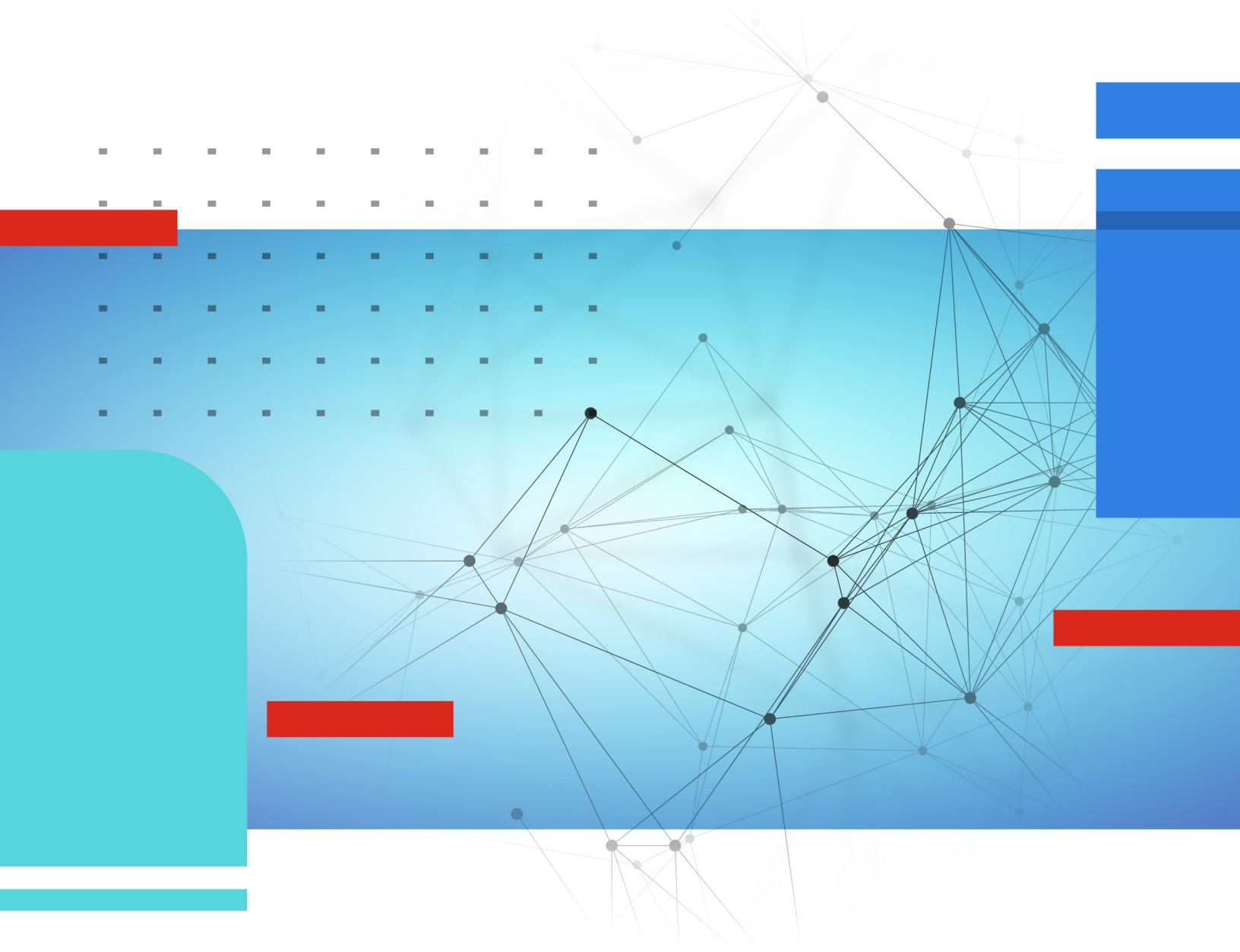
SQL Query Documentation

FortiAnalyzer 7.6.0



FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO LIBRARY
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/training-certification

FORTINET TRAINING INSTITUTE
https://training.fortinet.com

FORTIGUARD LABS
https://www.fortiguard.com

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

July 29, 2024
FortiAnalyzer 7.6.0 SQL Query Documentation
02-760-1055831-20240729

TABLE OF CONTENTS

Change Log

Date	Change Description
2024-07-29	Initial release.

Introduction

Datasets define what data is extracted from the database and represented in a report's chart. To create a report based on log messages in the local database, you can use either the predefined datasets or create your own custom dataset by querying the log message in the SQL database on the FortiAnalyzer.

While FortiAnalyzer provides pre-defined datasets that address the most common queries, you must understand Structured Query Language, also known as SQL, in order to effectively modify those datasets or create your own.

This document describes how to create and modify datasets in FortiAnalyzer. There is some information to explain SQL as it applies to the datasets, but it is not comprehensive.

FortiAnalyzer supports local ClickHouse SQL databases for the storage of log tables.

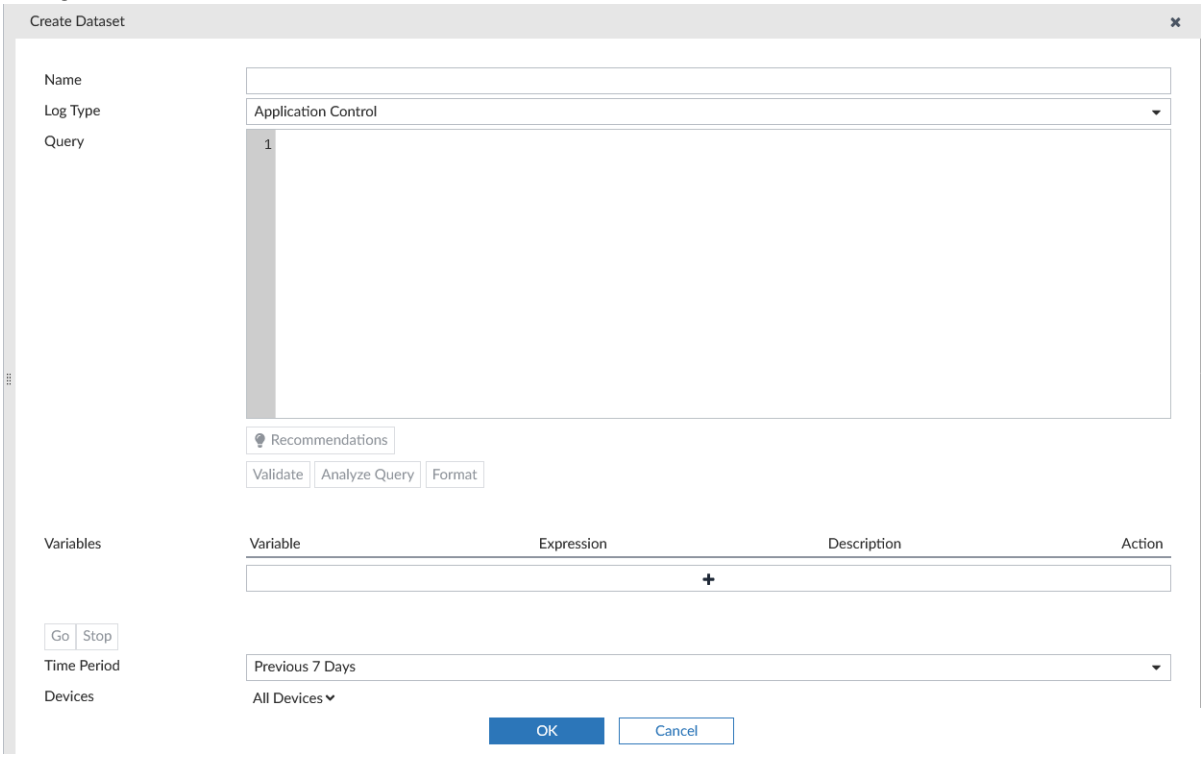
Creating datasets

The following procedure describes how to create datasets in FortiAnalyzer.

For additional details, see the FortiAnalyzer Administration Guide in the Fortinet Docs Library.

To create a custom dataset:

1. Go to *Reports* > *Report Definitions* > *Datasets*.
2. Click *Create New*.
3. Configure the dataset.



Name	Enter a name for the dataset.
Log Type	Select the log type to be used in the dataset. `$log` is used in the SQL query to represent the log type you select, and it is run against all tables of this type. If you are creating the dataset in a Fabric type ADOM, you must choose the log type under the appropriate device type. For example, FortiGate or FortiMail.
Query	Enter the SQL query used for the dataset. An easy way to build a custom query is to clone and edit a predefined dataset.

	While entering the SQL query in this field, automatic suggestions display a list of possible commands, table names, log fields, and more to use in your query where applicable. You can also mouse over related areas in the *Query* field to view the log fields available in the table. For example, you can mouse over `from $log` to view the log fields in that table.
Recommendations	Check for similar predefined dataset queries. The *Dataset Query Recommendations* pane displays the datasets in order of rating by similarity. You can expand the recommended dataset queries and *Copy* or *Format* the query, as needed. For an example, see Dataset recommendations on page 10.
Validate	Validate the entered SQL query. If any errors are present in the query, the details of the error are displayed below the query. If there are no errors in the query, the message will display *OK*. For an example, see Dataset validation on page 8.
Analyze Query	Perform a detailed analysis on the SQL query. *Analyze Query* displays the original SQL query, the transformed SQL query (if applicable), and the SQL validation results. For an example, see Analyzing a query on page 9. This function also allows users to view the hcache query that is used when a report using this dataset has enabled the auto-cache option for faster report generation. For more information on hcache, see Hcache on page 18.
Format	Automatically format the entered SQL query, making it easier to read, update, and detect errors.
Variables	Click the + to add variable, expression, and description information. If added, the expression for the variable will be used when configuring filters for reports that use this dataset. For example, if *Variable = User (user)* and *Expression = coalesce(nullifna(`user`), ipstr(`srcip`))*, then the expression will be used when *User (user)* is selected as the *Log Field* in a report's filter. See Filtering report output in the *FortiAnalyzer Administration Guide*.
Go	Test the SQL query before saving the dataset configuration. Click *Stop* to end a test in progress.
Time Period	From the dropdown, select a time period to run the SQL query against. When selecting *Custom*, enter the start date and time, and the end date and time.
Devices	From the dropdown, select devices to run the SQL query against.

The SQL dataset test function (*Go*) can be used to determine if any errors are present in the SQL format. It should not be used to test returned values as those may be different than the ones used in reports.

4. Click *OK* to save.

To add a dataset to a chart:

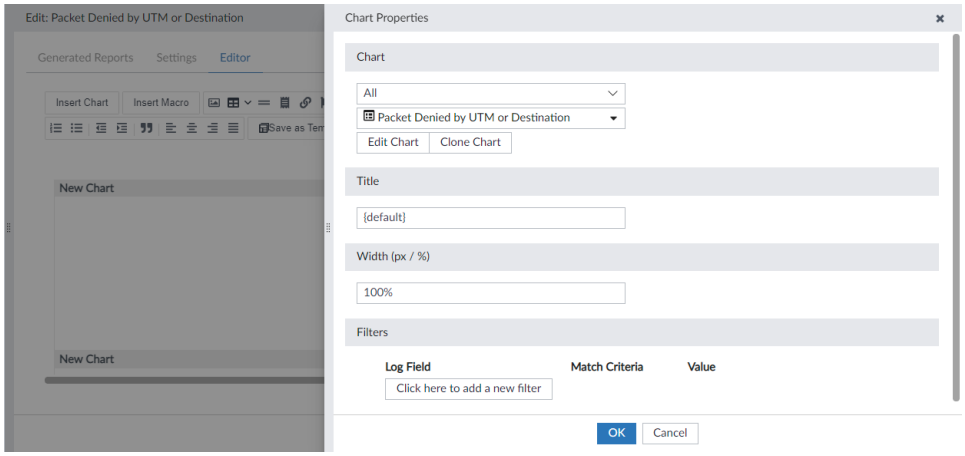
1. Go to *Reports* > *Report Definitions* > *Chart Library*, and click *Create New* or edit an existing chart.
2. From the *Dataset* dropdown, select your custom dataset.
3. Configure the remaining chart details, and click *OK*.
   The chart based on your custom dataset is now available for use in reports.

To use a chart in reports:

1. Go to *Reports* > *Report Definitions* > *All Reports*.
2. From the *Report* dropdown, click *Create New*.
   You can also edit an existing report to add the chart.
3. In the *Create Report* dialog, configure the options and click *OK*.
4.

Name	Enter a name for the report.
Create from	Select *Blank* or *Template*. If *Template* is selected, select a template for the report from the *Select Template* dropdown.
Save to Folder	From the dropdown, select a folder to save the report in. You can click the *Add* button to save the report to multiple locations.

5. Go to the *Editor* tab for the report, and click *Insert Chart*.
6. From the *Chart* dropdown, select the chart that you created.
7. Click *OK*.



8. Configure the remaining report settings, and save your changes.

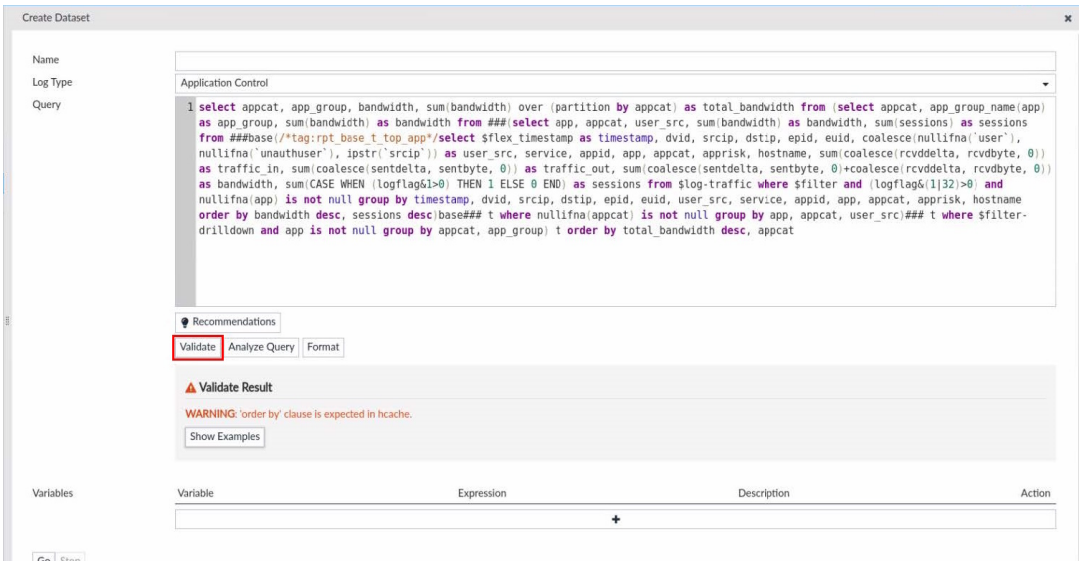
For more information, about creating charts and creating reports, see the FortiAnalyzer Administration Guide.

Dataset validation

Once a dataset has been created, you can validate the dataset query to confirm it works as intended.

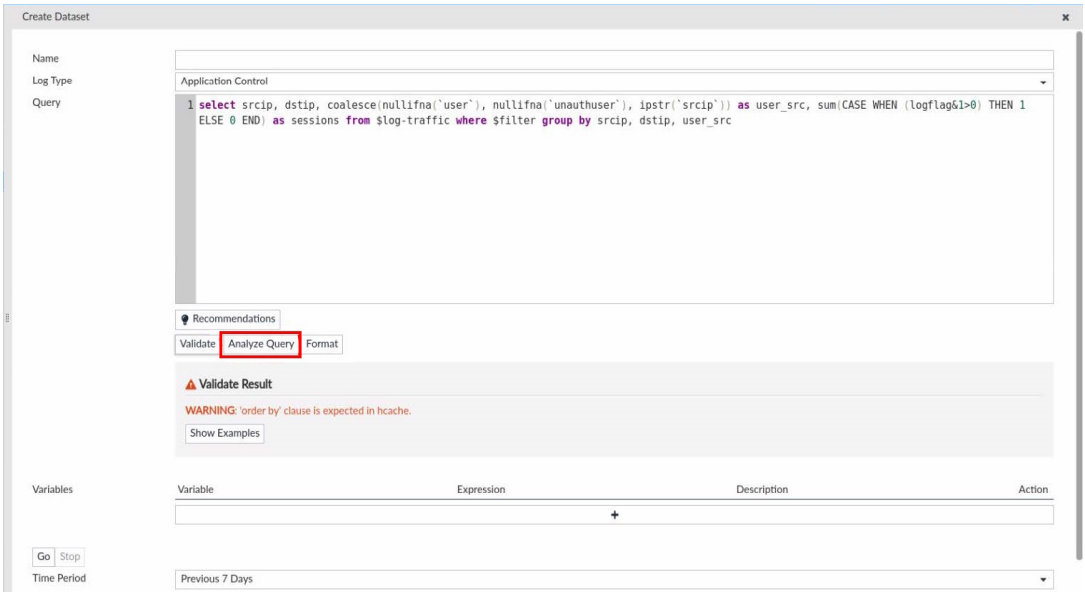
When creating or editing a dataset, click *Validate* to validate the dataset query.

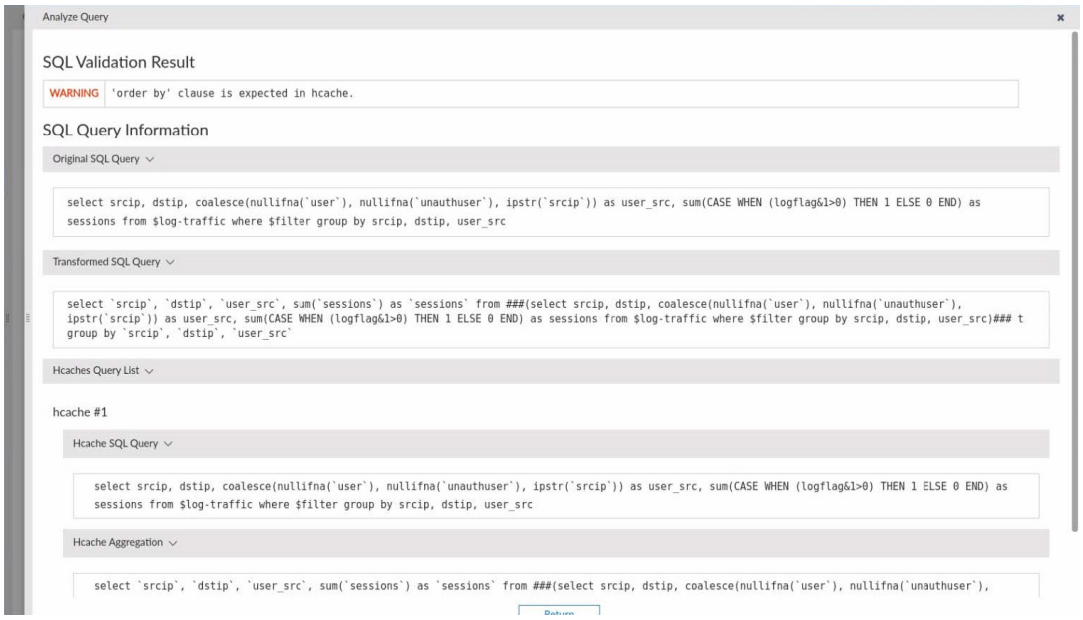
If there is any warning or error message in the validation result, click *Show Examples* to review query examples on how to correct the issues.



Analyzing a query

When creating or editing a dataset, click *Analyze Query* to review details of the dataset query.



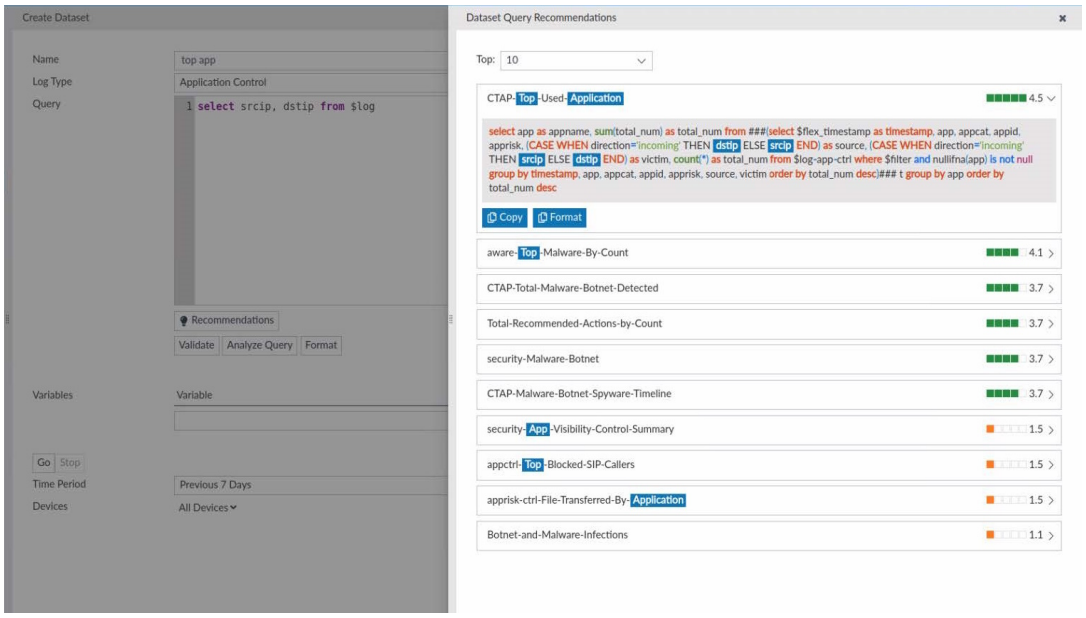


SQL Validation Result	The validation results of the query.
SQL Query Information	
Original Query	The original input query.
Transformed SQL Query	The whole dataset query after transformation. If the original query has hcache signs "`### (`" and "`) ###`", then no transformation is needed for the original query. If the original query has a log macro (for example, `$log`, `$log-traffic`) and does not have hcache signs "`### (`" and "`) ###`", the whole original query will be the hcache query. The report engine will automatically add hcache signs and the rest of the transformed query outside the hcache part for this dataset.
Hcache Query List	Every hcache query in this dataset will be listed.
Hcache SQL Query	The hcache query that is used for creating hcache tables.
Hcache Aggregation	The query used in hcache aggregation.

Dataset recommendations

When creating or editing a dataset, click *Recommendations* to check similar predefined dataset queries.

The query does not have to be complete or validated. FortiAnalyzer will try to extract log fields and function names from the input query string to find similar predefined datasets. If the dataset *Name* is not empty, it will be considered in the process as well. The matched keywords from your input and the predefined datasets will be highlighted in the results.



Troubleshooting SQL test queries

When creating or editing a dataset, click *Go* to test the query. This will display any errors that are present in the SQL query.

You can click *Format* to automatically format the entered SQL query in a dataset, making it easier to read, update, and detect errors.

To test a dataset query:

1. Follow the procedures in Creating datasets on page 6 or clone an existing dataset.
2. From the *Time Period* dropdown, select a time period to use for the test.
3. From the *Devices* dropdown, select the devices to use for the test.
4. Click *Go*.
   Once the test is finished, the query results are displayed.
5. Click *OK*.

If the SQL test is unsuccessful, an error message is displayed to indicate the cause of the problem in the query.

Following are some example error messages and possible causes:

```
ERROR: syntax error at or near...
```

- Check that SQL keywords are spelled correctly and that the query is well-formed. Table and column names are demarked by grave accent (`) characters. Single (') and double (") quotation marks will cause an error.

```
No Data
```

- The query is correctly formed, but no data has been logged for the specified log type. Check that you have configured and authorized a logging device of that type on the FortiAnalyzer.

Macros

Macros are translated to expressions in the query. For example, `${REPORT_SESSION}` is converted to `(logflag&1>0)`. Note that this macro is only available as of FortiAnalyzer 7.4.2.

`logflag&1>0` means to filter out other logs than report sessions; `logflag&(1|32)>0` means to keep report sessions as well as long live sessions. You can use `${REPORT_SESSION}` or `${REPORT_SESSION_WITH_LONGLIVE}` instead of the `logflag&1` or `32`.

More logflag related macros are as follows:

```
'${FGTLOG_F_REPORT_SESSION}': '1',
'${FGTLOG_F_BLOCKED_ACTION}': '2',
'${FGTLOG_F_CLOUD_APP}' : '4',
'${FGTLOG_F_FCT_SYSUSR}' : '8',
'${FGTLOG_F_BOTNET}' :'16',
'${FGTLOG_F_LONGLIVE_SESSION}' :'32',
'${FGTLOG_F_DNS_QNAME}' :'64',
'${REPORT_SESSION}': '''(logflag&${FGTLOG_F_REPORT_SESSION}>0)''',
'${REPORT_BLOCK_SESSION}': '''(logflag&(${FGTLOG_F_REPORT_SESSION}|${FGTLOG_F_BLOCKED_
     ACTION})=(${FGTL OG_F_REPORT_SESSION}|${FGTLOG_F_BLOCKED_ACTION}))''',
'${BLOCKED_ACTION}': '''(logflag&${FGTLOG_F_BLOCKED_ACTION}>0)''',
'${IS_CLOUDAPP}': '''(logflag&${FGTLOG_F_CLOUD_APP}>0)''',
'${IS_FCT_ENDUSER}': '''(logflag IS NULL OR logflag&${FGTLOG_F_FCT_SYSUSR}=0)''',
'${IS_BOTNET}': '''(logflag&${FGTLOG_F_BOTNET}>0)''',
'${REPORT_SESSION_WITH_LONGLIVE}': '''(logflag&(${FGTLOG_F_REPORT_SESSION}|${FGTLOG_F_
     LONGLIVE_SESSION})>0)''',
'${DNS_QNAME}': '''(logflag&${FGTLOG_F_DNS_QNAME}>0)''',
```

SQL tables

SQL is the database language that FortiAnalyzer uses for logging and reporting. Log data is inserted into the SQL database for log view and report generation. FortiAnalyzer uses a ClickHouse SQL database.

In an SQL database all information is represented as tables, and each table consists of a set of rows and columns. There are two types of tables:

- User tables, which contain information that is in the database, and
- System tables, which contain the database description.

Once a log table is matured, FortiAnalyzer will add a timestamp to the table number, and then start running hcache queries. When the table matures depends on the platforms. Most tables will be rolled around five million rows. If the table size is too big, it will cause issues such as out of memory. There is one table called "`table_ref`" to store the log table info.

FortiAnalyzer has master tables and child tables. A child table has "`ALLELSE`" in its name, and its master table can be found in column "`tbl_master`" in the table "`table_ref`".

You can use information from SQL tables to create custom datasets for use in report charts.

For example, below are some of the available tables:

`$ADOM_ENDPOINT` and `$ADOM_EPEU_DEVMAP`	FortiAnalyzer will learn UEBA for logs, and try to identify endpoints, endusers, device mappings.
`$ADOMTBL_PLHD_AUDIT_HST`	Used for security rating data.
`devtable_ext`	Used for device id mapping in the interface table.
`intfinfo`	Used for FortiGate interface data from RESTAPI.

Log types and subtypes

Log types each have a SQL table that can be specified when creating datasets. The available log types are visible when selecting the *Log Type* for the dataset.

Log types also include log subtypes, which are types of log messages that are within the main log type. For more information on log types and subtypes, see the FortiAnalyzer and FortiGate Log Message Reference guides on the Fortinet Document Library.

If you are combining data from multiple log types in a custom dataset, you must use the appropriate type name(s) in the SQL statement.

Log types available in FortiAnalyzer datasets

Device type	Log type	Name in SQL
FortiGate	Application Control	$log-app-ctrl
	Intrusion Prevention	$log-attack
	Content	$log-content
	Data Leak Prevention	$log-dlp
	DNS	$log-dns
	Email Filter	$log-emailfilter
	Event	$log-event
	File Filter	$log-file-filter
	GTP	$log-gtp
	Vulnerability Scan	$log-netscan
	Protocol	$log-protocol
	SSH	$log-ssh
	SSL	$log-ssl
	Traffic	$log-traffic
	Antivirus	$log-virus
	VoIP	$log-voip
	Web Application Firewall	$log-waf
	Web Filter	$log-webfilter
	Local Event	$log-local-event
FortiMail	Email Filter	$log-emailfilter
	Event	$log-event
	History	$log-history
	Antivirus	$log-virus
FortiAnalyzer	Application Control	$log-appevent
	Event	$log-event
FortiWeb	Attack	$log-attack
	Event	$log-event
	Traffic	$log-traffic

Device type	Log type	Name in SQL
FortiCache	Application Control	$log-app-ctrl
	Intrusion Prevention	$log-ips
	Content	$log-content
	Data Leak Prevention	$log-dlp
	Event	$log-event
	Vulnerability Scan	$log-netscan
	Traffic	$log-traffic
	Antivirus	$log-virus
	VoIP	$log-voip
	Web Filter	$log-webfilter
FortiClient	FortiClient System Event	$log-fct-event
	FortiClient Security Event	$log-fct-netscan
	FortiClient Traffic	$log-fct-traffic
Syslog	Syslog	
FortiManager	Application Control	$log-appevent
	Event	$log-event
FortiSandbox	Event	$log-event
	Vulnerability Scan	$log-netscan
	Antivirus	$log-virus
FortiDDoS	Intrusion Prevention	$log-ips
	Event	$log-event
FortiAuthenticator	Event	$log-event

Device type	Log type	Name in SQL
FortiProxy	Application Control	$log-app-ctrl
	Intrusion Prevention	$log-ips
	Data Leak Prevention	$log-dlp
	DNS	$log-dns
	Email Filter	$log-emailfilter
	Event	$log-event
	File Filter	$log-file-filter
	Protocol	$log-protocol
	SSH	$log-ssh
	SSL	$log-ssl
	Traffic	$log-traffic
	Antivirus	$log-virus
	VoIP	$log-voip
	Web Application Firewall	$log-waf
	Web Filter	$log-webfilter
FortiNAC	Asset	$log-asset
	Event	$log-event
FortiFirewall	DNS	$log-dns
	Event	$log-event
	File Filter	$log-file-filter
	GTP	$log-gtp
	SSH	$log-ssh
	SSL	$log-ssl
	Traffic	$log-traffic
FortiSOAR	Event	$log-event
FortiADC	Intrusion Prevention	$log-ips
	Event	$log-event
	Traffic	$log-traffic
FortiDeceptor	Event	$log-event

Device type	Log type	Name in SQL
FortiNDR	Attack	$log-attack
	Event	$log-event
	NDR	$log-netscan
FortiIsolator	Event	$log-event
	Traffic	$log-traffic
FortiEDR	Event	$log-event
FortiPAM	Data Leak Prevention	$log-dlp
	Event	$log-event
	Protocol	$log-protocol
	Secret	$log-security
	SSH	$log-ssh
	Traffic	$log-traffic
	Antivirus	$log-virus
FortiCASB	Data Leak Prevention	$log-dlp
FortiToken	Event	$log-event
Fabric	Normalized	

Log severity levels

Each log entry contains a level field that indicates the estimated severity of the event that caused the log entry. When a logging severity level is defined, the FortiAnalyzer unit logs all messages at and above the selected severity level. For example, if you select Error, FortiAnalyzer logs Error, Critical, Alert, and Emergency level messages.

The Debug log severity level is rarely used. Debug log messages are useful when the FortiAnalyzer unit is not functioning properly. Debug log messages are only generated if the log severity level is set to Debug. Debug log messages are generated by all subtypes of the event log.

To view information about log severity levels, see the FortiAnalyzer Log Message Reference.

Log fields

Each log table stored in an SQL database contains log fields that can be used in datasets.

You can view a full list of the fields available for each log type in the *FortiAnalyzer ClickHouse Schema* file available from the FortiCloud Support page.

Hcache

The hcache is a proprietary FortiAnalyzer report caching system. The hcache runs Report/FortiView queries on the matured DB log tables in advance and stores the results in the DB “hcache tables”. When a report or FortiView is running, it will search the pre-built hcache results first instead of running the whole dataset query based on the log tables. This significantly decreases the time for running queries.

In datasets, “`###(`“ and “`)###`” are the start and end hcache tags; the query between them is called an hcache query. These are used for creating hcache tables. The query outside the three hash signs will use the results in the hcache table.

The hcache tags are optional. If a dataset query does not use three hashes but has a log table macro, "`$log`", the whole query will be an hcache query. In this case, the report engine will automatically add three hashes and the rest of the query outside the hcache query. The whole query can be found by clicking *Analyze* when creating or editing the dataset.

Hcache base queries are between "`###base(`" and "`)base###`". Some hcache queries share the same hcache base queries.

Both Fortiview and dataset queries may have `/*tag:rpt_base_t_top_app*/` tags. Those tags are for naming the hcache, for example, “`rpt_base_t_top_app`” is a report top-application hcache base on traffic logs.

Filter-drilldown

The filter-drilldown, "`where $filter-drilldown`", comes outside the hcache query. When the filter values vary, the hcache query will not be affected, so the hcache table does not need to be rebuilt. Normally, you do not need to add the filter-drilldown to the query as it’s automatically added by the report engine where applicable.

`Group by` and `order by` clauses

Due to performance concerns, the number of rows in hcache tables are limited. An hcache table will be trimmed if its length exceeds the maximum limit. `Group by` and `order by` clauses are needed in the hcache query to aggregate and select the significant results.

Note that "`/*skipSTART*/`" and "`/*SkipEND*/`" are usually used around the `order by` clauses. The auto-cache engine will monitor if the data trimming happens while running the hcache. If there is no data trimming, then the `order by` is not necessary for the hcache. In this case, the `order by` clause inside the tags will be removed for better query performance.

The `/*fabricStart*/` and `/*fabricEnd*/` are for FortiAnalyzer fabric cases in the supervisor.

Grouping by `itime` or `dtime`

If a result is grouped by "`itime`" or "`dtime`", the important results may be trimmed and lost.

The `dtime` is the device time when FortiGate creates logs, and the `itime` is the timestamp when logs come to FortiAnalyzer.

If `dtime` does not exist, FortiAnalyzer will add it into logs. In fact, there is a special handling on FortiAnalyzer `dtime`:

`dtime` = the timestamp (epoch time) when the log was created + the timezone offset in seconds.

In such a way, it’s easy to get the FortiGate log local time with `from_dtime(dtime)` in SQL.

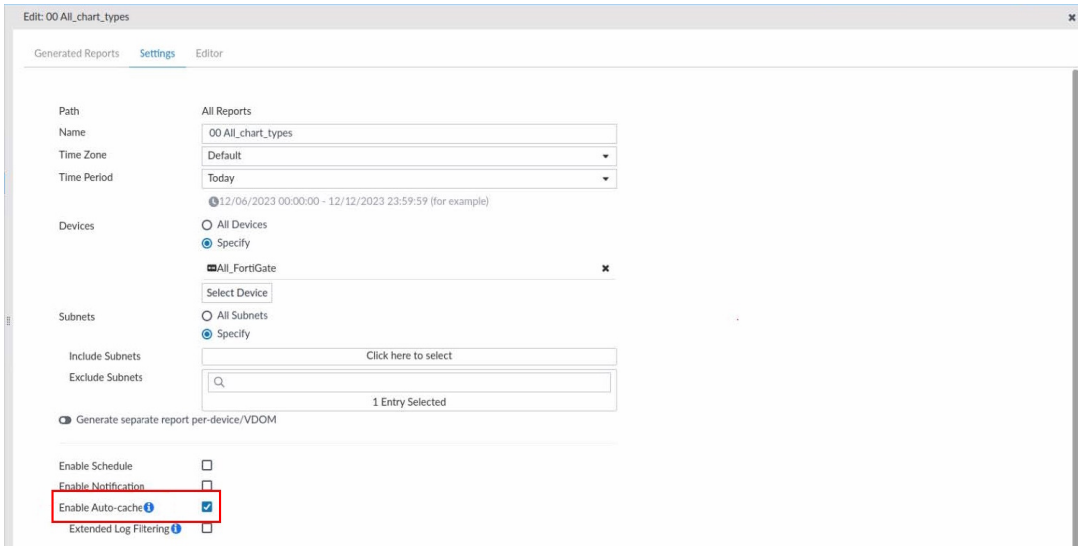
As there are often FortiGates from various time zones registered in FortiAnalyzer, the best practice is to use `itime`, which is in UTC timestamp and timezone independent. The `dtime` is mainly used when showing FortiGate local time or calculating time duration; for example, `max(dtime) - min(dtime)`.

Regarding time accuracy, as the default FortiGate config to send logs is near real-time, the `itime` timestamp created by FortiAnalyzer when it receives FortiGate logs is acceptable as opposed to the timestamp when FortiGate sends the log. When there is a very strict requirement on time accuracy, we can use `eventtime`, which is in nanoseconds when the FortiGate log is generated.

The "`$flex_timescale(timestamp)`" will give data points based on report range. For example, one data point per hour for a seven-day report, and one data point per five minutes for a one-day report.

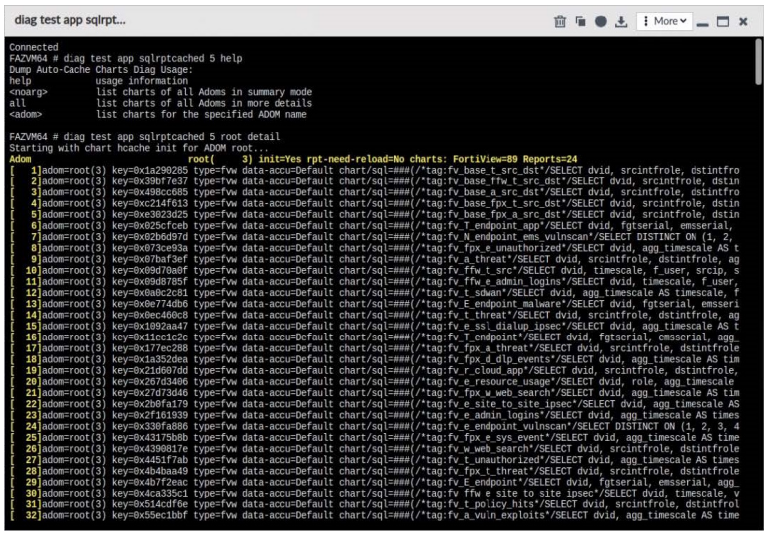
Grouping hcache queries

For each report, you can use the *Enable Auto-cache* setting to reduce the time for generating reports. All the charts and datasets of this report will be auto-cached, and their hcache tables will be created automatically when new logs come in and log tables are matured.



Details of auto-cached reports/charts/datasets can be found using the following CLI command:

```
diagnose test app sqlrptcached 5
```

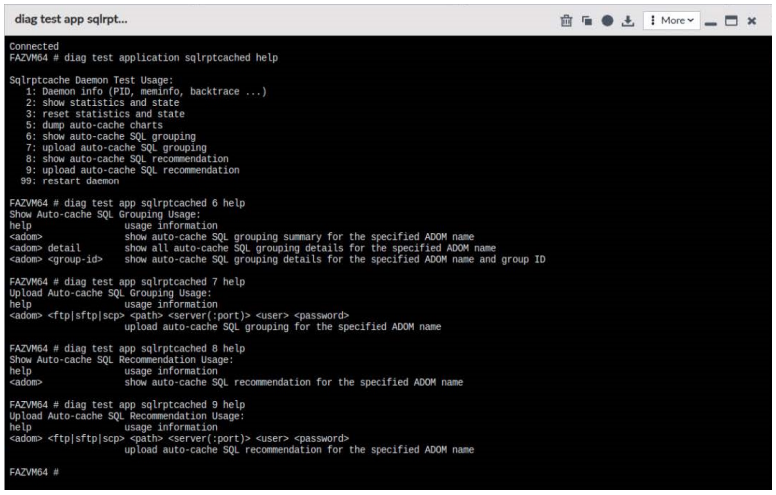


Too many auto-cache tasks will take a lot of resources and affect the performance of FortiAnalyzer. Ideally, we would like to combine similar hcache queries into one query, so fewer hcache tables need to be created for the auto-cached reports. Datasets that contain those hcache queries need to be adjusted accordingly.

The following CLI commands will show the results of grouping hcache queries in auto-cached datasets:

```
diagnose test application sqlrptcached 6
diagnose test application sqlrptcached 7
diagnose test application sqlrptcached 8
diagnose test application sqlrptcached 9
```

These CLI commands were added as of FortiAnalyzer 7.4.2.



Show grouping results

The following CLI command will show the grouping results of hcache queries from auto-cached datasets:

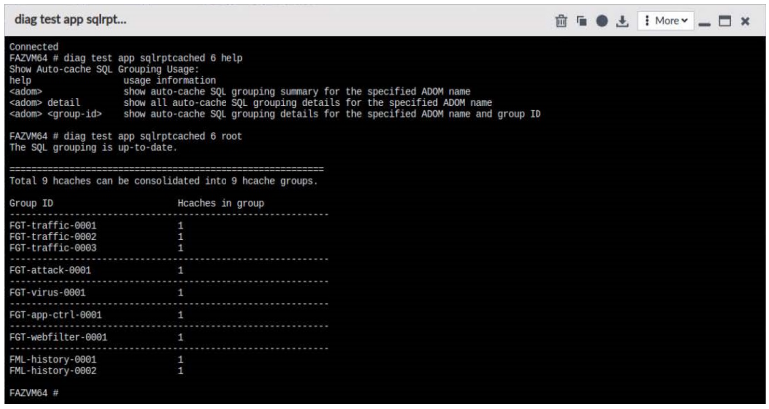
```
diagnose test application sqlrptcached 6
```

Use `help` to show the usage information for the command:

```
diagnose test application sqlrptcached 6 help
```

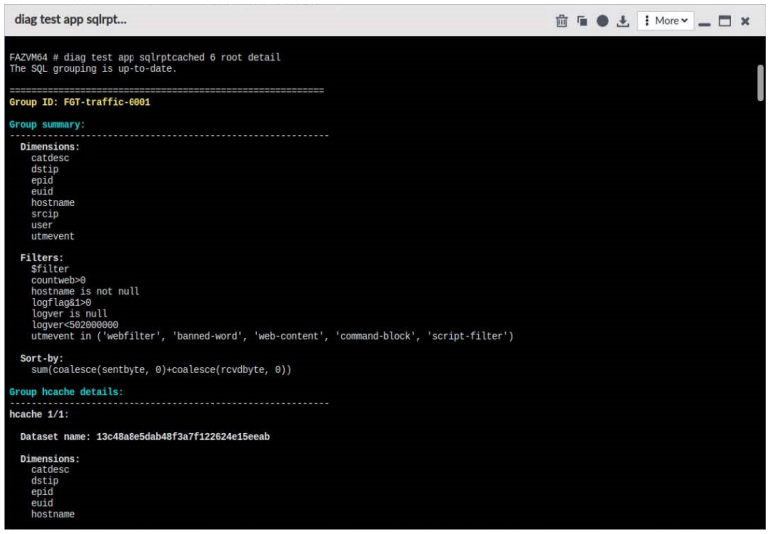
Include an ADOM name to show auto-cache SQL grouping summary for the specified ADOM:

```
diagnose test application sqlrptcached 6 <adom>
```



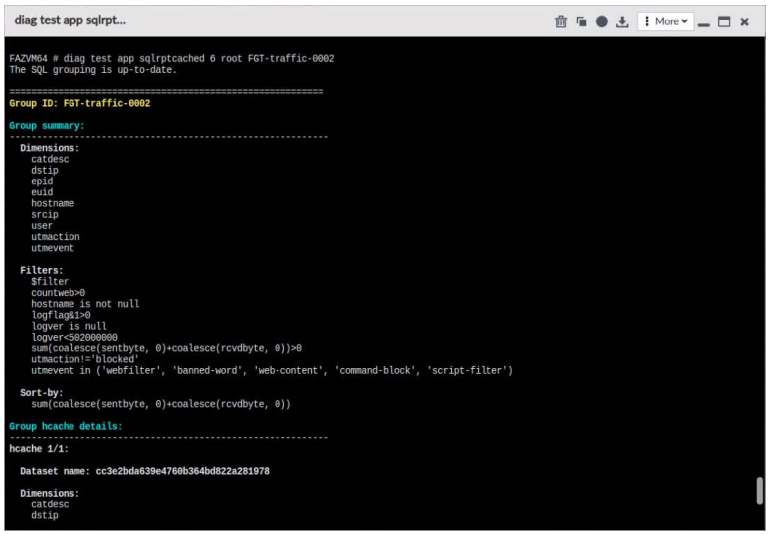
Use `detail` to show all auto-cache SQL grouping details for the specified ADOM name:

```
diagnose test application sqlrptcached 6 <adom> detail
```



Include a group ID to show auto-cache SQL grouping details for the specified ADOM name and group ID:

```
diagnose test application sqlrptcached 6 <adom> <group-id>
```



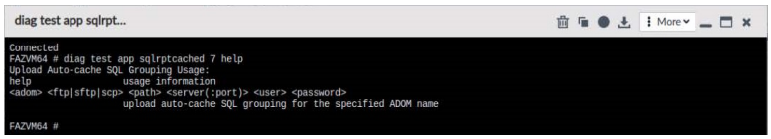
Upload grouping results

The following CLI command will upload the grouping results of hcache queries to an ftp/sftp/scp server:

```
diagnose test application sqlrptcached 7
```

Use `help` to show the usage information for the command:

```
diagnose test application sqlrptcached 7 help
```



Show recommendation results

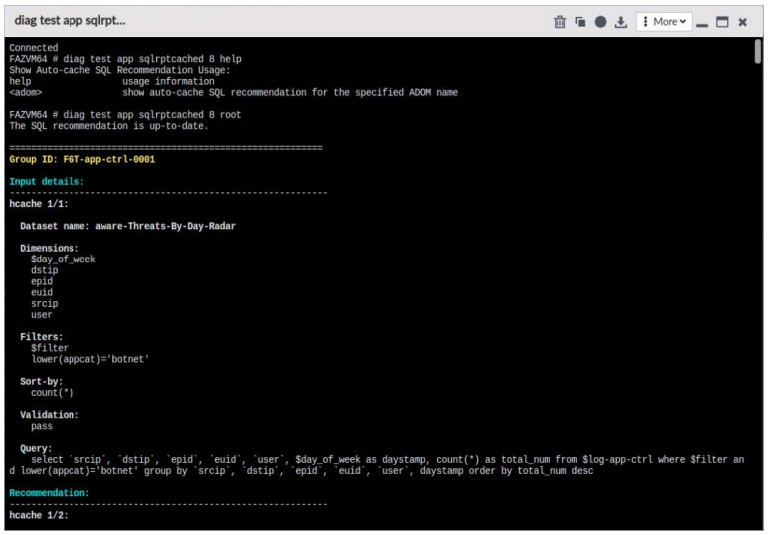
The following CLI command will show the grouping results of hcache queries from both auto-cached datasets and predefined datasets. In each group, there will be `Input details` which are hcache queries from auto-cached datasets.

```
diagnose test application sqlrptcached 8
```

The `Recommendation` queries are from predefined datasets, so you can check if your auto-cached hcache queries can be replaced by one of the predefined hcache queries.

Use `help` to show the usage information for the command:

```
diagnose test application sqlrptcached 8 help
```



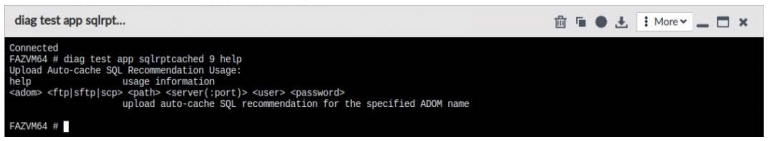
Upload recommendation results

The following CLI command will upload the recommendation results of hcache query groups to a ftp/sftp/scp server:

```
diagnose test application sqlrptcached 9
```

Use `help` to show the usage information for the command:

```
diagnose test application sqlrptcached 9 help
```



Examples of datasets

For examples, you can review the SQL queries for predefined datasets in the FortiAnalyzer Dataset Reference. These datasets can be cloned and edited in the FortiAnalyzer GUI to create custom datasets.

After you create the datasets, you can use them when you configure charts under *Reports* > *Report Definitions* > *Chart Library*. Configured charts can be selected when creating or modifying reports. For more information on creating charts and reports, see the FortiAnalyzer Administration Guide.

www.fortinet.com

Copyright© 2024 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's Chief Legal Officer, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.