default value is `-1` (disabled).

See also [set deadman \(on page 17\)](#) and [set timeout \(on page 29\)](#).

set qhash <hash-value>

In conjunction with the `qid` option, refers to a previously executed query.

set qid <query-id>

In conjunction with the `qhash` option, refers to a previously executed query.

set queryflags <flag-list>

Specifies restrictions to the query. They are specified as a comma-separated list. The default value is `unset`, which means that backup sets are included and the wildcard characters '?' and '*' are treated as literals.

Supported flags are:

Value	Description
<code>archived</code>	When combined with EXCLUDEPC , limit query to unarchived data; otherwise limit to archived data.
<code>content</code>	Restrict query to file content.
<code>dontexcludepc</code>	Modifier for the PRODUCED flag, as described below.
<code>excludepc</code>	Modifier for the PRODUCED and ARCHIVED flags, as described above and below.
<code>federate</code>	Apply query to a federation.
<code>files</code>	Limit query to the file level.
<code>hidearchivable</code>	Do not include archivable files.
<code>hideextractable</code>	Do not include extractable files.
<code>ignorededup</code>	Disable deduplication.
<code>inversededup</code>	Limit query to duplicates.
<code>locked</code>	Restrict query to locked data.
<code>nobackups</code>	Do not include backup sets.
<code>onlyarchivable</code>	Limit query to archivable files
<code>onlydeleted</code>	Limit query to deleted data.
<code>onlydirs</code>	Limit query to only folders.
<code>onlyextractable</code>	Limit query to extractable files.
<code>persist</code>	Persist query data on disk.
<code>produced</code>	When combined with either ARCHIVED or EXCLUDEPC but <i>not</i> DONTEXCLUDEPC , limit query to unproduced data; otherwise limit to produced data.

Value	Description
unlocked	Restrict query to unlocked data.
wildcardscan	Treat '?' and '*' as wildcards, and not literals.
withdeleted	Include deleted data.
withdirs	Include folders.

set quietoutput { off | on }

Suppresses the printing of very large query strings as part of the query results summary. The default value is `off`, meaning the query string is printed by default.

set ranks <rank-list>

Defines specific results to return. The ranks are specified in a comma-separated list. The list is unset by default. If no `<rank-list>` argument is specified, any previous ranks setting will be cleared.

Specifying ranks will override any previously defined `offset` and `howmany` settings.

See also [set offset \(on page 26\)](#), [set howmany \(on page 24\)](#), and [set ntocache \(on page 25\)](#).

set regexlang <value>

Defines the regular expression language to use for pattern matching. The default value is `wildcards`.

Supported values are:

Value	Description
basic	basic POSIX regular expressions
extended	extended POSIX regular expressions
wildcards	wildcard expressions using '?' and '*'

set setowner -o <owner> -d <dispname> -t <type> -a <apptype> -s <server>

Specifies parameters required when format is defined as `setowner`.

set showtags { off | on }

When enabled, any tags on the FID associated with each query result are included in the output. The default value is `off`. This option applies only when [format \(on page 17\)](#) is specified as `qfstat`.

set timeout <milliseconds>

Specifies an absolute query timeout in milliseconds. Its default value is 0 msec (disabled).

See also [set deadman \(on page 17\)](#) and [set progress \(on page 26\)](#).

set verbose { off | on }

Controls whether additional information should be printed. Any additional information is printed to stderr. The default value is `on`.

set verbosity <degree>

Controls how much additional information should be printed. Its default value is 1.

set view <value>

Specifies how data is presented. The default value is `responsive`.

Supported values are:

Value	Description
backupset	Display the backupset associated with the object found.
family	Display all objects in the family of the object found.
familydups	Display all duplicates in the family of the object found.
familywithdups	Display all objects including duplicates in the family of the object found.
parent	Display only the parent associated with the object found.
parentdups	Display the parents associated with only duplicates of the object found.
parentwithdups	Display the parents associated with the object found and its duplicates.
responsive	Display the actual objects found.
responsivedups	Display only duplicates of the objects found.
responsivewithdups	Display all objects found including duplicates.

set viewmask <flag-list>

Sets the **Views to Process** preference. It defines the set of views that a given query should prepare, eliminating the need for the system to rerun the full query when a different view is requested. Please see the *Index Engines Search Guide* for more information.

The `<flag-list>` is a comma-separated list of values as defined for [set view \(on page 29\)](#). After the query is run, the user can change the `view` flag to modify how the data is presented when [query \(on page 12\)](#) is rerun using the previously returned [qid \(on page 27\)](#) and [qhash \(on page 27\)](#).

SHOW

The `show` command displays the current configuration of options.

Usage is:

```
show [ <option> ]
```

Options are:

Options	Description
<code>amask</code>	Mask of file attributes to include in query (default: 0x27F)
<code>attachmode</code>	Specify how attachments should be handled (default: sepattach)
<code>colorscheme</code>	Color scheme to use to highlight a query operand (default: underscore)
<code>context</code>	show snippets for each query result (default: off)
<code>dedup</code>	How to deduplicate the results (default: path)
<code>deadman</code>	Time, in milliseconds, to wait before canceling an abandoned query (default: disabled)
<code>depth</code>	Maximum depth for tree display (default: 10)
<code>extractargs</code>	Additional parameters to send with extract format (default: undefined)
<code>format</code>	Format in which to return the results (default: qfstat)
<code>formatstr</code>	Specify formatting characteristics based on format (default: empty string)
<code>howmany</code>	Number of results per query request (default: 10)
<code>ids</code>	Show node-database-fid for each query result (default: off)
<code>keywordstyle</code>	Modify how keyword output is displayed (default: 0)
<code>knownftypemode</code>	Known file types included in query (default: all)
<code>locations</code>	Limit results to the defined subtrees (default: search all)
<code>logflags</code>	Mask to disable logging and/or caching (default: none)
<code>ntocache</code>	Number of results to cache server side (default: 10)
<code>offset</code>	Offset into the results from where to start display (default: 0)
<code>orderby</code>	How results should be ordered (default: unsorted)

Options	Description
prodsig	Deduplication flags to generate production signature (default: none)
production	The production with which to associate the results (default: undefined)
progress	Time, in milliseconds, between progress updates (default: disabled)
qid	The query ID of a saved query (default: 0)
qhash	Hash value to use for a query (default: 0)
queryflags	Mask of special query directives (default: none)
quietoutput	Remove query string from search summary (default: off)
ranks	Specific results to fetch by rank (default: undefined)
regexlang	Define the regular expression language to use (default: wildcards)
setowner	Additional parameters to send with setowner format (default: undefined)
showtags	Show tags on a fid for each query result (default: off)
timeout	Time, in milliseconds, for query timeout (default: disabled)
verbose	Controls printing of extra information to stderr (default: on)
verbosity	Controls how much extra information is printed to stderr (default: 1)
view	Specify how data is presented (default: responsive)

SOURCE

The `source` command is used to read and execute commands from a file. Usage is:

```
source <file>
```

TREE

Show servers, shares, and directories. Usage is:

```
tree [-rsubroot_resid] [ <path> ]
```


Chapter 2. C Library Interface (This section is in progress)

```
#include "qshell.h"
```

Initialize the qshell environment:

```
void qshell_init(qshell_t *s);
```

Connect to the query server:

```
int qshell_connect(qshell_t *s);
```

Process a qshell command line:

```
int qshell_line(qshell_t *s, char *line);
```

Read and process commands specified in an input file:

```
int qshell_readfile(qshell_t *s, const char *filename);
```

Tear down the query server connection:

```
void qshell_deinit(qshell_t *s);
```


Appendix A. CSV Properties and Fields

Use the CSV property numbers listed in [CSV fields \(on page 36\)](#) to select fields to include in the **CSVFILE** or **REPORTS** format output. Each element of the properties list is a property number and an optional argument (the latter which applies only to **TRUSTED**, **EXPRESSION**, **RECONSTRUCTION**, **ESSENCE**, and **PITSTATE**), and an optional label.

EXAMPLE

The following example generates a CSV with columns for Size, the path and file name concatenated as Full Path, and a third column showing the File Type:

```
properties=4,89("U:PATH + FILENAME")[Full Path],10
```

CSV fields

This table, alphabetized by each CSV field name, lists the field tokens you can specify in a search when using the WHERE operator. Property Numbers are included for users who need them for the qshell format string.

CSV Field Name	Field Token	Type	Property Number
Access Time	ATIME	time	6
ACL List	ACLLIST	string	82
Aggregated ID	AGGRID	integer	151
Archived	ARCHIVED	string	36
Attached Document Count	ATTACHDOCCOUNT	integer	64
Attached Document Signature	ATTACHDOCSIG	string	81
Audit Logs	AUDITLOGS	string	83
Author	AUTHOR	string	12
Backup Client Type	BACKUPCTYPE	integer	93
Backup Format	BACKUPFORMAT	string	131
Backup Format Revision	BACKUPFORMATREV	string	132
Backup Host	BACKUPHOST	string	23
Backup Number	BACKUPNUM	integer	112
Backup Policy	BACKUPPOLICY	string	24
Backup Servers	BACKUPSERVERS	string	140
Backupset Aux Info	BACKUPSETAUX	string	86
Backupset ID	BACKUPSETID	string	25
Backupset ID Long Form	BACKUPSETLONGID	string	87
Backup Software	BACKUPSOFT	string	22
Backup Time	BACKUPTIME	time	26
Backup Volume	BACKUPVOL	string	21
Barcode	BARCODE	string	45
Bcc:	BCC	string	52
Bcc: Domains	BCCDOMAINS	string	100
Capacity Licensing Container	CAPLICCONTAINER	string	139

CSV Field Name	Field Token	Type	Property Number
Cc:	CC	string	14
Cc: Domains	CCDOMAINS	string	99
Conditional Terms	CONDTERMS	string	67
Conditional Terms (Top)	CONDTERMSTOP	string	94
Container Path	CONTPATH	string	70
Corrupt Flag	CORRUPTFLAG	integer	158
Creation Time	CTIME	time	7
Data Connector Name	DCN	string	61
Deactivation Time	DEACTTIME	time	101
Deduplicated Copies	DEDUPCOPIES	integer	40
Deduplicated Copy Size	DEDUPCOPYSIZE	integer	55
Deletion Time	DTIME	time	58
Document Author	DOCAUTH	string	75
Document Class	DOCUMENTCLASS	string	54
Document Content Signature	CONTSIG	string	41
Document Creation Time	DOCCTIME	time	38
Document Essence	ESSENCE	string	34
Document Modification Time	DOCMTIME	time	37
Document Reconstruction	RECONSTRUCTION	string	69
Document Title	DOCTITLE	string	73
Durable ID	DURID	string	29
Email From:	EMAILFROM	string	74
Email Received Time	EMAILRECEIV	time	78
Email Sent Time	EMAILSENT	time	76
Email Subject	EMAILSUBJ	string	72
Encrypted Child Percentage	ENCRYPTPERCENT	integer	142
Estimated Extracted File Size	ESTEXTRACTSIZE	integer	59
Expiration Time	EXPIRETIME	time	147
Expression (variable)	EXPRESSION	variable	89

CSV Field Name	Field Token	Type	Property Number
Extension Mismatch	EXTMISMATCH	integer	145
Extraction Licensing Size	EXTRACTLICSIZE	integer	57
Family ID	FAMILYID	string	47
Family Item Number	FAMITEMNUM	integer	85
File Entropy	FILEENTROPY	integer	110
File Entropy Delta	FILEENTROPYDELTA	integer	143
Filename	FILENAME	string	3
File Similarity	FILESIMILARITY	integer	144
File Type	FILETYPE	string	10
File Type Changed	FILETYPECHANGED	string	150
File Type Display Name	FILETYPEDISP	string	90
Filter Metadata	FILTMETA	string	66
Filter Version	FILTVER	string	65
Flags	FLAGS	integer	51
From: Domains	FROMDOMAINS	string	97
Immediate Parent File	IMMEDPARENTFID	integer	84
Index As	IXPATH	string	135
Indexed Owner	INDEXEDOWNER	string	17
Indexed User	INDEXEDUSER	string	20
Index End Time	INDEXENDTIME	time	154
Index Start Time	INDEXSTARTTIME	time	153
Ingestion Licensing Size	INGESTLICSIZE	integer	56
Internet Message ID	MESSAGEID	string	50
Job UUID	JOBUUID	string	155
Label Creation Time	LABELCTIME	time	49
Last Modifier	LASTMODIFIER	string	8
Malware Discovered Time	MALWAREDISCTIME	time	160
Malware Name	MALWARENAME	string	157
Media Class	MEDIACLASS	string	46

CSV Field Name	Field Token	Type	Property Number
Modification Time	MTIME	time	5
Nonarchivable Reason	NONARCHREASON	string	109
Non-Email Document Creation Time	DOCDOCCTIME	time	77
Non-Email Document Modification Time	DOCDOCMTIME	time	79
Nonextractable Reason	NONEXTRREASON	string	108
Originating Server	ORIGSERVER	string	148
Owner	OWNER	string	15
Owner Display Name	OWNERDISPLAY	string	16
Owner Group Membership	OWNERINS	string	95
Parent FIDs	PARENTFIDS	string	152
Path	PATH	string	2
Path ID	PATHID	integer	42
Point-In-Time State	PITSTATE	string	146
Productions	PRODUCTIONS	string	63
Production Signature	PRODSIG	string	71
PST Entry ID	PSTENTRYID	string	43
Query Server File	FID	integer	31
Query Server Folder	FOLDER	integer	33
Query Server Message	MESSAGE	integer	32
Query Server Node	NODE	integer	27
Query Server Segment	SEGMENT	integer	28
Query Server Segment UUID	SEGMENTUUID	string	60
Query Start Time	QSTARTTIME	time	91
Rank	RANK	integer	0
Regular Expression Expansion	REGEX	string	96
Relevance	RELEVANCE	integer	1
Scope	SCOPE	string	92
Scope ID	SCOPEID	string	39

CSV Field Name	Field Token	Type	Property Number
Sender	SENDER	string	102
Sequence Number	SEQUENCENUMBER	string	48
Size	SIZE	integer	4
Special Patterns	PATTERNS	string	35
Status Message Aborted	STMABORTED	string	118
Status Message Cancelled	STMCANCELLED	string	121
Status Message Corrupt	STMCORRUPT	string	113
Status Message Disabled	STMDISABLED	string	119
Status Message Encrypted	STMENCRYPT	string	115
Status Message Internal Error	STMINTERNALERR	string	116
Status Message Missing	STMMISSING	string	120
Status Message Unsupported	STMUNSUPPORTED	string	114
Status Message Warning	STMWARNING	string	117
Sticky Comments	STICKYCOMMENTS	string	133
Sticky Label	STICKYLABEL	string	44
Subroot Folder	SUBROOT	integer	149
Supplemental Signature	SUPSIG	string	80
Tags	TAGS	string	62
Tag Set Time	TAGTIME	time	159
Tape Manager Database Tape IDs	TAPEIDS	string	129
Title	TITLE	string	11
To:	TO	string	13
To: Domains	TODOMAINS	string	98
Trusted Files	TRUSTED	integer	178
User	USER	string	18
User Display Name	USERDISPLAY	string	19
Verbose Flags	VERBOSEFLAGS	string	53
Version	VERSION	string	9
Views	VIEWS	string	68

CSV Field Name	Field Token	Type	Property Number
Volume Block Counts	VOLNBLKS	string	138
Volume Head Signatures	VOLHEADSIGS	string	136
Volume Paths	VOLUMEPATHS	string	134
Volume Tail Signatures	VOLTAILSIGS	string	137

Appendix B. User Expression Fields

The `user_expression` and `user_detail_expression` fields of the format-string associated with the REPORTS and REPORTSCSV formats are the same types of expressions used by the WHERE: query operator.



Note: For details, please see the WHERE: operator in the *Index Engines Query Operators Guide*.

When specified as part of the format-string associated with the CSVFILE format, the expression is prefaced by `U:`, i.e., `89(U:user-expression)`, for historical reasons.

EXAMPLE 1

To concatenate the folder name and filename in the output:

```
PATH + FILENAME
```

EXAMPLE 2

To compile a list of all unique email domains among the `To: Domains`, `Cc: Domains` and `Bcc: Domains` fields in the output:

```
uniq(csvsplit(TODOMAINS) + csvsplit(CCDOMAINS) + csvsplit(BCCDOMAINS))
```



Note: In the *Index Engines Query Operators Guide*, all of the WHERE operator examples evaluate to a number, which is interpreted as a truth or falsehood. In reports and CSV files, this is not required. The expression can represent a number, time, string, or a set of strings.

Hitachi Vantara

Corporate Headquarters 2535 Augustine Drive
Santa Clara, CA 95054 USA www.HitachiVantara.com community.HitachiVantara.com

Regional Contact Information

Americas: +1 866 374 5822 or info@hitachivantara.com

Europe, Middle East and Africa: +44 (0) 1753 618000 or info.emea@hitachivantara.com

Asia Pacific: +852 3189 7900 or info.marketing.apac@hitachivantara.com

