Delinea

Server Suite

Auditing Find Sessions Guide

Version: 2024.x

Publication Date: 7/26/2024

Server Suite Auditing Find Sessions Guide

Version: 2024.x, Publication Date: 7/26/2024

Table of Contents

Using Find Sessions

Find Sessions is a separate executable file, installed in the same directory as Audit Analyzer, that you can use to find and open audited sessions. The program provides a graphical user interface and a command line interface for specifying the search criteria. You can use either interface to find sessions of interest. From the Find Sessions graphical user interface, you can also replay, update the review status, view the desktops used for any sessions found, display the list of indexed commands or events, and copy the session URI.

Starting Find Sessions

You can start Find Sessions from the Windows command line, using a web browser, or by selecting the View DirectAudit Sessions menu option in other applications, such as Access Manager and Active Directory Users and Computers.

For example, in Access Manager or Active Directory Users and Computers, you can select a computer or user, right-click, then select View DirectAudit Sessions to open Find Sessions. To start Find Sessions from the Windows command line, you can navigate to the Audit Analyzer installation directory and run the following command in a command prompt window:

```
findsessions /ia
```

Find Sessions Return Codes

For your reference, Find Sessions supports the following return codes to report the status of an operation performed:

This code	Indicates this result
0	The operation was successful.
1	The operation failed because Find Sessions could not parse the Session URI.
2	The operation failed because Find Sessions could not parse the user input.
3	The operation failed because Quick queries are not supported.
4	The operation failed because there were errors in the AQL format.
5	The operation failed because of an incompatible version of AQL was detected.
6	The operation failed because no installation was selected.
7	The operation failed because the installation specified was not found.

This code	Indicates this result
8	The operation failed because the AQL string contains the <group by> keyword.
9	The operation failed because no sessions were selected.
10	The operation failed because Find Sessions could not export the list of events.
11	The operation failed because Find Sessions could not export the session list.
12	The operation failed because Find Sessions could not export UNIX input or output.
13	Not all selected sessions were deleted.

Specifying the Sessions to Find

After you start Find Sessions by selecting View DirectAudit Sessions, from the Windows command line, or in a web browser, the program displays a graphical user interface for selecting search criteria. You can use the Common or Advanced search criteria to find sessions of interest. The Find Sessions dialog box then displays the results that match the criteria you specify. You can then replay, update the review status, display the list of indexed commands or events, copy session URI, or view the desktops used in any of the sessions returned.

In most cases, you can find the sessions you are interested in through some combination of user name, computer name, and session time displayed on the Common tab. If you right-click to View DirectAudit Sessions from a specific computer or user, that computer or user is automatically defined as the search criteria. If you want to specify additional criteria, such as review status or auditor name, you can click the Advanced tab.

To specify criteria by which to find sessions:

1. Start Find Sessions.
2. Select the desired installation from the Installation list.
3. On the Common tab, enter the basic search criteria as applicable for the sessions you want to find:
   a. User: Type all or part of the user name to find sessions for a particular user account.
   b. Machine: Type all or part of the computer name to find sessions run on a particular computer.
   c. Session start time: Select this option to find sessions based on when the session started. If you select this option, you can refine thesearch to include sessions started or not started in a specific numberof days, hours, or minutes, or to include sessions started or notstarted today, yesterday, this week, last week, this month, last month, this year, or last year.
4. Click **Find Now** to find the sessions that match the criteria you specified.
5. Click **Clear All** to start a new query.

Specifying Advanced Criteria

In some cases, you might want to specify additional criteria for a search or to search exclusively on an attribute not found on the Common tab. For example, you might want to find only those sessions that have yet to be reviewed or

all of the sessions where a specific command or application was used. To add criteria or perform these types of specialized searches, you can click the **Advanced** tab.

To Specify Advanced Criteria for Finding Sessions:

1. Start **Find Sessions**.
2. Select the desired installation from the **Installation list**.
3. Click the **Advanced** tab.
4. Click **Add** to add a new criterion.
5. Select an appropriate attribute from the Attribute list based on the sessions you want to find.

   For example: You can search for sessions based on the period of time in which they were active or based on a specific state. You can also search for sessions based on the activity that took place during the session. For example, you can find sessions where specific UNIX commands or Windows applications were used.
6. Select the appropriate criteria for the attribute you selected, then click **OK**.
7. The specific selections you can make depend on the attribute selected. For example, if the attribute is Review Status, you can choose Equals and the review state you want to find. If you select the attribute Comment, you can specify Contains any of and type the string that you want to find any part of.
8. When searching for user names or computers on the **Advanced** tab, use the Starts with option. If you use the default to match exactly, you must include the fully qualified domain name of the user or computer.
9. Click **Add** to add another criterion until you have defined all of the attributes for which you want to find sessions.
10. Click **Find Now** to find the sessions that match the criteria you specified.
11. Click **Clear All** to start a new query.

Adding Advanced Criteria

If you have more than one advanced criteria, different criteria attributes, such as Session Time and State, are separated by an implicit AND operation. Onl sessions that match both criteria are returned. If you have repeated criteria attributes, for example, time is not in past 10 days; time is in last month, the attributes are separated by an implicit OR operation. Sessions that match either criteria are returned.

Editing and Removing Advanced Criteria

You can edit and remove any of the advanced criteria you specify in Find Sessions. For example, if you are not finding the appropriate sessions, you might need to change or remove the criteria you have defined.

To Edit or Remove Find Sessions Criteria:

1. Start Find Sessions.
2. Select the desired installation from the Installation list.
3. Click the **Advanced** tab.
4. Select the criterion in the list of Define Criteria.
5. Click **Edit** to modify the definition or **Remove** to remove the criterion.

Finding Sessions From a Command Line

You can run Find Sessions as a command line utility on computers where Audit Analyzer is installed. The command line interface can be useful, for example, if you may want to find, export, or delete sessions as part of a script.

You can view usage information for the command line interface using the /help option.

To Use the Command Line Interface for Find Sessions:

1. Open a Command window and navigate to the Audit Analyzer directory.

   ```
   cd "C:\\Program Files\\Centrify\\Audit\\AuditAnalyzer"
   ```

2. Run the findsessions command with the /help option to view usage information.

   ```
   findsessions /help
   ```

3. Specify search criteria for finding sessions using the following format:

   ```
   findsessions /i="*InstallationName*" /u="*username*" /m="*computerName*"/t="*yyyy-MM-dd HH:mm:ss*"
   ```

   The installation name is required. You must also specify at least one of the other criteria (user name, computer name, or time). You can also combine the search criteria to refine your search.

   For user name and computer name, you can specify a portion of a name to find all sessions matching that name portion. For time, if you specify a date without a time, the assumed time is 12 midnight. For example, if you do the following search and you have sessions on computers named "KH-Win7" and "KH-W8," the results include sessions for both computers.

   ```
   FindSessions /i="DefaultInstallation" /m="KH-W"
   ```

   The following example finds sessions for "Admin" and "Administrator" users:

   ```
   FindSessions /i="DefaultInstallation" /u="Admin"
   ```

   The following example finds sessions that were running at a specific time regardless of what time the sessions started or ended:

   ```
   FindSessions /i="DefaultInstallation" /t="2015-01-21 5:25:00"
   ```

   You can also find sessions for multiple users or computers by separating the user names or computer names using a semi-colon (;).For example, to search for audited sessions for the users maya and fred, you can specify both users in the command line like this:

   ```
   FindSessions /i="DefaultInstallation" /u="maya;fred"
   ```

   For more complex queries, you can also use AQL syntax on the command line.

   For details, see Finding sessions using AQL syntax.

Find Sessions Command Line Usage Examples

You can view usage information for the command line interface using the /help option. That information is included here as well.

Usage:

FindSessions.exe [Connection] [Query] [Action] [Parameter]

Connection:
/i=<installation name> or /installation=<installation name>
Make a connection to the specified DirectAudit Installation.

Query:
Query can be defined by AQL or individual search criteria

/a=<aql statement> or /aql=<aql statement>
Use the specified AQL as a search criteria to find the audited sessions from DirectAudit databases.
This option should not be used together with /user, /machine or /activetime.

/u=<user name> or /user=<user name>
Find all audited sessions for a particular user from DirectAudit databases.
This option can be used together with /machine and /activetime, which means the returned sessions need to fulfill all specified criteria.
This option should not be used with /aql option.

/m=<machine name> or /machine=<machine name>
Find all audited sessions for a particular machine from DirectAudit databases.

This option can be used together with /user and /activetime, which means the returned sessions need to fulfill all specified criteria. This option should not be used with /aql option/t=<time> or /activetime=<time>.

/t=<time> or /activetime=<time>
Find all active audited sessions at a particular time from DirectAudit databases. This option can be used together with /user and /machine, which means the returned sessions need
to fulfill all specified criteria. This option should not be used with the /aql option.

/r="role1;role2" or /role="role1;role2"
Find all sessions with role role1 OR role2. Must be used with /export="UnixCommandZZ_BAR_ZZUnixInputZZ_BAR_ZZUnixInputOutput". If /role and /ticket are used together, sessions meeting role AND ticket criteria are searched.

/k="ticket1;ticket2" or /ticket="ticket1"
Find all sessions with trouble ticket ticket1 OR ticket2. Must be used with /export="UnixCommandZZ_BAR_ZZUnixInputZZ_BAR_ZZUnixInputOutput". If /role and /ticket are used together, sessions meeting role AND ticket criteria are searched.

Action:
/delete
Delete the sessions by the query.

/export=[SessionListZZ_BAR_ZZWashEventsZZ_BAR_ZZUnixCommandZZ_BAR_ZZUnixInputZZ_BAR_ZZUnixInputOutput]
Export the sessions by the query. This option should used with /path option.

Parameter:
/path
Folder to save the export files. This option should used with /export option

/format=[htmlZZ_BAR_ZZhtmZZ_BAR_ZZcsvZZ_BAR_ZZpdfZZ_BAR_ZZxml]
Export the session list. this option should used with /export=SessionInfo /path=<folder path>

/suppresswarning
Suppress warning messages.

/onerror=[continue]
Continue processing session list if one or more databases are unreachable.

Examples:

FindSessions /installation="installation sample" /aql="1 time is in this week"

FindSessions /installation="installation sample" /aql="1 inputcommand = \"dzdo*\"\"" /delete

FindSessions /installation="installation sample" /aql="1 text = \"dzdo\"\"" /suppresswarning /export="UnixInput" /path="folder path"

FindSessions /installation="installation sample" /user="user sample" /machine="machine sample" /activetime="2011-12-24 15:30:45"

FindSessions /installation="installation sample" /aql="1 module = \"Windows PowerShell\"\"" /export="SessionList" /format="html" /path="folder path"

FindSessions /installation="installation sample" /aql="1 time is in this month"

/export="UnixInputOutput" /path="folder path" /role="role1;role2" /ticket="ticket1;ticket2"

If the last field that you're search for includes double quotes, you need to escape the quotes. For example, findsessions -i”MyInstallation” /aql”1 time is in this week” doesn't have this issue but FindSessions /i="MyInstallation" /a="1 sessionid = \"a4006f206465-4db1-a2e7-a4e1f646c835\"\" does.

Finding sessions using AQL syntax

If you are an experienced programmer and want to write complex queries, you can use AQL statements on the command line.

To use AQL to find sessions at the command line:

1. Open a command window and navigate to the Audit Analyzer directory.

   cd “C:\Program Files\Centrify\Audit\AuditAnalyzer”

2. Run the findsessions command with the following syntax:

   FindSessions /i=”*InstallationName*” /aql=”*AQL query text*”

   For example, the following is a simple query that searches for sessions that were running in the current week:

   findsessions -i=”MyInstallation” /aql=”1 time is in this week”

   To find a specific session using the session identifier, you might write a query similar to the following:

   FindSessions /i="MyInstallation" /a="1 sessionid = \"a4006f206465-4db1-a2e7-a4e1f646c835\"\"

   To find a specific session using the user display name, you might write a query similar to the following:

   findsessions /i="installationname" /a="1 displayname=\"maya*\"\"

When you enter a search term, AQL looks for an exact match. To search for sessions that start with the term you entered, add an asterisk to the search term. For example, user=\\"maya*\\"\\" finds sessions for users such as maya@acme.com, mayan@acme.com, and so forth. Otherwise, a search for user=\\"maya\\"\\" returns nothing and a search for user=\\"maya@acme.com\\"\\" returns sessions for just that one user.

If the last field that you're search for includes double quotes, you need to escape the quotes. For example, findsessions \-i=”MyInstallation” /aql=”1 time is in this week” doesn't have this issue but FindSessions /i="MyInstallation" /a="1 sessionid = \\"a4006f206465-4db1-a2e7-a4e1f646c835\\"\\" does.

Note:

Note:

Simplifying AQL queries

Writing valid AQL queries for the command line can be challenging. The basic format for AQL statements in Backus-Naur notation consists of the following parts:

<aql> ::= <version> {<quick_terms>} ZZ_BAR_ZZ {<type> ZZ_BAR_ZZ <filter>}

To simplify the process of generating the AQL queries you want to use on the command line, you can use Audit Analyzer to create a new private query and use the user interface to specify the query criteria. After you have created the query, you can right-click the query node, and click **Export Query Definition** to save the query definition as a file. You can then extract the AQL statement from the query definition. You can then delete the private query node from Audit Analyzer if it is not needed.

For example, run the command with the definition from the private query:

findsessions -i=”MyInstallation” /aql="1 type= shellui, wingui; time is in this week; review = Reviewed"

Audit Query Language overview

You can use the Audit Query Language to search for audited sessions with Find Sessions from a command line interface.

The Audit Query Language (AQL) serves two purposes:

- **Query definition:** The Audit Management Server database stores the query definition as an AQL statement.
- **Query language:** In order to query for audit information, the audit & monitoring service sends AQL statements to the Audit Management Server database.

When you enter an AQL query, the system stores this as the query definition. The query definition defines what information is of interest and how to group the results. In some cases, you might retrieve the results over multiple phases, depending on how you want to present the information.

For example: The query "get all Windows audit sessions, grouped by user" has two phases:

1. Gather a list of all users who have Windows audit sessions
2. Show all the Windows sessions for each user who is listed in Step 1.

In each phase, the Audit & Monitoring generates the AQL statement and sends it to the Audit Management Server database in order to query for audit information. This part is when the AQL statements function as a query language.

Here is an AQL statement example:

1 Type=wingui; orderby=time DESC; time is in this week; user=”joe*”,”mark*”;machine=”domaincontroller”’

The example query would return audited Windows sessions in the last week where Joe or Mark logged in to the domain controller system, and the results would be listed in descending order of when they occurred.

In general, the format of an AQL statement can either be just some quick search terms or a statement with the following parts:

- Audit trail types
- Group-by
- Order-by
- Predicates

Backus-Naur Form (BNF) definition of AQL

Here is the Backus-Naur Form (BNF) definition of the AQL language syntax so that you can see how the query language is constructed.

<aql> ::= <version> {<quick_terms>} ZZ_BAR_ZZ {<type> ZZ_BAR_ZZ <groupby> ZZ_BAR_ZZ <filter>}

<version> ::= any numeric number. Currently, we support only 1.

<quick_terms> ::= <word> ZZ_BAR_ZZ <and_words> ZZ_BAR_ZZ <or_words> ZZ_BAR_ZZ <exact_combined_words>

<and_words> ::= <word> (" " <word>)+

<or_words> ::= <word> ("OR" <word>)+

<exact_combined_words> ::= """ <and_words> """

<word> ::= any printable string except white spaces

<type> ::= "type" ("=" | ":") <typename> {"," <typename>}

<groupby> ::= "groupby" ("=" | ":") <groupname> {"," <groupname>}

<orderby> ::= “orderby” (“=”ZZ_BAR_ZZ”:”) <columnname> {“ASC”ZZ_BAR_ZZ”DESC”}

<filter> ::= <normal_filter> | <negative_filter>

<normal_filter> ::= <string_filter> ZZ_BAR_ZZ <time_filter> ZZ_BAR_ZZ <number_filter> ZZ_BAR_ZZ <enum_filter> ZZ_BAR_ZZ

<boolean_filter> | <ip_filter>

<negative_filter> :: = "not(" <normal_filter> ")"

<string_filter> :: = <string_field> <string_op> """ <string_val> """ {"," """ <string_val> """}

<string_op> ::= "=" | "!="

<time_filter> ::= <single_time_filter> ZZ_BAR_ZZ <between_time_filter> ZZ_BAR_ZZ <in_predefined_time_filter> |

<in_past_filter>

<single_time_filter> ::= <time_field> <single_time_op> <single_time_val> <!-- single_time_val is in

format of "yyyy-mm-dd hh:mm:ss" -->

<time_op> :: = "is before" ZZ_BAR_ZZ "is after" ZZ_BAR_ZZ "is not before" ZZ_BAR_ZZ "is not after"

<between_time_filter> ::= <time_field> <between_time_op> <between_time_val>

<between_time_op> ::= "is between" | "is not between"

<between_time_val> ::= <single_time_val> " and " <single_time_val>

<in_predefined_time_filter> ::= <time_field> <in_predefined_time_op> <predefined_time_val>

<in_predefined_time_op> ::= "is in" | "is not in"

<predefined_time_val> ::= "today" ZZ_BAR_ZZ "yesterday" ZZ_BAR_ZZ "this week" ZZ_BAR_ZZ "last week" ZZ_
BAR_ZZ "this month" ZZ_BAR_ZZ "last month" ZZ_BAR_ZZ

"this year" | "last year"

<in_past_filter> ::= <time_field> <in_past_op> <digit>+ <unit_of_time>

<in_past_op> ::= "is in_past" | "is not in_past"

<unit_of_time> ::= "day" ZZ_BAR_ZZ "hour" ZZ_BAR_ZZ "minute"

Note:Currently, AQL has filters only for strings and time.

AQL usage examples

AQL usage examples

/i

/installation

/aql

/a

/user

/u

/machine

/m

/activetime/

/t

/suppresswarning

/sw

/delete

/export

/r

/role

/path

/format

/ticket

/k

Find Sessions Usage formats:

FindSessions /i=<installationName> /a=<AQL query>

FindSessions /i=<installationName> /u=<user or semi-colon-separated list of users> /m=<machine or semi-colon-separated list of machines> /t=<YYYY-MM-DD HH:MM:SS>

FindSessions /i=<installationName> /a=<AQL query> /{delete}

FindSessions /i=<installationName> /a=<AQL query> /export=<SessionList> /format=<htmlZZ_BAR_ZZhtmZZ_BAR_ZZcsvZZ_BAR_ZZpdfZZ_BAR_ZZxml> /sw /path=<folderPath>

FindSessions /i=<installationName> /a=<AQL query> /export=<SessionListZZ_BAR_ZZWashEventsZZ_BAR_ZZUnixCommandZZ_BAR_ZZUnixInputZZ_BAR_ZZUnixInputOutput> /sw /path=<folderPath>

FindSessions /i=<installationName> /a=<AQL query> /export=<UnixCommand|UnixInput\UnixInputOutput> /sw /path=<folderpath> /r="role1;role2" /"ticket1;ticket2"

Find Sessions: Usage examples without AQL:

FindSessions /i="DirectAudit" /user="user sample" /machine="machine sample" /activetime="2018-12-24 15:30:45"

FindSessions /i="DirectAudit" /user="maya;fred" /machine="KH-Win7;KH-Win8" /activetime="2018-12-24 15:30:45"

Find Sessions: Usage examples with AQL:

FindSessions /i="DirectAudit" /aql="1 time is in this week"

FindSessions /i="DirectAudit" /aql="1 module = \"Windows PowerShell\"\"" /delete

FindSessions /i="DirectAudit" /aql="1 text=\"dzdo\"\"" /export="UnixCommand" /path="folder path"

FindSessions /i="DirectAudit" /aql="1 inputcommand = \"dzdo*\"\"" /suppresswarning /export="UnixInputOutput" /path="folder path"

FindSessions /i="DirectAudit" /aql="1 sessionid=\"D108F7B2-F4FB-FB42-A6E7-A40454780690\"\"

AQL quick search terms

ou can just enter a series of keywords if you just want to do a quick search.

Here are some examples:

- joe : any data fields that contain the word ‘joe’
- joe john : any data fields that contain both words ‘joe’ and ‘john’

- joe OR john : any data fields that contain ‘joe’ or ‘john’
- "joe john" : any data fields that contain the exact phrase “joe john”

The default operator designated by a space between terms is evaluated as an “AND” operator, so there is no need to include “AND” between terms. Explicit operator takes precedence over implicit operator, thus “OR” is always evaluated before the absence of an operator.

Here are some examples of quick search term queries in AQL:

The database searches the following data fields with the keywords in quick search terms:

- User (username)
- Machine (machine name)
- Time (audit trail data record start time)
- Module
- Text

AQL audit trail types example

AQL audit trail types example

If desired, you can refer to the types of audit trails to include in the results, such as Windows sessions or UNIX sessions. You can specify one or more audit trail types with the "type:" parameter.

If you don't include this parameter, the results include all audit trail types.

If you specify more than audit trail type, the results are those that fit all specified parameters.

Example:

type=wingui, shellui

AQL group-by example

If desired, you can specify how to group the results.

Example:

groupby=user, date

When you specify multiple groupby criteria, the database groups the results by the first criterion and displays the immediate result by ignoring the remaining criteria. When you double-click the results, then the database displays more results according to the remaining criteria.

FindSessions does not support the use of groupby.

AQL order-by example

If desired, you can specify how to sort the AQL query results by using orderby.

For example:

orderby=time, user ASC, machine DESC

The sort order options are as follows:

- ASC: sort results in ascending order
- DESC: sort results in descending order

If you don't specify a sort order, the system uses ‘ASC’ by default.

AQL Predicates

Using predicates in your AQL query is entirely optional. You can filter the result set by any number of predicates or none at all. Each predicate expresses a condition that must be true in order for the service to include a record in the result set.

There is an implicit ‘AND’ between each predicate. If you repeat a predicate for a field, there is an implicit 'OR' between them.

Each predicate refers to a field in the schema of an audit trail type. If the field name does not specify an audit trail type, then the field must exist for all selected audit trail types. If the field name specifies an audit trail type specified with the "type:" parameter, then the predicate applies only to that audit trail type.

AQL predicate behavior examples

Example A: Type=wingui, shellui; user = “joe”

The above example selects all Windows and UNIX sessions for joe.

Example B: Type=wingui, shellui; shellui.user = “joe”

The above example selects all Windows sessions but only UNIX sessions for joe.

The service categorizes predicates according to the field data type:

- String
- Number
- Boolean
- Date / time
- IP
- Enumeration

AQL string predicate behavior

Here are some examples of how to filter an AQL query based on string predicates:

AQL query	Filter behavior
field = "<string>",”<string2>”,…	exact match
field != "<string>", “<string2>”,…	not equals (exact match)
field = "<string>*", “<string2>*”,…	starts with
field != "<string>*", “<string2>*”,…	not starts with

AQL number predicate behavior

Here are some examples of how to filter an AQL query based on number predicates:

AQL query	Filter behavior
field = <number>	equals
field != <number>	not equal
field >= <number>	greater than or equal
field > <number>	greater than
field <= <number>	smaller than or equal
field < <number>	smaller than

You can replace <number> with any integer or floating point number, such as 1 or -3.14.

AQL Boolean predicate behavior

Here are some examples of how to filter an AQL query based on boolean predicates:

AQL query	Filter behavior
field = true	true
field != false>	false

AQL Date and time predicate behavior

Here are some examples of how to filter an AQL query based on date and time predicates:

AQL query	Filter behavior
field is (not) before <datetime>	before a specific date and time or not
field is (not) after <datetime>	after a specific date and time or not
field is (not) between <datetime> <datetime>	between two dates and time or not
field is (not) in_past <number> <unit>	in the past period of time or not, where the unit is day, hour, or minute
field is (not) in <predefined time>	field is not in the predefined time or not

Replace <datetime> with a particular date and time with the following format:

- Y-M-D, for example 2019-12-15
- Y-M-D h:m:s, for example 2019-12-15 15:30:00

Replace <predefined time> with any of the following values:

- today
- yesterday
- this week
- last week
- this month
- last month
- this year
- last year

AQL IP predicate behavior

Here are some examples of how to filter an AQL query based on IP address predicates:

AQL query	Filter behavior
field = <ip>	equals
field != <ip>	not equal
field >= <ip>	greater than or equal
field > <ip>	greater than
field <= <ip>	smaller than or equal
field < <ip>	smaller than

AQL enumeration predicate behavior

Here are some examples of how to filter an AQL query based on enum predicates:

AQL query	Filter behavior
field = <enum>	equals
field != <enum>	not equal

Replace <enum> with the values appropriate for the field you're querying against.

For example, filtering for a session state involves specifying an enum value:

state = Terminated

state != InProgress

AQL keywords

Keyword	Predicate type
Session time	Date/Time predicate
UNIX command time	Date/Time predicate
State	Enum predicate: Unknown, InProgress, Terminated, Disconnected, Completed, ToBeDeleted
Review status	Enum predicate: None, ToBeReviewed, Reviewed, PendingForAction, KeepForever, ToBeDeleted
Session size	Numeric predicate (in kilobytes)
Unix outputs and commands	String predicate
User	String predicate
Machine	String predicate
Auditstore	Number predicate
Parameters of commands and applications	String predicate
Unix command name	String predicate
Windows applications	String predicate
Comment	String predicate
Session Id	String predicate
Client name	String predicate
User display name	String predicate
Account	String predicate
Text	String predicate
Module	String predicate
Tag	String predicate

AQL usage examples

/i

/installation

/aql

/a

/user

/u

/machine

/m

/activetime/

/t

/suppresswarning

/sw

/delete

/export

/r

/role

/path

/format

/ticket

/k

Find Sessions Usage formats:

FindSessions /i=<installationName> /a=<AQL query>

FindSessions /i=<installationName> /u=<user or semi-colon-separated list of users> /m=<machine or semi-colon-separated list of machines> /t=<YYYY-MM-DD HH:MM:SS>

FindSessions /i=<installationName> /a=<AQL query> /{delete}

FindSessions /i=<installationName> /a=<AQL query> /export=<SessionList> /format=<htmlZZ_BAR_ZZhtmZZ_BAR_ZZcsvZZ_BAR_ZZpdfZZ_BAR_ZZxml> /sw /path=<folderPath>

FindSessions /i=<installationName> /a=<AQL query> /export=<SessionListZZ_BAR_ZZWashEventsZZ_BAR_ZZUnixCommandZZ_BAR_ZZUnixInputZZ_BAR_ZZUnixInputOutput> /sw /path=<folderPath>

FindSessions /i=<installationName> /a=<AQL query> /export=<UnixCommand|UnixInput\UnixInputOutput> /sw /path=<folderpath> /r="role1;role2" /"ticket1;ticket2"

Find Sessions: Usage examples without AQL:

FindSessions /i="DirectAudit" /user="user sample" /machine="machine sample" /activetime="2018-12-24 15:30:45"

FindSessions /i="DirectAudit" /user="maya;fred" /machine="KH-Win7;KH-Win8" /activetime="2018-12-24 15:30:45"

Find Sessions: Usage examples with AQL:

FindSessions /i="DirectAudit" /aql="1 time is in this week"

FindSessions /i="DirectAudit" /aql="1 module = \"Windows PowerShell\"\"" /delete

FindSessions /i="DirectAudit" /aql="1 text=\"dzdo\"\"" /export="UnixCommand" /path="folder path"

FindSessions /i="DirectAudit" /aql="1 inputcommand = \"dzdo*\"\"" /suppresswarning /export="UnixInputOutput" /path="folder path"

FindSessions /i="DirectAudit" /aql="1 sessionid=\"D108F7B2-F4FB-FB42-A6E7-A40454780690\"\"

Accessing Sessions with a Web Browser

On computers that have Audit Analyzer installed, you can also find and play back sessions from a web browser. Because the cda:// protocol is automatically registered on the computer with Audit Analyzer, you can use a web browser to open Find Sessions or to replay a specific session. For example, you can embed a cda:// link in a web page to automatically generate a list of sessions, or you might want to embed a link to a session or set of sessions in a web-based report or event notification.

Opening Find Sessions from a web browser

You must be able to specify a query using AQL syntax to open Find Sessions from a web browser. If you want to start playing back a session from a web browser, you must know the session identifier. You can extract the session identifier from the session URI.

To start Find Sessions from a web browser:

1. Open a web browser.
2. Type the installation name and a search string using AQL syntax in the address bar of the web browser.
3. For example, if you want to search an installation named MyInstallation5 for sessions that involved the Administrator user, you would type the following in the address bar:
4. cda://MyInstallation5/?search=\"1 user=\"Administrator*\"\
5. Click **Allow** to open the Find Sessions with the Advanced tab displayed and “user=Administrator*” listed for the Define Criteria.
6. Click **Find Now** to find sessions matching the criteria you specified.

Playing back a session from a web browser

If you want to start playing back a session from a web browser, you must know the session identifier. You can extract the session identifier from the session URI.

To get the session identifier:

1. In the session player, select File > Copy Session URI.
2. Open a text editor and paste the session URI into the file.
3. Delete the portion of the URI that identifies the player and installation, so that only the object GUID remains.
4. For example, if the URI looks like this:
5. rep://myInstallation/b62bc280-678c-439a-aec3-09a9b7ee4395
6. Remove the part of the URI so that you only have the session identifier:
7. b62bc280-678c-439a-aec3-09a9b7ee4395

To play back a specific session from a web browser:

1. Open a web browser.
2. Type the installation name and session ID in the address bar of the web browser:
3. cda://*<installationName>*/*<session_id>*
4. For example:
5. cda://myInstallation/b62bc280-678c-439a-aec3-09a9b7ee4395
6. The session player opens and plays the specified session.

Exporting sessions and session data

In addition to specifying the criteria for finding sessions of interest, you can use Find Session to selectively export session data to a file. You can export the following information:

- A list of sessions matching the criteria you specify.
- An indexed list of events associated with the Windows sessions that match the criteria you specify.
- An indexed list of commands associated with the UNIX sessions that match the criteria you specify.
- The UNIX input associated with the UNIX sessions that match the criteria you specify.
- The UNIX input and output associated with the UNIX sessions that match the criteria you specify.

You specify the export operation, type of data to export, file format, and file location using the following command line options:

/export=[SessionListZZ_BAR_ZZWashEventsZZ_BAR_ZZUnixCommandZZ_BAR_ZZUnixInput
|UnixInputOutput]
/format=[htmlZZ_BAR_ZZhtmZZ_BAR_ZZcsvZZ_BAR_ZZpdfZZ_BAR_ZZxml]
/path=<folder_path>

You can use these options in combination with other criteria, such as /user or /machine, to export information for a specific user, computer, or time. You can specify the /format option used for exporting the sessions of interest. If you don’t specify the /format option, sessions matching the criteria you specify are exported as comma-separated

values (.csv) in a text file. If you are exporting Windows events, UNIX commands, UNIX input, or UNIX input and output, each session is exported as a separate file in the format you specify.

If you are exporting UNIX commands, UNIX input, or UNIX input and output, you can also use the command line options /role and /ticket to export sessions based on specific role or trouble-ticket information. Before you can use these options, however, you must configure the information required. For example, if you want to find all of the UNIX commands executed by a user running the db_backup role, you must first define and assign the db_backup role using Access Manager.

Exporting a session list

To export a list of sessions from the command line, use the following syntax:

FindSessions /i="*InstallationName*" /export="SessionList"
/format="*format*" /path="*folder*"

For example, to export the session list for all users in HTML format and save the output in the C:\Temp\Exported Sessions folder, you would type a command like this:

FindSessions /i="MyInstallation" /export="SessionList"
/format="html" /path="C:\Temp\Exported Sessions"

The command generates the list of sessions in the format specified. In this case, the command would generate an HTML file named SessionList in the C:\Temp\Exported Sessions folder with the following information for each session exported:

- User name, display name, account used, computer name, and audit store for the session.
- Start time, end time, and current state of the session.
- Client name associated with the session.
- Review status, user who last modified the review status, the time the status was last modified, and the comment added when the session was last modified.
- Size of the session in KB.
- Session URI that can be used to replay the session.

Exporting Windows events

To export an indexed event list for Window sessions from the command line, use the following syntax:

FindSessions /i="*InstallationName*" /export="WashEvents" /path="*folder*"

For example, to export the indexed event list for the sessions associated with a specific user and save the output in the C:\Temp\Session Events folder, you would type a command like this:

FindSessions /i="MyInstallation" /user="chris.howard"
/export="WashEvents" /path="C:\Temp\Session Events"

The command generates the list of events as comma-separated values in a text file. For example:

"Time","Application","Title","Type","Desktop","Audited","Role","Ticket"

"1/29/2015 1:53:14 PM", "Windows Explorer", "Start", "Application Activate", "Default", "Y","<None>","<None>"
"1/29/2015 1:53:56 PM", "DirectAuthorize System Tray", "Options", "Application Activate", "Default",

"Y","<None>","<None>"
...
"1/29/2015 3:00:51 PM", "Windows Explorer", "Start", "Window Activate", "LocalSQLAdmin",
"Y","<None>","<None>"
"1/29/2015 3:01:16 PM", "Microsoft SQL Server Management Studio Express", "Microsoft SQL Server Management Studio Express", "Application Activate", "LocalSQLAdmin", "Y","<None>","<None>"
.

Exporting UNIX command lists

To export an indexed command list for UNIX sessions from the command line, use the following syntax:

FindSessions /i="*InstallationName*" /export="UnixCommand" /path="*folder*"

For example, to export the indexed command for the sessions associated with a specific computer and save the output in the C:\Temp\UNIX folder, you would type a command like this:

FindSessions /i="MyInstallation" /machine="rhes-63"
/export="UnixCommand" /path="C:\Temp\UNIX"

The command generates the list of commands as comma-separated values in a text file. For example:

"Time","Command","Role","Ticket"
"10/9/2014 3:12:14 PM","/bin/bash ","<None>","<None>"
"10/9/2014 3:12:19 PM","adflush","<None>","<None>"
"10/9/2014 3:12:23 PM","su -","<None>","<None>"
"10/9/2014 3:12:27 PM","Password: ","<None>","<None>"
"10/9/2014 3:12:30 PM","adflush","<None>","<None>"
"10/9/2014 4:26:14 PM", "exit","<None>","<None>"

Searching for sessions by role or trouble-ticket information

When you use the /export=UnixCommand option, you can also use the command line options /role and /ticket to export sessions based on specific role or trouble-ticket information.

Use /role to specify search criteria based on one or more privilege elevation service roles. You can specify multiple roles separated by semicolons (;). For example, add /role="db_backup/zonename;mail_admin/zonename" to the command line to search for UNIX sessions that were run using the db_backup or mail_admin role.

When you search for sessions by role name, be sure to include the zone name. Otherwise, FindSessions doesn't return the sessions and instead displays the message, "No session is selected to be exported".

FindSessions /i="MyInstallation" /export="UnixCommand"
/role="db_backup/zonename;mail_admin/zonename" /path="C:\Temp\UNIX"

You can use the /ticket option to specify search criteria based on the trouble-ticket information if you have configured in the dzcheck script to collect this information. You can specify multiple tickets separated by semicolons (;). For example, add /ticket="ticket 1;ticket 2" to the command line to search for sessions ticket1 or ticket2 were specified.

You cannot use wildcards to search for role names or ticket information. If you specify both the /role and /ticket options, FindSessions returns the sessions that match both the specified roles and the specified trouble-ticket information. For information about configuring the dzcheck script and how to capture trouble-ticket information, see the *Administrator's Guide for Linux and UNIX*.

Exporting UNIX input

To export UNIX input from the command line, use the following syntax:

FindSessions /i="*InstallationName*" /export="UnixInput" /path="*folder*"

For example, to export the UNIX input for a specific user and save the output in the C:\Temp\Input folder, you would type a command like this:

FindSessions /i="MyInstallation" /user="tai-u1" /export="UnixInput" /path="C:\Temp\Input"

The command exports UNIX input to a text file. For example:

"UnixInputData","Role","Ticket"
"[1/20/2015 4:13:38 PM] K: PS1=NetShell:<CR>","<None>","<None>"
"[1/20/2015 4:13:38 PM] K: stty kill ^u erase ^h<CR>","<None>","<None>"

"[1/20/2015 4:13:38 PM] K: TERM=dumb<CR>","<None>","<None>"
"[1/20/2015 4:13:38 PM] K: set TERM=dumb<CR>","<None>","<None>"
"[1/20/2015 4:13:40 PM] K: cat /etc/passwd<CR>","<None>","<None>"
"[1/20/2015 4:13:40 PM] K: echo $?<CR>","<None>","<None>"
"[1/20/2015 4:13:40 PM] K: cat /etc/group<CR>","<None>","<None>"
"[1/20/2015 4:13:40 PM] K: echo $?<CR>","<None>","<None>"

When you use the /export=UnixInput option, you can also use the command line options /role and /ticket to export sessions based on specific role or trouble-ticket information. For details about using these options, see Using Find Session.

Exporting UNIX input and output

To export UNNIX input and output from the command line, use the following syntax:

FindSessions /i="*InstallationName*" /export="UnixInputOutput" /path="*folder*"

For example, to export UNIX input and output for a specific computer and save the output in the C:\Temp\Output folder, you would type a command like this:

FindSessions /i="MyInstallation" /m="firefly-sf" /export="UnixInputOutput" /path="C:\Temp\Output"

The command exports UNIX input and output to a text file. For example:

"UnixInputOutputData","Role","Ticket"
"[1/21/2015 10:53:20 AM] 0: /bin/bash ","<None>","<None>"
"[1/21/2015 10:53:23 AM] 1: [maya@firefly-sf Desktop]$ pwd","<None>","<None>"
"[1/21/2015 10:53:23 AM] K: pwd<CR>","<None>","<None>"
"[1/21/2015 10:53:23 AM] 2: /home/maya/Desktop","<None>","<None>"
"[1/21/2015 10:53:34 AM] 3: [maya@firefly-sf Desktop]$ cd /tmp","<None>","<None>"
"[1/21/2015 10:53:34 AM] K: cd /tmp<CR>","<None>","<None>"
"[1/21/2015 10:53:54 AM] K: ls -al in*<CR>","<None>","<None>"
"[1/21/2015 10:53:54 AM] 4: [maya@firefly-sf tmp]$ ls -al in*","<None>","<None>"

"[1/21/2015 10:53:54 AM] 5: -r-xr-xr--. 1 root root 313027 Dec 16 05:51 install.sh", "<None>","<None>"
"[1/21/2015 10:54:04 AM] K: su -<CR>","<None>","<None>"
"[1/21/2015 10:54:04 AM] 6: [maya@firefly-sf tmp]$ su -","<None>","<None>"

"[1/21/2015 10:54:10 AM] 7: Password: ","<None>","<None>"
"[1/21/2015 10:54:10 AM] K: xxxxxxxx<CR>","<None>","<None>"

When you use the /export=UnixInputOutput option, you can also use the command line options /role and /ticket to export sessions based on specific role or trouble-ticket information. For details about using these options, see Using Find Sessions.

Suppressing warning messages

By default, Find Sessions will generate warning messages if you attempt to export sessions without expected activity. For example, if you run a command to export UNIX input and output using /export="UnixInputOutput" and there is no user input activity, you might see warning messages similar to the following:

Finished exporting the sessions successfully.

Warning, URI:rep://BLD08/f435d61c-f191-4344-8adf-9d1432cb35ea,
Message: There is no user inputs captured in this session.

You can safely suppress these warning messages using the /suppresswarning or /sw command line option. For example, you might run a command similar to this:

C:\AuditAnalyzer> findsessions /i="BLD08" /role="verify"
/format=csv /path="C:\Temp" /export="UnixInputOutput"
/a="1 time is in today" /suppresswarning

This command would export the UNIX output without displaying warning messages about there being no user input.

Deleting sessions

You can also use Find Sessions to delete sessions matching the criteria you specify from the command line. You can use the /delete option in combination with other criteria, such as /user or /machine, to delete information for a specific user, computer, or time. However, if you specify the /delete on the command line, all of the sessions returned by the query are deleted.

To delete sessions from the command line, use the following syntax:

FindSessions /i="*InstallationName*" /delete

For example, to delete the sessions for a specific user on a specific computer, you would type a command like this:

FindSessions /i="MyInstallation" /user="tai-u1"
/machine="rhes63" /delete

Note that you cannot use the /delete option in combination with the /export option. If you want to export session information before deleting, you must do so in two separate operations.

Sample script for deleting multiple sessions

You can use Find Sessions to delete multiple sessions manually from the command line or using Windows Task Scheduler to automate the task. However, if you are deleting multiple sessions at once, you might want to execute the command from a batch file to ensure that Find Sessions will wait for the operation to complete and return the result of the operation.

The following is a sample script to delete sessions from TestInstallation recorded in the current month.

```
----------Start of FindSessions_Delete.bat----------
 @ECHO OFF
 cd "C:\Program Files\Centrify\Audit\AuditAnalyzer"
 Start /WAIT FindSessions.exe /i="TestInstallation" /a="1 time is in this month" /delete
 if ERRORLEVEL 1 (goto FindSessionError)
 goto Succeeded
 :FindSessionError
 echo ########################################################################

echo ## FindSession execution failed. ErrorLevel: %ERRORLEVEL% ##
 echo ########################################################################

goto exit
 :Succeeded
 echo FindSession execution succeeded.
 :exit
 ----------End of FindSessions_Delete.bat----------
```

You can use a similar batch file if you want to export multiple sessions at the same time. To write a script for exporting information, you would specify the type of information to export and the path for saving the exported output. For example, if you want to export UNIX commands for MyInstallation to the C:\UNIX folder, the script could include a command like this:

```
Start /WAIT FindSessions.exe /i="MyInstallation"
 /export=”UnixCommand” /path=”C:\UNIX”
```