HP Operations Analytics

For the Linux operating system

Software Version: 2.10

HP Operations Analytics Help

Document Release Date: May 2014

Software Release Date: May 2014

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Microsoft, Windows, and Windows NT are U.S. registered trademarks of the Microsoft group of companies.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: **http://h20230.www2.hp.com/selfsolve/manuals**

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to: **http://h20229.www2.hp.com/passport-registration.html**

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at: **http://www.hp.com/go/hpsoftwaresupport**

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

HP Software Solutions Now accesses the HPSW Solution and Integration Portal Web site. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this Web site is **http://h20230.www2.hp.com/sc/solutions/index.jsp**

Contents

Chapter 1: Getting Started with HP Operations Analytics

About

What can OpsA do for me?

Welcome to Operations Analytics, an analysis tool that provides a unified approach to proactively manage and solve simple and complex IT operations problems.

In today's complex data center environments, the source of a problem is not always easy to detect using traditional management and troubleshooting tools that look only for pre-determined solutions to known potential problems. For example, many management and troubleshooting tools are designed to provide analytics for a specific problem context, such as root cause isolation, outlier detection, and service level agreement violation. They provide these services by using a specific data set and analytics technique.

With Operations Analytics you generate insights from the data in your IT environment that you choose to collect. And because identifying the most useful analytics to derive from the data generally depends on the problem context, with Operations Analytics you, the user, provide each data request in the form of a search query.

Operations Analytics enables you to use simple search queries using the Phrased Query Language (PQL) to view metric, topology, event, and log file information related to the context you specify. Operations Analytics also enables you to use its Analytics Query Language (AQL) for more precise searches; for example, when you know the exact log file message or combination of analytics required to troubleshoot a problem.

When entering a search query, Operations Analytics offers suggestions as you type. It then uses your query to analyze the information available and displays the most important and related metrics.

Operations Analytics processes data according to your search query. These results assist you with the following kinds of tasks:

- Identify and analyze the pattern of problems in your IT environment.
- Identify the cause of resource or application usage problems.
- Troubleshoot server and network performance problems.
- Identify configuration or inventory changes.

What are OpsA's main features?

- **Dashboards.** OpsA allows you to create your own dashboard or to use one of the out-of-the-box dashboards. Dashboards are collections of Query Panes, which display specific metrics in your choice of visual representations. The dashboards can also display the log viewer. For more details, see "Dashboards and Query Panes" on page 13.

- **Search.** To use Operations Analytics, you must first define the context of the problem or area for which you want information. To do so, use the Search Query field. Operations Analytics then uses the search query you specify to determine the related metrics, topology, inventory, event, and log file information to display. For more details, see "Search Tool" on page 26.
- **Play Back History.** Play back your search query results using the Playback feature.. For more details, see "Play Back History" on page 163.
- **Predictive Analytics.** OpsA's predictive analytics enables you to generate a prediction line for one or more metrics based on past behavior and seasonal trends. For more details, see "Predictive Analytics" on page 165.
- **Log Analytics.** Log Analytics is a forensic tool that helps you locate the most significant log messages in a given time range. For more details, see "Log Analytics" on page 167.
- **Topology Management.** The Topology Manager enables you define a logical hierarchy for monitored hosts. You can group hosts together based on their function, their location, or any other grouping that is meaningful to you when organizing your services. For more details, see "Topology Manager" on page 172.
- **User Management.** OpsA allows you to create and manage user accounts. For more details, see "Manage Users and Tenants" on page 221.

Tasks

How do I start using OpsA?

We recommend starting with one of the following tasks:

- Use the OAEnvironmentOverview dashboard to help determine, at a glance, problem areas to investigate more closely in your IT environment.

  Click Operations Analytics to navigate to the **OAEnvironmentOverview** dashboard. See "Dashboards and Query Panes" on page 13 for more information.

- Enter a search query that defines the context of the problem you are trying to solve.

  For example, you might query for CPU utilization information for a specific host name or for memory utilization for all database instances for a specified application.

  As you type, Operations Analytics provides a list of suggestions to help define the context of the problem you are trying to get information about. See "Search Tool" on page 26 for more information.

- Select an existing dashboard from the **Dashboards** menu.
- Create a new dashboard by selecting **New** from the **Dashboards** menu.

User Interface

How does the Operations Analytics console work?

1 Home Page

The Operations Analytics logo opens the OAEnvironmentOverview dashboard. This dashboard provides an overview of the following information for the hosts in your IT environment:

- Top five CPU utilization (cpu_util)
- Top five disk utilization (disk_io_rate)
- Top five memory utilization (mem_util)
- Top five network utilization (net_packet_rate)

Use this dashboard to help determine, at a glance, problem areas to investigate more closely in your network environment.

Note the following:

- Operations Analytics displays the LogsOverview dashboard when you initially log on to Operations Analytics.
- Each subsequent time you log on, Operations Analytics displays the last dashboard you accessed. In the **Dashboard** menu a check mark indicates the dashboard in use.
- Shared dashboards that have been provided by other members of your user community are appended with the name of the user who provided the dashboard.

To access the home page, click the Operations Analytics logo to return to the OAEnvironmentOverview dashboard.

See "Out-of-the-Box Dashboards Provided by Operations Analytics" on page 13 for more information.

2 Search Query

Defines the context for the data you want to examine.

Operations Analytics gathers and analyzes the data based on the search query you enter.

To perform a search, enter the string to search for. As you type, a list of suggestions are displayed to exhance the search query. This list is dynamically generated based on your data.

3 Time Range

Specifies the time frame within which Operations Analytics should obtain the data to display.

Use the Time Range menu to specify the time in hours, days, or months.

Note: The time range is historical. It spans the selected time range ending at the current time.

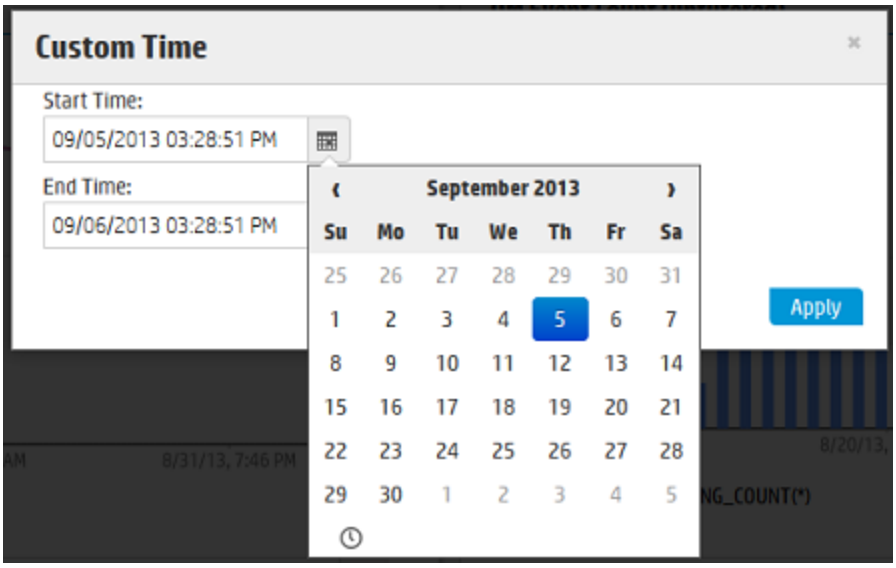
Use the **Custom Time** option when you want to specify a start and end date using the Operations Analytics calendar.

By default, Operations Analytics uses a time range of 1 Hour.

To use the time range feature:

1. In the Time Range menu, click .
2. Select the time in hours, days, or months.

3. To specify a start and end date using the Operations Analytics calendar, select **Custom Time** .
4. Click the calendar icon to display the calendar for either **Start Time** or **End Time** as shown in the following example.



5. After you have completed selecting your Start Time and End Time Dates, click **Apply**.

See "Filter Search Query Results " on page 161 for more information.

4 Time Line

Enables you to filter the time segment for which the data is displayed.

This feature is useful when you want to fine tune the Time Range selected.

To filter your analysis by time segment, slide each end of the time line to the beginning and end point of the time you want to use:

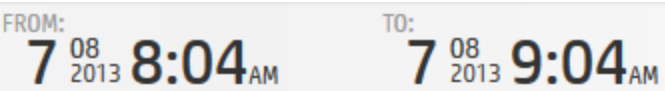
See "Filter Search Query Results " on page 161 for more information.

5 Time Segment

Displays the time segment you selected from the Time Line.

See "Filter Search Query Results " on page 161 for more information.

After you slide each end of the time line to the beginning and end point of the time you want to use, the From and To time changes to match the latest selection.



6 New Query Pane

Enables you to add one or more query panes using one of the following:

- Analytics Query Language (AQL) query
- AQL function

- AQL expression

To add a new query, see "Dashboards and Query Panes" on page 13.

7 Log In Information

Displays your user name. See "About User Accounts" on page 221 for more information.

Enables you to do the following:

- Access user settings
  - Change your password. See "Change Your User Account Password" on page 225 for more information.
  - Log out.

To log out, click your user account name and select **Logout**.

8 HP ArcSight Logger

To launch HP ArcSight Logger, click and select the HP ArcSight Logger IP address or host name to which you want to connect.

9 Settings

Access the following settings ( ).

- *Operations Analytics administrators only*. **User Management**. For details, see "Manage Users and Tenants" on page 221
- **Topology Management**. For details, see "Topology Manager" on page 172

10 Help, Reference Pages and About

Access the following information for Operations Analytics:

- Help
- Reference pages - descriptions of command line interface commands.
- License, database, and version information

11 Dashboards

Lists the following saved dashboards:

- Provided by Operations Analytics.
- Shared by the users in your user community (tenants).
- Saved by the current user.

You can select a dashboard from this list rather than using the search query to create your own.

For more details, see "Dashboards and Query Panes" on page 13.

12 Playback

Replay Operations Analytics results.

This option is useful to help you identify when a problem began to occur.

For more details, see "Play Back History" on page 163

13 Dashboard Area

An Operations Analytics dashboard is the graphical user interface for troubleshooting your IT operations problems.

For more details, see "Dashboards and Query Panes" on page 13.

Note: When you first access Operations Analytics, it displays the LogsOverview dashboard. Each subsequent time you log on, Operations Analytics displays the last saved dashboard you accessed. In the **Dashboards** menu a check mark indicates the dashboard in use.

Chapter 2: Dashboards and Query Panes

A dashboard is the graphical user interface for troubleshooting your IT operations problems.

Dashboards are collections of Query Panes defined in a specific layout. Dashboards allow you to customize your layout and can be shared with other users.

To access

- Enter a new search query in the Search Query field.
- Select an existing dashboard from the **Dashboards** menu.
- Create a new dashboard by selecting **New** from the **Dashboards** menu.

Learn About

Overview

A dashboard is the graphical user interface for troubleshooting your IT operations problems.

Dashboards are collections of Query Panes defined in a specific layout. Dashboards allow you to customize your user interface and save the settings.

The first time you access Operations Analytics, it displays the **LogsOverview** dashboard. This dashboard lists all of the log messages from the log files that have been configured to be collected in your IT environment. Use this dashboard as a starting point to look for errors that might have occurred.

Out-of-the-Box Dashboards Provided by Operations Analytics

Note the following about dashboards provided by Operations Analytics:

- Only a user assigned to the Tenant Admin user group can change a dashboard provided by Operations Analytics. See "About User Groups" on page 221 and opsa-tenant-manager.sh for more information about the Tenant Admin user group.
- To save changes to a dashboard provided by Operations Analytics, open the **Dashboard** menu and select **Save As**.
- Each time you enter a Phrased Query Language (PQL) query that is not a dashboard name, Operations Analytics displays the default visualizations for the metrics, topology, inventory, event, and log file information available in response to your search query. See "Working with Query Panes" on page 154 for more information about default visualizations.
- If a search query results in too many instances to display, Operations Analytics uses the query to return log file messages matching any phrases in the search query. See "Refine Your Search Query " on page 70 for more information.
- If no tags, metrics or dashboards match the query, Operations Analytics returns log file

messages matching any phrases in the search query.

- If a search query does not return results, Operations Analytics displays the message that no matches could be found.

Dashboards Provided by Operations Analytics

Name	Description
Logs Apache	**Note:** This dashboard is available only if you have installed the *Apache HTTP Server Access File* and *Apache HTTP Server Error File* SmartConnectors provided by HP ArcSight Logger. SmartConnectors are not included as part of Operations Analytics. Displays the following information. Information for access log and error log are displayed next to each other: • Log messages count over time • Log messages count by severity • Top 10 hosts with failure messages • Total errors per host • Log messages
BPM Applications Overview	**Note:** See "Configuring an HP Business Process Monitor Collection" in the HP Operations Analytics Configuration Guide for the configuration steps required to display this dashboard information. Use the BPM Applications Overview to view the following: • Application Availability Over Time The heat map value in this dashboard is the number of failed transactions. • Application Performance Over Time • Application Layer Performance Over Time • Top 10 Transactions Performance • Top 10 Locations Performance

Dashboards Provided by Operations Analytics, continued

Name	Description
Logs Linux	**Note:** This dashboard is available only if you have installed the *Linux Audit File* and *Linux Syslog File* SmartConnectors provided by HP ArcSight Logger. SmartConnectors are not included as part of Operations Analytics. Displays the following information. Information is calculated per host. • Log messages count over time • Log messages count by severity • Top 10 hosts with failure messages • Top 10 log message categories • Log messages
Logs Search	**Note:** See "Installing and Configuring HP ArcSight Logger" in the HP Operations Analytics Installation Guide for the configuration steps required to display this dashboard information. Displayed by default when you initially log on to Operations Analytics. This dashboard provides an overview of the following information for the log messages in your IT environment: • Log Messages - All • Log Messages - Syslog Only

Dashboards Provided by Operations Analytics, continued

Name	Description
NNMi Network SPI	Displays the following information: • Top 10 Network Interfaces with Utilization In • Top 10 Network Interfaces with Utilization Out • Top 10 network interfaces based on highest error percentages • Top 10 network interfaces based on highest discard percentages • Top 10 network interfaces based on highest in and out throughput • Top 10 network devices based on highest CPU utilization • Top 10 network devices based on highest memory utilization • Top 10 unavailable nodes • Top 10 network devices based on highest SNMP response times
OA Environment Overview	**Note:** See "Configuring an HP Operations Agent Collection" in the HP Operations Analytics Configuration Guide for the configuration steps required to display this dashboard information. This dashboard provides an overview of the following information for the hosts in your IT environment: • Top 10 CPU utilization (cpu_util) • Top 10 disk utilization (disk_io_rate) • Top 10 memory utilization (mem_util) • Top 10 network utilization (net_packet_rate) Use this dashboard to help determine, at a glance, problem areas to investigate more closely in your network environment. To return to this dashboard, click Operations Analytics.

Dashboards Provided by Operations Analytics, continued

Name	Description
OM Events	**Note:** See "Configuring an HP Operations Manager (HPOM) Events Collection" in the HP Operations Analytics Configuration Guide for the configuration steps required to display this dashboard information. Use this dashboard to view the following information: • Event Count Over Time • Top 10 Hosts with Event Count Over Time • Event Count by Host - Current Week • Event Count by Host - Previous Week • Event Count by Severity - Current Week • Event Count by Severity - Previous Week • Table of the first 500 OM events
OMi Events	**Note:** See "Configuring an HP Operations Manager i (OMi) Events Collection" in the HP Operations Analytics Configuration Guide for the configuration steps required to display this dashboard information. Use this dashboard to view the following information: • Total count of the OMi events over time • Percentage of OMi events by host • Total count of OMi events by State • Top hosts that have highest number of OMi events • Percentage of OMi events by application • Event count by the host • Event count by host from the previous week • Event count by severity • Event count by severity from the previous week • Table of the first 500 OMi events

Dashboards Provided by Operations Analytics, continued

Name	Description
Opsa Health	**Note:** See "Checking Operations Analytics System Health" in the HP Operations Analytics Configuration Guide for the configuration steps required to display this dashboard information. Displays the metrics, topology, and log information available for the following Operations Analytics servers and appliances: • Operations Analytics Collector Appliance • Operations Analytics Server Appliance • List of configured collections that OpsA is collecting data for. This dashboard provides current details about Operations Analytics system health. See "Check the Health of Operations Analytics" on page 231 for more information.
SiteScope Environment Overview	Displays the following information monitored by SiteScope: • Top CPU Utilization • Top Disk Utilization • Top Memory Utilization • Top 10 Hosts with Ping Roundtrip Time • Top 10 Hosts with URL Content Roundtrip Time • Top 10 Hosts with JMX Physical Memory
OpsA Meta Info	Displays the following information for the collections in your IT environment: • Collections and any tags for each collection • Columns (metrics) per collection and tag names per column • Columns defined as keys. See "How to View Collection Information" on page 217 for more information.

Dashboards Provided by Operations Analytics, continued

Name	Description
Logs Windows	**Note:** This dashboard is available only if you have installed the *Microsoft Windows Event Log - Local* SmartConnector provided by HP ArcSight Logger. SmartConnectors are not included as part of Operations Analytics. Displays the following information. Information is calculated per host. • Log messages count over time • Log messages count by severity • Top 10 log message categories • Top 10 hosts with failure messages • Log messages

Tasks

How to Save a Dashboard

Dashboards are automatically saved when you add/remove Query Panes or modify the dashboard layout.

To copy a dashboard and save it under a new name, see the procedure for copying a dashboard below.

Tip: If you want to experiment with different dashboard layouts, save a copy of the original layout under a different name. Otherwise, OpsA will overwrite the original dashboards as it automatically saves any changes you make.

How to Copy a Dashboard

1. Navigate to the **Dashboard** menu.
2. Click **Manage**.
3. Click the check box ☑ for the dashboard you want to copy.
4. Click **Copy**.
5. In the **Specify a new name** dialog, enter the name of the copied dashboard.
6. Click **OK**.

   The copied dashboard appears in the **Dashboards** menu.

How to Delete a Dashboard

1. Navigate to the **Dashboard** menu.
2. Click **Manage**.
3. Click the check box ☑ for each dashboard you want to delete.
4. Click **Delete**.
5. Click **OK**.

   The dashboard name is removed from the **Dashboards** menu.

How to Share a Dashboard

1. Navigate to the **Dashboards** menu.
2. Click **Manage**.
3. Click the check box ☑ for each dashboard you want to share.
4. Click **Share**.

   Each dashboard you select is available to all users in the same tenant.

   Note: Shared dashboards that have been provided by other members of your user community are appended with the name of the user who provided the dashboard.

How to Stop Sharing a Dashboard

1. Navigate to the **Dashboards** menu.
2. Click **Manage**.
3. Click the check box ☑ for each dashboard you want to unshare.
4. Click **Unshare**.

   Note: Each dashboard you select is removed from the dashboard menu of other users in your user community (tenant).

How to Add or Edit a Query Pane

1. Click +.
2. Navigate to the **Query** tab.

3. Do one of the following:

   - In the **(NEW PANE)** attribute, enter the AQL query, AQL function name, or AQL expression for the new query pane.

     OR

   - Select an AQL function.

     Enter values for any of the AQL function arguments that apply.

     Note the following:

     - Your Operations Analytics administrator can provide descriptions for the arguments required for each AQL Function provided. See "Add / Edit Query Pane - Query Tab" on page 24 for information about how to view these descriptions.

     - If descriptions are not provided, you can also view the collection information configured for your IT environment. This collection information might also assist you in providing values for the arguments required.

       Click **Show Tags** to view a new query pane that displays the collections (property group uid), tags (tag name), and columns (property uid) available. Click **Show Properties** to view a new query pane that displays the collections (property group uid), columns (property uid), and whether the column contains **metric**[1] or **attribute**[2] values.

       Also see "How to View Collection Information" on page 217 for more information about how to view the meta data stored for your collections.

     Click here for a brief description of the possible AQL function argument types.

     AQL Function Argument Types

Argument Type	Description
analytic	Specifies an analytic function that can be applied to overall aggregate analytic functions, moving aggregate analytic functions, or raw metrics. These analytic functions include: topN, bottomN, inverse_pctile, pctile, outlier, or rank. See "About Analytic Functions" on page 98 for more information.
collection	Specifies the name of the collection for which Operations Analytics should return search results.

[1]Typically a measurement stored in a collection. For example, CPU utilization.
[2]A descriptor stored in a collection for an entity, such as host_name.

AQL Function Argument Types, continued

Argument Type	Description
custom	Indicates that Operations Analytics cannot identify the argument type. Check the description for the AQL function that appears in the Query tab when adding or editing a query pane. Also, check with your Operations Analytics administrator for assistance with providing values for these arguments.
entity	Specifies the type of entity attribute on which you want to filter; for example, host_name.
filter	Specifies the filter value to use in the `where` clause of the AQL function. For example, when used with host name, you might enter the following filter value to return data for only the servers in the co.usa.enterprise. com domain: `\"*\.co.usa.enterprise.com"`.
grouping	Specifies an argument required for the group by clause.
function	Specify the overall aggregate or moving aggregate analytic function you want Operations Analytics to use. See "Analytic Functions Provided by Operations Analytics" on page 100 for the list of analytic functions provided by Operations Analytics.
metric	Either of the following: ○ Name of the metric column. ○ Tag that represents the metric column.
ordering	Specifies an argument required for the `order by` clause.

4. *Optional*. Use the **Visualization** tab to change the visualization that is displayed.

   a. Navigate to the **Visualizations** tab.

   b. Navigate to the Visualizations options:

   c. Select the visualization you want to use.

   d. Navigate to another tab or click **OK**.

e. **Note:** If you select a visualization that is not valid for the data displayed, Operations Analytics displays the default visualization for the AQL query.

See "Working with Query Panes" on page 154 for more information about visualizations.

5. Use the **Parameters** tab to provide the parameter values, if any, to the selected AQL function.

Note: Any parameter value you provide overrides the associated value selected using another method in the Operations Analytics console. For example, if you specify a time interval using the $interval parameter, Operations Analytics uses the value for $interval rather than the time line segment selected. See "Filter Search Query Results " on page 161 for more information about time line segments.

a. Navigate to the **Parameters** tab.

b. Provide the parameter values you want to use.

Tip: Mouse over a parameter to view its description.

To restore the parameter values to their original default values, click **Defaults**.

c. Navigate to another tab or click **Save** to save your changes.

How to Resize a Query Pane.

Navigate to the query pane you want to change. Mouse over the upper right-hand corner of the query pane. Use the query pane toolbar to change the height and width of the query pane:

How to Delete a Query Pane from the Dashboard

Click **x** in the upper right-hand corner of the pane to close the query pane and remove it from your dashboard.

UI Description

Dashboard Menu

Item	Description
Dashboard Name List	Operations Analytics lists all of the dashboards available for your use. These include: • Dashboards created by the current user. • Dashboards shared by other users in the same user community (tenant). **Note:** The shared dashboards, including those provided by Operations Analytics appear at the top of the list.
New	Creates a new dashboard.
Save As	If you are in an unsaved dashboard as a result of a search, Save As saves the search results as a dashboard. If you are in a saved dashboard, Save As creates a copy with a new name.
Manage	Enables you to copy, share, unshare, or delete a dashboard that you no longer need from the **Dashboards** menu. **Note:** You can delete only dashboards that you created.

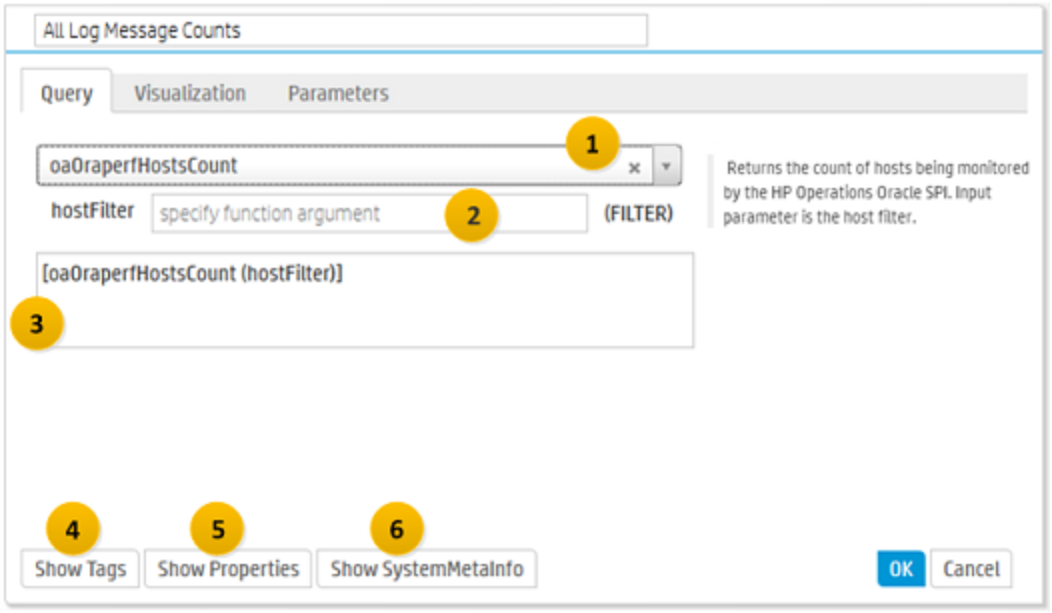
Add / Edit Query Pane - Query Tab

When adding a new query pane, you can use the **Query** tab to specify the pre-defined AQL function you want to use as your search query.

The following illustration highlights the main features of the Query tab.

Note: You can also choose to enter your own AQL query. If you want to use an AQL function, either select one from the list or create the function using a text editor. See "Define Analytic Query Language (AQL) Functions" on page 233 for more information.

Click each number for information about a feature.


All Log Message Counts
Query
Visualization
Parameters
1
oaOraperfHostsCount
Returns the count of hosts being monitored by the HP Operations Oracle SPI. Input parameter is the host filter.
hostFilter
specify function argument
2
(FILTER)
[oaOraperfHostsCount (hostFilter)]
3
4
Show Tags
5
Show Properties
6
Show SystemMetaInfo
OK
Cancel

Chapter 3: Search Tool

To use Operations Analytics, you must first define the context of the problem or area for which you want information. To do so, use the Search Query field. Operations Analytics then uses the search query you specify to determine the related metrics, topology, inventory, event, and log file information to display.

Tip: You can also view information in an existing dashboard without entering a search query. To do so, select a dashboard from the **Dashboards** menu. The **Dashboards** menu lists the dashboards that are provided by Operations Analytics and that have been saved by you or shared by other users in your user community (Tenant). See "Out-of-the-Box Dashboards Provided by Operations Analytics" on page 13 for more information.

You can take any of the following approaches to your search query:

- **Phrased Query Language (PQL)**:

  PQL is a more natural search language that uses phrases to define your search. You must select or enter PQL in the Search Query field. Each PQL query results in a dashboard that is comprised of one or more **query panes**[1].

  Use this search as a starting points for troubleshooting a problem or investigating your environment.

  Use the list of suggestions provided by Operations Analytics for guidance.

  Operations Analytics provides suggested searches based on the Phrased Query Language described in "Elements in a Phrased Query (for Metrics)" on page 57.

  See "About Guided Search (PQL)" on page 46 and "About the Phrased Query Language (PQL)" on page 50 for more information.

- **Analytics Query Language** (**AQL**):

  AQL is a more structured query language. It requires a specific syntax and results in a single query pane. Each query pane contains a visualization (for example, line chart, pie chart, heat map, bar chart, or sunburst chart) of the data returned.

  Use the Operations Analytics AQL, when the Phrased Query Language syntax is not specific enough to return the data you need. It is also useful when you want to perform more detailed queries. When using the more flexible, but more complex AQL, it is helpful if you have some knowledge of databases and are familiar with the data that is being collected by Operations Analytics. See "About the Analytics Query Language (AQL) " on page 71 for more information.

[1]Displays the results of an Analytic Query Language (AQL) query, AQL function, or AQL expression. If you use the Phrased Query Language (PQL) in your search, HP Operations Analytics converts the PQL query to one or more AQL queries and subsequent query panes.

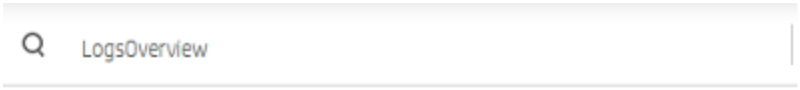
Note: Any PQL search that you use to create a dashboard is converted to one or more Analytics Query Language (AQL) queries.

Tip: The best way to start using AQL is to edit the AQL query that has been used to create an existing query pane. You can view the associated AQL query for any dashboard query pane by editing the query pane. When editing a query pane you can also select an Analytic Function that contains one or more AQL queries. See "Dashboards and Query Panes" on page 13 for more information.

Note: Operations Analytics does not accept AQL queries in the Search Query field. To create a dashboard using AQL queries, use + to add one or more **query panes**[1]. See "Dashboards and Query Panes" on page 13 and "About the Analytics Query Language (AQL) " on page 71 for more information.

To enter a search query:

1. Navigate to the Search Query field:

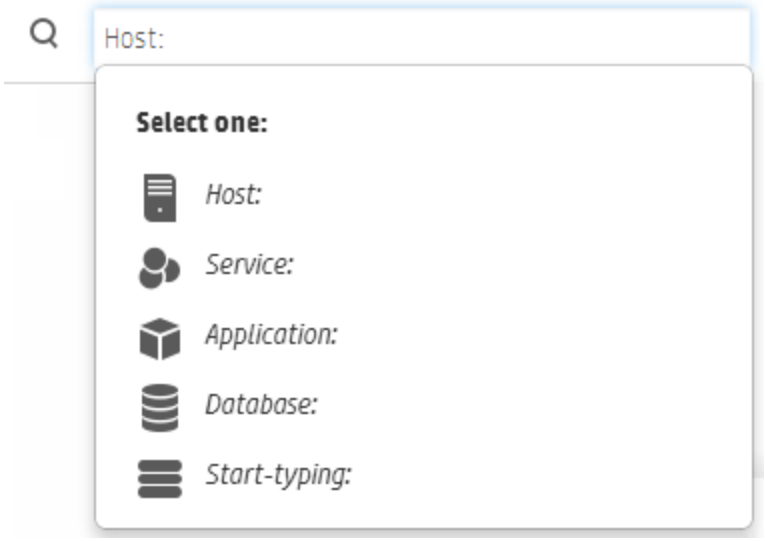


2. Do one of the following:

[1]Displays the results of an Analytic Query Language (AQL) query, AQL function, or AQL expression. If you use the Phrased Query Language (PQL) in your search, HP Operations Analytics converts the PQL query to one or more AQL queries and subsequent query panes.

- Start with an entity.

  i. Place your cursor in the Search Query field.



  Operations Analytics displays a list of possible queries for the entity selected.

  Tip: Use the arrows keys to navigate the list of suggestions.

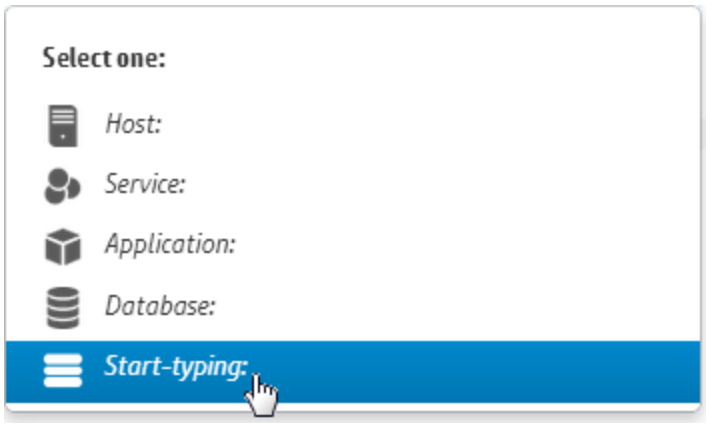
  ii. Select an entity from the list of options:

  iii. Each time you select a suggested query, press the space bar to view the next set of possible queries..

  iv. Continue this process until your query is complete.

- Start typing.

  i. Place your cursor in the search field and select **Start typing** from the list of options:



  Note: Using the meta data available, Operations Analytics provides up to 15 suggestions per query. Operations Analytics also lists up to five suggestions from your Search History.

  See "About Phrased Query Suggestions" on page 53 for more information.

  ii. Do one of the following:

    i. Select from the list of suggestions.

    Press the space bar to continue to view suggestions.

    ii. Type a word or phrase.

    As you type, Operations Analytics provides a list of suggested phrases to continue to define possible queries.

  iii. Continue selecting a suggestion or typing a word or phrase until you find or enter the query that best matches the information you want to view.

3. *Optional*. Select a time range from the menu. The default time range is 1 Hour. See "Filter Search Query Results " on page 161 for more information.

   Tip: Selecting a time range also executes your search query. See "Dashboards and Query Panes" on page 13 for more information.

4. Press **[Enter]** to execute your query.

Operations Analytics adds the completed search query to your search history for later suggestions.

Note: If a search query results in too many instances to display, Operations Analytics uses the query to return log file messages matching any phrases in the search query. See "Refine Your Search Query " on page 70 for more information.

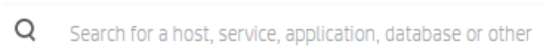
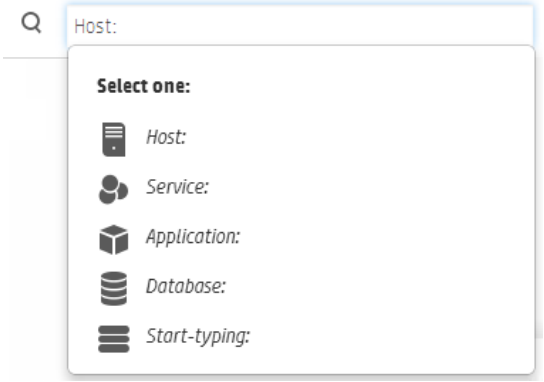
Operations Analytics uses its default dashboard layout and populates the dashboard with the data requested by your search.

Determine the Search Approach to Use

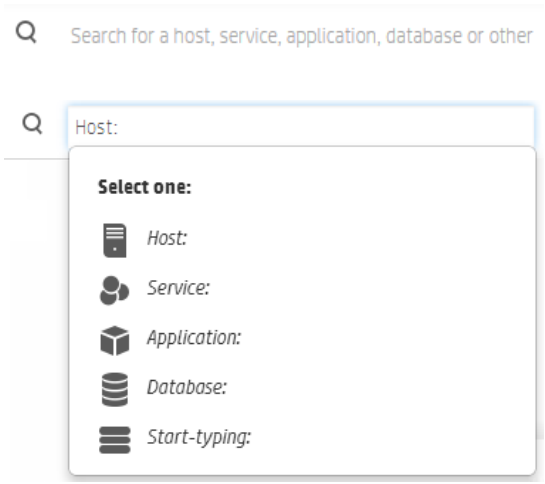
To use Operations Analytics, you define the context of the problem or area for which you want information. To do so, use the Search Query field to select or enter a search query. Operations Analytics then uses the search query you specify to determine the related metrics, topology, inventory, event, and log file information to display.

Operations Analytics offers multiple ways for you to search your data. Use the table below to determine which search is best for your needs.

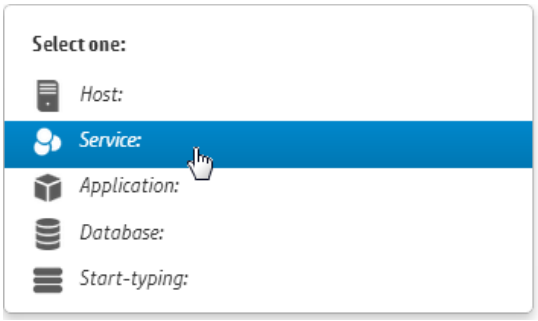
Decide Which Search to Use

When you want to...	Use this...	How to Get Started...
You are using Operations Analytics for the first time	PQL	1. Place your cursor in the Search Query field:  You should view the following options:  2. Select the option that best describes the entity might be related to the area you want to investigate. **Tip:** Select **Advanced** only if you do not know the entity for which you want information. For more details, see "About Guided Search (PQL)" on page 46.

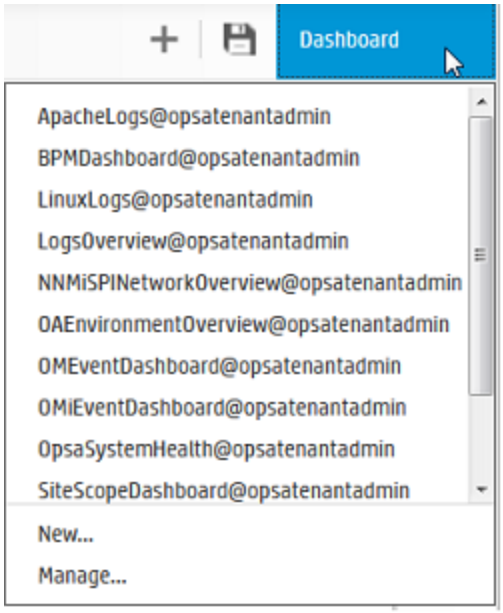
Decide Which Search to Use, continued

When you want to...	Use this...	How to Get Started...
You know the problem is related to a host, service, application, or database	PQL	1. Place your cursor in the Search Query field:  2. Select an entity (**Host**, **Service**, **Application**, **Database**) that might be related to the area you want to investigate. For more details, see "About Guided Search (PQL)" on page 46.

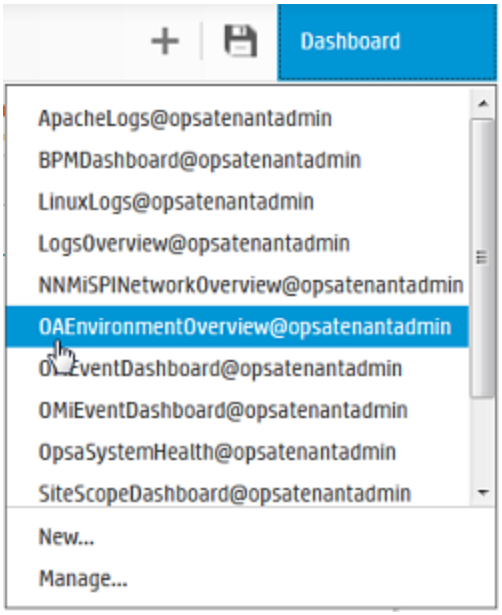
Decide Which Search to Use, continued

When you want to...	Use this...	How to Get Started...
You want to view information about a service; for example the topology relationships among servers for a service	PQL	1. In the Search Query field menu, select **Service**:  2. Select the suggestion that includes the service name. 3. Continue to select the suggestions that best describes the service information you want to view. For more information, see "Topology Manager" on page 172, "About the Phrased Query Language (PQL)" on page 50 and "Elements in a Phrased Query (for Metrics)" on page 57

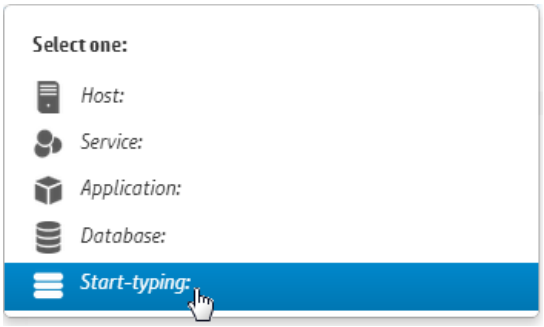
Decide Which Search to Use, continued

When you want to...	Use this...	How to Get Started...
You know the name of an existing dashboard	**Dashboards** menu	1. Navigation to the **Dashboards** menu:  2. Select the Dashboard name. For more details, see "Dashboards and Query Panes" on page 13.

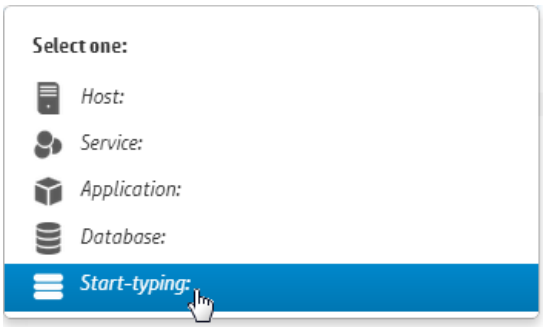
Decide Which Search to Use, continued

When you want to...	Use this...	How to Get Started...
You want to view the overall status of your IT environment	**Dashboards** menu	1. From the **Dashboads** menu, select **OAEnvironmentOverview:**  2. Use this dashboard to help determine, at a glance, problem areas to investigate more closely in your network environment. This Operations Analytics dashboard provides an overview of the following information for the hosts in your IT environment: ▪ Top 5 CPU utilization (cpu_util) ▪ Top 5 disk utilization (disk_io_rate) ▪ Top 5 memory utilization (mem_util) ▪ Top 5 network utilization (net_packet_rate) For more details, see "Out-of-the-Box Dashboards Provided by Operations Analytics" on page 13.

Decide Which Search to Use, continued

When you want to...	Use this...	How to Get Started...
You want to view status or health information other than what is displayed in the OAEnvironmentOverview dashboard.	PQL	1. From the **Dashboards** menu, select **SystemMetaInfo**. 2. Navigate to the **Collection Tags** and **Tags per Collection Column** tables. 3. Look for words in the **tag names** column to determine the types of data that are configured to be collected. 4. From the Search Query field, select **Start-typing**:  5. Enter a tag name that appears in the SystemMetaInfo dashboard and that describes overview information you want to view; for example: **status** or **health** 6. Press the space bar to continue to view suggestions. 7. Select the suggestion that most closely matches the information you want to view. For more details, see "About the Phrased Query Language (PQL)" on page 50.

Decide Which Search to Use, continued

When you want to...	Use this...	How to Get Started...
You know the problem area, but are not certain of the host or application that is causing the problem	PQL	1. From the **Dashboards** menu, select **SystemMetaInfo**. 2. Navigate to the **Collection Tags** and **Tags per Collection Column** tables. 3. Look for words in the **tag names** column to determine the types of data that are configured to be collected. 4. From the Search Query field, select **Start-typing**:  5. Enter a tag name that appears in the SystemMetaInfo dashboard and that describes the problem area; for example: **cpu utilization**, **change**, **error**, **disk utilization**, **memory performance** 6. Select from the list of suggestions. **Tip:** You can also type a word or phrase at any time after you select **Advanced**. 7. Press the space bar to continue to view suggestions. 8. Select the suggestion that most closely matches the information you want to view. For more details, see "Out-of-the-Box Dashboards Provided by Operations Analytics" on page 13.

Decide Which Search to Use, continued

When you want to...	Use this...	How to Get Started...
You are viewing an existing dashboard, but want more specific information than what is displayed.	AQL	1. Navigate to the query pane you want to change. 2. Place the mouse over the top section of the query pane and click  . 3. To review the functions provided by Operations Analytics, see "Analytic Functions Provided by Operations Analytics" on page 100. **Tip:** Use the moving aggregate analytic functions provided by Operations Analytics. 4. In the **Query** tab, do one of the following: ▪ Edit the AQL query displayed. ▪ Select an AQL function from the list provided. **Tip:** If you are not sure which AQL function to select, look at the description displayed to the right of the function syntax. ▪ Enter a new AQL query. 5. To learn the AQL syntax and requirements, see "About the Analytics Query Language (AQL) Syntax and Structure" on page 72. 6. To view AQL query examples, see "Examples of Using Analytic Functions in AQL Queries" on page 121 For more details, see "Dashboards and Query Panes" on page 13.

Decide Which Search to Use, continued

When you want to...	Use this...	How to Get Started...
You are viewing an existing dashboard and want to change the visualization	Use the **Visualization** tab in the query pane.	1. Navigate to the query pane you want to change. 2. Place the mouse over the top section of the query pane and click  . 3. In the **Visualization** tab, select the visualization you want to view. **Note:** If the visualization you select is not valid for the data displayed, Operations Analytics displays the data in a table view. For more details, see "Dashboards and Query Panes" on page 13.

Decide Which Search to Use, continued

When you want to...	Use this...	How to Get Started...
You want to view trend information over time for any of the following: • Averages • Total values • Total counts • Minimum or maximum values	AQL	1. Navigate to the query pane you want to change. 2. Place the mouse over the top section of the query pane and click  . **Tip:** To add a new query pane instead, click  . 3. In the **Query** tab, do one of the following: ▪ Edit the AQL query displayed. ▪ Select an AQL function from the list provided. **Tip:** If you are not sure which AQL function to select, look at the description displayed to the right of the function syntax. ▪ Enter a new AQL query. 4. To review the functions provided by Operations Analytics, see "Analytic Functions Provided by Operations Analytics" on page 100. **Tip:** Use the moving aggregate analytic functions provided by Operations Analytics. 5. To learn the AQL syntax and requirements, see "About the Analytics Query Language (AQL) Syntax and Structure" on page 72. 6. To view AQL query examples, see "Examples of Using Analytic Functions in AQL Queries" on page 121 For more details, see "Analytic Functions Provided by Operations Analytics" on page 100, "About the Analytics Query Language (AQL) Syntax and Structure" on page 72, and "Examples of Using

Decide Which Search to Use, continued

When you want to...	Use this...	How to Get Started...
		Analytic Functions in AQL Queries" on page 121.

Decide Which Search to Use, continued

When you want to...	Use this...	How to Get Started...
You want to view totals (summary) information for any of the following: • Averages • Total values • Total counts • Minimum or maximum values	AQL	1. Navigate to the query pane you want to change. 2. Place the mouse over the top section of the query pane and click  . **Tip:** To add a new query pane instead, click  . 3. In the **Query** tab, do any of the following: ▪ Edit the AQL query displayed. ▪ Select an AQL function from the list provided. **Tip:** If you are not sure which AQL function to select, look at the description displayed to the right of the function syntax. ▪ Enter a new AQL query. 4. To review the functions provided by Operations Analytics, see "Analytic Functions Provided by Operations Analytics" on page 100. **Tip:** Use the overall aggregate analytic functions provided by Operations Analytics. 5. To learn the AQL syntax and requirements, see "About the Analytics Query Language (AQL) Syntax and Structure" on page 72. 6. To view AQL query examples, see "Examples of Using Analytic Functions in AQL Queries" on page 121 For more details, see "Analytic Functions Provided by Operations Analytics" on page 100, "About the Analytics Query Language (AQL) Syntax and Structure" on page 72, and "Examples of Using

Decide Which Search to Use, continued

When you want to...	Use this...	How to Get Started...
		Analytic Functions in AQL Queries" on page 121.

Decide Which Search to Use, continued

When you want to...	Use this...	How to Get Started...
You want to apply an additional analytic to the data collected. For example you want to view any of the following results: • topN values • bottomN values • outlier data • percentile ranking	AQL	1. Navigate to the query pane you want to change. 2. Place the mouse over the top section of the query pane and click  . **Tip:** To add a new query pane instead, click  . 3. In the **Query** tab, do one of the following: ▪ Edit the AQL query displayed. ▪ Select an AQL function from the list provided. **Tip:** If you are not sure which AQL function to select, look at the description displayed to the right of the function syntax. ▪ Enter a new AQL query. 4. To review the functions provided by Operations Analytics, see "Analytic Functions Provided by Operations Analytics" on page 100. **Tip:** Use the additional analytic functions that can be applied to moving aggregate and overall aggregate analytic functions provided by Operations Analytics. 5. To learn the AQL syntax and requirements, see "About the Analytics Query Language (AQL) Syntax and Structure" on page 72. 6. To view AQL query examples, see "Examples of Using Analytic Functions in AQL Queries" on page 121 For more details, see "Analytic Functions Provided by Operations Analytics" on page 100, "About the Analytics Query Language (AQL) Syntax and

Decide Which Search to Use, continued

When you want to...	Use this...	How to Get Started...
		Structure" on page 72, and "Examples of Using Analytic Functions in AQL Queries" on page 121.
You want to create your own AQL queries for re-use	AQL	1. Write your own AQL functions using a text editor and then import these functions into Operations Analytics. Each text file you create can contain any number of AQL functions. A set of AQL functions that reside in a single file are known as an AQL module 2. To learn the AQL syntax and requirements, see "About the Analytics Query Language (AQL) Syntax and Structure" on page 72. 3. To view AQL query examples, see "Examples of Using Analytic Functions in AQL Queries" on page 121 For more details, see "Define Analytic Query Language (AQL) Functions" on page 233.

Decide Which Search to Use, continued

When you want to...	Use this...	How to Get Started...
You want to add another pane to your dashboard	AQL	1. Click +. 2. In the **Query** tab, do one of the following: ▪ Edit the AQL query displayed. ▪ Select an AQL function from the list provided. **Tip:** If you are not sure which AQL function to select, look at the description displayed to the right of the function syntax. ▪ Enter a new AQL query. 3. To review the functions provided by Operations Analytics, see "Analytic Functions Provided by Operations Analytics" on page 100. 4. To learn the AQL syntax and requirements, see "About the Analytics Query Language (AQL) Syntax and Structure" on page 72. 5. To view AQL query examples, see "Examples of Using Analytic Functions in AQL Queries" on page 121 For more details, see "Analytic Functions Provided by Operations Analytics" on page 100, "About the Analytics Query Language (AQL) Syntax and Structure" on page 72, and "Examples of Using Analytic Functions in AQL Queries" on page 121.

About Guided Search (PQL)

Use the Guided Search approach to the Operations Analytics Phrased Query Language (PQL) in the early stages of troubleshooting a problem or investigating your IT environment.. With this approach, place the cursor in the Search Query field and let Operations Analytics guide your search.

When using Guided Search you need to understand the following concepts:

Collection

Operations Analytics stores metrics, topology, inventory, log file, and event information in the form

of collections. Each collection is associated with a database table in which an Operations Analytics Collector stores the data collected.

Data is stored in a collection as either a metric or an attribute.

Each collection is associated with one or more tags.

Metric

A metric is a measurement stored in a collection; for example, disk_write_byte_rate.

Key Column

A column in a collection table that is considered to uniquely identify an entity can be configured as a key column. Operations Analytics recognizes values that reside in a key column in a PQL search. When using Guided Search, any value listed as a suggestion is either a metric value that resides in a key column, a column name or a tag.

Attribute

An attribute is a descriptor stored in a collection for an entity, such as host_name.

Tag

A tag is a word that is associated with a collection or with a metric or attribute that is stored as part of a collection.

Tags are used in the Operations Analytics Phrased Query Language (PQL) to create an Operations Analytics dashboard. They help to define the following:

Note: Tags are not limited to these example uses.

- Entities for which you want information, such as **host**, **database**, and **application**
- Hardware and software components, such as **cpu**, **memory**, **disk**, **interface**, **tablespace**, **process**, and **threads**
- Metrics or problem areas, such as **utilization**, **availability**, **performance**, and **change**

See "About Tags" on page 175 for more information.

Service

Your Operations Analytics administrator might have configured one or more services using the Topology Manager.

When a service is configured, the Operations Analytics administrator specifies the groups that are included in the service. These groups, can include any of the following entities:

- application servers
- database servers

- web servers
- other groups; for example, based on location

You can select a Guided Search entry that queries information on an entire service, on a single group of a service, or on a particular instance (for example, database instance) that is associated with a service.

See "Searching for a Service Defined in Topology Manager" on page 69 for more information.

Using Guided Search can be a one- to two-step process.

Step One: Select the entity for which you want information. See Entity Descriptions.

1. Place your cursor in the Search Query field.
2. Select an entity from the list of suggestions.

Entity Descriptions

Entity	Description
Host	Select **Host** when you want to return information about a specific host in your IT environment. In response, Operations Analytics provides a list of the hosts for which it has collected information.
Service	**Note:** To use this option, your Operations Analytics administrator must have configured one or more services using the Topology Manager. See "Topology Manager" on page 172 for more information. Select **Service** when you want to view information about one of your services. You can select a Guided Search entry that queries information on an entire service, on a single group of a service, or on a particular instance (for example, database instance) that is associated with a service.
Application	Select **Application** when you want to view information about an application. For example, a user group might have reported that an application has slow response times.
Database	Select **Database** when you suspect a problem is associated with a database, database server, or when you want to view information about one or more database servers in your IT environment.

3. Press the space bar to view suggestions to continue your search.

Operations Analytics suggests the names of the first 15 hosts for which information is available.

Note: You can choose to press Enter at any time to view the results. If a search query results in too many instances to display, Operations Analytics uses the query to return log file messages matching any phrases in the search query. See "Refine Your Search Query " on page 70 for more information.

Step Two: Narrow Your Search (Select Drill to: or Focus on:)

Narrowing your search can be an iterative process. Each subsequent list of suggestions further limits the information returned.

Tip: If you do not want to narrow your search, select **All** to return all of the data available for the entity selected.

Operations Analytics provides two ways to narrow your search: **Drill to:** and **Focus on:**. See Narrow Your Guided Search.

Narrow Your Guided Search

Suggestion	Description
Drill to:	Applies to **Service** only. **Note:** Your Operations Analytics administrator must have configured one or more services using the Topology Manager. When a service is configured, the Operations Analytics administrator specifies each group that is included in the service. See "Topology Manager" on page 172 for more information. Use **Drill to**: when you want to filter your query using topology information. For example, you can narrow your search using a host name that is configured for the service you selected.
Focus on:	Applies to **Host**, **Application**, and **Database** only. Use **Focus on:** when you want to narrow your search using a tag . Use the **SystemMetaInfo** dashboard to view the tags available in your IT environment. See "How to View Collection Information" on page 217 for more information.

Operations Analytics adds the completed search query to your search history for later suggestions.

Note: If a search query results in too many instances to display, Operations Analytics uses the query to return log file messages matching any phrases in the search query. See "Refine Your Search Query " on page 70 for more information.

The results of each phrased query is an Operations Analytics dashboard. Operations Analytics uses its default dashboard layout and populates the dashboard with the data requested by your search. See "Dashboards and Query Panes" on page 13 for more information.

About the Phrased Query Language (PQL)

Use Phrased Query Language (PQL) in the early stages of troubleshooting a problem or investigating your IT environment.

Note: To access the PQL described in this topic, select **Start typing:** from the Search Query field options and type a word or phrase that begins to describe the type of problem you want to resolve or the area you want to investigate. Use the list of suggestions provided by Operations Analytics for guidance to continue typing or select from the list of suggestions.

The Operations Analytics Phrased Query Language (PQL) dynamically incorporates the information described in the PQL Suggestions table to guide you as you define the context of the problem you are trying to solve.

Note: Operations Analytics organizes suggestions by category. See PQLSuggestions for possible categories.

PQL Suggestions

Suggestion Types	Provided by
Text strings used to identify either a collection or specific metric or attribute. These text strings are known as tags.	Operations Analytics and your Operations Analytics administrator
Text strings that represent column names. For example, you might want Operations Analytics to return all values for **cpu_util** for a specified host. **Note:** Precede any column name with the tag of interest. See "Filter the Scope of a Phrased Query Language (PQL) Query " on page 65 for more information.	Operations Analytics and your Operations Analytics administrator

PQL Suggestions, continued

Suggestion Types	Provided by
Text strings that are values for columns defined as keys. Operations Analytics uses columns defined as keys to understand topology relationship across collections. These suggestions use the following syntax: withkey <*key_column_value_1*>, <*key_column_value_2*>,<*key_column_value_3*> When selecting key column values in your search, note the following: • Up to three key column values can be included in each **withkey** combination for a single query. For example, both of the following queries are valid: **withkey myhost.enterprise.com, oracledbhost1, private** **withkey host1,instance1 withkey state1** When you enter a query that includes multiple column values Operations Analytics displays information for only those records (rows) that match all values. Using the first example, Operations Analytics returns the data requested for any record (row) that includes both **myhost.enterprise,com** and **oraclehost1** and **private**. • A key column value that includes spaces is enclosed in quotes; for example: **withkey "my instance1"** • A key column value that includes an asterisk is defining a pattern for which Operations Analytics should search. It is most useful for defining a host name pattern. For example, the following key column value finds all host names that include **enterprise**: **withkey *enterprise** • A single query can contain column values that include quotes, do not include quotes, or that include both types of values; for example: **withkey oracledbhost, "my instance"** See "About Keys and Link Tags" on page 174 and "Filter the Scope of a Phrased Query Language (PQL) Query " on page 65 for more information.	Operations Analytics and your Operations Analytics administrator

PQL Suggestions, continued

Suggestion Types	Provided by
When querying for topology service information, you can also use key column values to narrow your search to return information for only a specified group or a specified instance within a group. To do so, use the **filtering** keyword as shown in the following syntax: service withkey <*service_name*> filtering <*node_grouptier_tag*> withkey <*group_name_value*> The following example returns all information for MyService1: **service withkey MyService1** The following example returns all information for the group configured for MyService1: **service withkey MyService1 filtering groups** The following example returns the related groups information for only the instance named **groupName1**: **service withkey MyService1 filtering groups withkey groupName1** See "About Keys and Link Tags" on page 174, and "Filter the Scope of a Phrased Query Language (PQL) Query " on page 65 for more information.	Operations Analytics and your Operations Analytics administrator
Search history **Note:** When using your search history, Operations Analytics orders the suggestions according to historical frequency. For example, searches that have been entered more often and more recently, appear closer to the top of the list.	The Operations Analytics user community (Tenant)

See Suggestion Categories for the maximum number of suggestions that Operations Analytics displays per category.

Suggestion Categories

Category	Maximum Number of Suggestions
Metadata	15
Search History	5

See "About Phrased Query Suggestions" on the next page for more information.

Click here for information about how Operations Analytics is able to provide suggestions.

Each time you press [**Enter**] to execute a search query, Operations Analytics does the following:

Tip: Selecting a time range also executes your search query. See "Dashboards and Query Panes" on page 13 for more information.

- Stores your search history.
- Uses the meta data information stored about the metrics, topology, inventory, event, log files, and tags to understand and identify patterns, including synonyms. For example, if Oracle, db, and database are valid entries in the meta data information, Operations Analytics recognizes each of the following search query entries and determines you want to obtain information about one or more Oracle databases: `oracle`, `oracles`, `db`, or `database`.

The results of each phrased query is an Operations Analytics dashboard. See "Dashboards and Query Panes" on page 13 for more information.

For more information about PQL, see the topics described in the following table.

Additional Topics that Might be of Interest

Information	Topic
The types of words or phrases to include in your phrased query (for metrics).	"Elements in a Phrased Query (for Metrics)" on page 57
The types of words or phrases to include in your phrased query (for log files).	"Elements in a Phrased Query (for Log Files)" on page 64
The types of suggestions Operations Analytics provides.	"About Phrased Query Suggestions" below
Examples of phrased queries mapped to a possible task and the type of results to expect for each.	"Phrased Query Examples" on page 67

About Phrased Query Suggestions

When you begin to type a Phrased Query Language (PQL) search query, Operations Analytics dynamically incorporates the meta data and search history information available to guide you as you type.

Note: To access the PQL suggestions described in this topic, select **Start typing:** from the Search Query field options and type a word or phrase that begins to describe the type of problem you want to resolve or the area you want to investigate. Use the list of suggestions provided by Operations Analytics for guidance to continue typing or select from the list of suggestions.

Click each suggestion category for more information.

Note: As you type, Operations Analytics finds the character sequence anywhere within the possible suggestions list. Only suggestions that narrow your search are displayed. For complete PQL search queries, see PQL Example Queries (for Metrics) and "Phrased Query Examples" on page 67.

Meta Data

Operations Analytics also tries to match your query to meta data using the methods described in the following **Meta Data Suggestions** table.

Note: Operations Analytics stores metrics, topology, inventory, log file, and event information in the form of collection tables. These collection tables are also known as property groups. The columns that represent the metrics collected and that store values within these tables are also known as properties. A property can be either an **attribute**[1] or a **metric**[2].

Meta Data Suggestions

Suggestions	Description	Example Phrases
Column names	Operations Analytics looks for any column names that begin with the phrase you specify.	If you type **util**, Operations Analytics looks for any column name that includes **util** (for example, **cpu_ util** and **mem_util**). If you use a column name, Operations Analytics also checks for topology relationships. See **Column values** in this table.
Tags associated with collections or collection columns (**metrics**[3] or **attributes**[4])	Operations Analytics looks for any tags associated with collection or column names that begin with the phrase you specify.	If you enter the character string **perf,** Operations Analytics suggestions include any tag combinations that include **perf** (for example, **performance** and **network performance**).

[1]A descriptor stored in a collection for an entity, such as host_name.
[2]Typically a measurement stored in a collection. For example, CPU utilization.
[3]Typically a measurement stored in a collection. For example, CPU utilization.
[4]A descriptor stored in a collection for an entity, such as host_name.

Meta Data Suggestions, continued

Suggestions	Description	Example Phrases
Column values	Operations Analytics looks for the value in the key columns for each applicable collection.	Operations Analytics suggestions include values for columns configured as keys. These key columns can be used to define topology relationships. For example, **withkey oracleinst1** indicates that you want Operations Analytics to search across collections for only information related to **oracleinst1**. When selecting suggestions that contain column values, note the following: • Up to three key column values can be included in each **withkey** combination for a single query. For example, both of the following queries are valid: **withkey myhost.enterprise.com, oracledbhost1, private** **withkey host1,instance1 withkey state1** When you select a suggestion that includes multiple column values Operations Analytics displays information for only those records (rows) that match all values. Using the first example, Operations Analytics returns the data requested for any record (row) that includes both **myhost.enterprise,com** and **oraclehost1** and **private**. • A key column value that includes spaces is enclosed in quotes; for example: **withkey "my instance1"** • A key column value that includes an asterisk is defining a pattern for which Operations Analytics should search. It is most useful for defining a host name pattern. For example, the following key column value finds all host names that include **enterprise**: **withkey *enterprise** • A single suggestions can contain column values that include quotes, do not include quotes, or that include both types of values; for example:

Meta Data Suggestions, continued

Suggestions	Description	Example Phrases
		withkey oracledbhost, "my instance" See "Elements in a Phrased Query (for Metrics)" on the next page and "Filter the Scope of a Phrased Query Language (PQL) Query " on page 65 for more information.
Service topology information	The keyword **service** indicates you want Operations Analytics to return only the data related to the service you specify. **Note:** The service must be configured using the **Topology Management** option from the Settings menu. See "Topology Manager" on page 172 for more information.	To query for only the data related to a specified service, select the keyword **service**. You define services using the **Topology Management** option from the Settings menu. When defining a service, you specify each group that is associated with the service. You can query for all information available for a specified group. The following example returns all of the groups information related to the service named **MyService1:** **service withkey MyService1 filtering groups** When querying for service information, you can also narrow your search to return information for only a specified instance within a group. To do so, select the **filtering** keyword. The following example returns the related group information for only the instance named **groupName1**: **service withkey MyService1 filtering groups withkey groupName1**

Search History

Operations Analytics tries to match your query to your search history using the methods described in the following **Search History Suggestions** table.

Search History Suggestions

Suggestions	Description	Example Phrases
Search history queries	Operations Analytics looks for any phrases you have used previously. When using your search history, Operations Analytics orders the suggestions according to historical frequency. For example, searches that have been entered more often and more recently, appear closer to the top of the list.	If you type **my**, Operations Analytics might include the phrase **myhost.co.com** if it was used recently in a previous search.

Also see "About the Phrased Query Language (PQL)" on page 50

Elements in a Phrased Query (for Metrics)

The Phrased Query Language (PQL) is comprised of the following types of elements to query metric or other structured collections.

Note: To access the PQL described in this topic, select **Start typing:** from the Search Query field options and type a word or phrase that begins to describe the type of problem you want to resolve or the area you want to investigate. Use the list of suggestions provided by Operations Analytics for guidance to continue typing or select from the list of suggestions.

Tip: You can use **log** key word in any query to include log messages in the query results. See "Elements in a Phrased Query (for Log Files)" on page 64 for more information.

Tags

Note: When using tags in your PQL search, you must use the tags configured for use by your Operations Analytics administrator or provided by Operations Analytics. If you do not know the list of tags available, see "How to View Collection Information" on page 217. Operations Analytics also includes possible tags in the list of suggestions.

Type a tag name that describes the problem you are investigating. Operations Analytics displays suggestions as you type.

Note: Tags are not case sensitive. They are always converted into lower case when stored.

Tip: When determining the tags to enter, think in terms of the **entity** (software and hardware) , the type of **metrics** you want to view for the entity, and the **analysis** you want Operations Analytics to apply to the metric. See the Types of Information to include in PQL table for more information.

Entity Instance

Represents the hosts or host for which you want information. Entity instance might also be an application or database instance.

To specify an entity instance using an column value.

Column Name

Column name is the name of a column in a collection. For example, a collection might include **host_name** or **cpu_util** as the column name.

When you include the name of a column in your search, Operations Analytics returns the metrics stored in that column.

Column Values

PQL enables you to include a column value (for example, the name of a database instance) in your

search query. Any column value that you provide must reside in a column that is configured as a key. It can also be part of a link tag configuration in the collection.

To include a key column value in your search query, use the following syntax:

`withkey <key_column_value_1>, <key_column_value_2>,<key_column_value_3>`

For example: `withkey *enterprise.com,dbhostname1`

When including key column values in your search, note the following:

- Up to three key column values can be included in each **withkey** combination for a single query. For example, both of the following queries are valid:

  withkey myhost.enterprise.com, oracledbhost1, private

  withkey host1,instance1 withkey state1

  When you enter a query that includes multiple column values Operations Analytics displays information for only those records (rows) that match all values. Using the first example, Operations Analytics returns the data requested for any record (row) that includes both **myhost.enterprise,com** and **oraclehost1** and **private**.

- A key column value that includes spaces is enclosed in quotes; for example:

  withkey "my instance1"

- A key column value that includes an asterisk is defining a pattern for which Operations Analytics should search. It is most useful for defining a host name pattern. For example, the following key column value finds all host names that include **enterprise**:

  withkey *enterprise

- A single query can contain column values that include quotes, do not include quotes, or that include both types of values; for example:

  withkey oracledbhost,"my instance"

To include any column value in your search, use the following syntax:

<tag> <column_name>=<column_value>

For example: **database dbname=vnode1**

See "How to View Collection Information" on page 217 for more information about how to view the keys and link tags available in your IT environment.

Click here for more information about **keys**.

The Operations Analytics administrator can specify one or more columns in a collection as a **key**. Keys identify columns in a collection that you want Operations Analytics to use to match metrics for one entity (collection row) to the same or related entity (collection row) across collections. For example, your Operations Analytics administrator might find that host_name is an attribute that identifies the host in most of the collections. However, perhaps in one or two collections, server_

name is the attribute used to identify the host. In this scenario, the Operations Analytics administrator specifies **host_name** as a key column in the collections that include the host_name attribute and **server_name** as a key column in the collections that include server_name. When you enter a host name value in your PQL search, Operations Analytics looks for that value in all key columns across collections.

Key columns can also be used to narrow a search within a single collection. When using a key column to search within only one collection, Operations Analytics returns only those metrics for the specified key column value. For example, if the **host_name** column is defined as a key in a cpu metrics collection, the host_name key column enables you to search for cpu metrics for a specific host name.

To specify a column value for a column that is configured as a key, use the `withkey` keyword. See the Types of Information to include in PQL table for more information.

Keys also enable you to filter the scope of a search using link tags. They are used when you want to search custom collections using topology relationships to specify the metrics you want Operations Analytics to return. For example, you might want to search multiple collections for only metrics related to a database instance for an application. See "About Keys and Link Tags" on page 174 for more information.

To specify a column value that is associated with a link tag, use the `withkey` and `filtering` keywords in your search.

Click here for more information about **link tags**.

Link tags are special tags that associate two collections. They establish relationships between one or more columns defined as keys . Values contained in a key column can then be used to filter one collection by the instances in another collection.

Link tags are useful when you have a custom collection that stores topology information and you want to filter additional collections to return only the data related to a specified host or application instance.

See "About Keys and Link Tags" on page 174 for more information.

To view the collections, keys and tags available select the **SystemMetaInfo** dashboard. For more information, see "How to View Collection Information" on page 217.

Note: For complete PQL search queries, see PQL Example Queries (for Metrics) and "Phrased Query Examples" on page 67.

Types of Information to include in PQL

Category	Description	Example Phrases
Entity Tags	Represents the type of object or the instance of the object or objects for which you want information. Entity classes can have the following relationships: • An entity can be contained in another. For example an application resides on a host. • An entity can depend on another. For example, an application might depend on a database. • When entities have a relationship, you can use both entities in your query; for example **host interface** or **application database**,	**host**, **application**, **database**, **interface**
Entity Instance	Represents an instance of the object or objects for which you want information. Operations Analytics supports specifying entity instances using the following methods: • Enter a column value.	**withkey eth0 server_ name=myoracledb**
Metric Tags	Represents the metric or set of metrics from the collection that you want to include in your results. To view the collection, tag, and metric information defined for your IT environment, see "How to View Collection Information" on page 217.	**performance**, **change**, **error**, **disk utilization**, **memory performance** **Tip:** You can also combine tags; for example: **cpu utilization**
Analytic Tags	Your Operations Analytics administrator can also use tags to identify the metrics that have values that assist with a particular kind of analysis, such as **status** or **health**. The analytic tag you provide must be configured as a tag in one or more collections.	**status**, **health**
Column Name	When you want all values for a particular attribute (column), include the column name in your search	**cpu_util**

Types of Information to include in PQL, continued

Category	Description	Example Phrases
Column value	A column value can be specified in either of the following ways: • <tag> <*column_name*>=<*column_value*> For example: `cpu dbname=vnode1` **Tip:** When using multiple <*column_name*>=< *column_value*> entries in the same search, **do not** separate each entry using commas. • <tag> withkey <*key_column_value_1*>, <*key_column_value_2*>,<*key_column_value_3*> `cpu withkey oracleinstance1,virtual` When you specify a column value, the column must be configured as a key. **Tip:** You can include <*column_name*>=<*column_value*> and withkey <*key_column_value*> in the same search. See "Filter the Scope of a Phrased Query Language (PQL) Query " on page 65 for more information.	**server_name=server1.enterprise.com withkey oracleinstance1, virtual**

Types of Information to include in PQL, continued

Category	Description	Example Phrases
Topology service information	The keyword **service** indicates you want Operations Analytics to return only the data related to the topology service you specify. **Note:** The topology service must be configured using the **Topology Management** option from the Settings menu. See "Topology Manager" on page 172 for more information.	To query for only the data related to a specified topology service, select the keyword **service**. You define topology services using the **Topology Management** option from the Settings menu. When defining a service, you specify groups that are associated with the service. You can query for all information available for a specified group. The following example returns all of the database groups related to the service named **MyService1:** **service withkey MyService1 filtering database** When querying for service information, you can also narrow your search to return information for only a specified instance within a group. To do so, select the **filtering** keyword. The following example returns the related groups information for only the database instance named **instance1**: **service withkey MyService1 filterring dabase withkey instance1** **Note:** If the **filtering** keyword is used in the query, it can also include up to three key column values, as shown in the following example: **withkey host1 state2 withkey instance1 filtering <tag> host10,instance10,state10**

See Suggestion Categories for the maximum number of suggestions that Operations Analytics displays per category.

Suggestion Categories

Category	Maximum Number of Suggestions
Metadata	15
Search History	5

See "About Phrased Query Suggestions" on page 53 for more information.

Note: If a search query results in too many instances to display, Operations Analytics uses the query to return log file messages matching any phrases in the search query. See "Out-of-the-Box Dashboards Provided by Operations Analytics" on page 13 and "Refine Your Search Query " on page 70 for more information.

Click here to see the PQL Examples table for possible search queries.

Note: The actual phrases available in your environment depend on the tags configured by your Operations Analytics administrator. If you are an Operations Analytics administrator, see "Configuring Tenants and Collections" in the HP Operations Analytics Configuration Guide for more information about how to configure tags.

PQL Example Queries (for Metrics)

Entity Class	Metric Tag	Analytic Tag	Entity Instance or Column Value	Possible Results
database	performance primary		withkey oracleinstance1	• Primary performance metrics for the database instance oracleinstance1.
	primary		withkey *enterprise.com	• Metrics that are tagged as primary in all collections for hosts in the enterprise.com domain.
	events		severity="critical"	• Events with a severity of critical.
	network performance primary			• Primary performance metrics for all NNMi-related collections.
	memory	status	withkey 10.53.46.78	• Metric values related to memory for the host with an IP address of 10.53.46.78.
	cpu performance		withkey myhost.mycompany.com	• CPU performance metrics values for the host myhost.mycompany.com.

PQL Example Queries (for Metrics), continued

Entity Class	Metric Tag	Analytic Tag	Entity Instance or Column Value	Possible Results
host		status		• Metrics values tagged with **status** for all hosts in the IT environment.
	memory utilization			• Memory usage for all hosts in the IT environment.
database		health	withkey myhost	• Database health metrics for myhost.
host		health		• Health metrics for all hosts in the IT environment.
	cpu utilization			• CPU utilization metrics for all hosts in the IT environment.

You can include additional keywords and tags to use for filtering your query. See "Filter the Scope of a Phrased Query Language (PQL) Query " on the next page for more information.

To view the required syntax for search log files, see "Elements in a Phrased Query (for Log Files)" below

To view more examples organized by task, see "Phrased Query Examples" on page 67.

Note: To filter your query by time, you do not need to use a search query. See "Filter Search Query Results " on page 161 for more information.

Elements in a Phrased Query (for Log Files)

Note: To access the PQL described in this topic, select **Start typing:** from the Search Query field options.

The Phrased Query Language (PQL) uses the following syntax to query log files:

```
"<string>"
```

```
log ("<string> AND|OR <string>")
```

Note the following:

- To include quotes within your search query, precede each quote with the backslash character.
- Log file queries for log files configured in HP ArcSight Logger are not used for metric searches.

- If you have multiple HP ArcSight Logger servers configured, Operations Analytics searches for the text string or strings in the log files of each server.
- You can also include tags in your log queries. For example, `system log("severity AND critical")` finds all metrics tagged system and log file messages containing **severity** and **critical**.

Click here for example log file queries for log files that are configured in HP ArcSight Logger:

PQL Example Queries (for Logs)

Log File Message String	Example Log File Queries	Results
connection error	log ("connection error")	• Error messages containing the text string **connection error.** • Log file entries containing the text string **connection error**.
connection error "	log ("connection error\"")	• Error messages containing the text string **connection error** followed by a value that begins with the quote (") character (for example **connection error "port 80"**) • Log file entries containing the text string **connection error**.followed by a value that begins with the quote (") character (for example **connection error "port 80"**
severity AND critical	log("severity AND critical")	• Error messages containing the text string **severity** and **critical**. • Log file entries containing the text string **severity** and **critical**.
error OR warning	log("error OR warning")	• Error messages containing the text string **error** or **warning**. • Log file entries containing the text string **error** or **warning**.

Filter the Scope of a Phrased Query Language (PQL) Query

Note: To access the PQL described in this topic, select **Start typing:** from the Search Query field options and type a word or phrase that begins to describe the type of problem you want to resolve or the area you want to investigate. Use the list of suggestions provided by Operations Analytics for guidance to continue typing or select from the list of suggestions.

You can filter the scope of your Phrased Query Language (PQL) search by specifying one or more of the following:

- Collection column name and the value that you want to use to filter the results
- Column value

Note: To filter your query by time, you do not need to use a search query. See "Filter Search Query Results " on page 161 for more information.

To specify a filter in a PQL query, use the syntax described in Filter PQL Queries. See "Elements in a Phrased Query (for Metrics)" on page 57 for more information.

Click here for information about using pattern matching in your PQL query.

Use the asterisk when you want Operations Analytics to match a text string pattern. The asterisk can appear anywhere within the text string. See Pattern Matching in PQL Queries for examples.

Pattern Matching in PQL Queries

Example	Results
"*.co.enterprise.com"	Returns results for all host names that end with **co.enterprise.com**
"co.*.com"	Returns results for all host names that begin with **co** and end with **.com** (for example: **co.fc.com** and **co.denver.com**).

Note: To view the collections available, select the **SystemMetaInfo** dashboard from the Dashboards menu. See "How to View Collection Information" on page 217 for more information.

Filter PQL Queries

Filter	Syntax	Example
tag column name and value	<tag> <*column name*>=<*column value*>	`database dbname=vnode1`

Filter PQL Queries, continued

Filter	Syntax	Example
column value using a column configured as a key **Tip:** Use this search when you don't know the column name. Your Operations Analytics administrator can configure a maximum of three key columns per collection. **Note:** You can include up to three key column values in the same <tag> withkey search.	<tag> **withkey** <*key_column_value_1*>, <*key_column_value_2*,<*key_column_value_3*	`database withkey oracleinstance1`
filter collections using the **filtering** keyword	**service withkey** <*service_name*> **filtering** <*node_grouptier_tag*> **service withkey** <*service_name*> **filtering** <*node_grouptier_tag*> **withkey** <*group_name_value*>	`service withkey MyService filtering groups`

Click here for example PQL queries:

Example PQL Queries Filtered by Scope

PQL Query	Results
`oracle performance withkey *enterprise.com`	Display all metrics associated with the tags **oracle** and **performance** for all host names in the ***.enterprise.com** domain.
`cpu_util withkey *enterprise.com`	Display the values for the **cpu_util** metric for all host names in the ***enterprise.com** domain.
`opsa withkey *enterprise.com,instance1`	Display the metrics associated with the tag **opsa** for the local host and for all hosts in the enterprise.com domain.
`service withkey MyService filtering groups withkey groupName1`	Used to filter the collection of database metrics for MyService. Displays only results for the database metrics for the group named **groupName1**.

Phrased Query Examples

Note: To access the PQL described in this topic, select **Start typing:** from the Search Query field options and type a word or phrase that begins to describe the type of problem you want to

resolve or the area you want to investigate. Use the list of suggestions provided by Operations Analytics for guidance to continue typing or select from the list of suggestions.

This help topic includes examples of phrased queries that you might use for basic troubleshooting tasks. See "Elements in a Phrased Query (for Metrics)" on page 57 for more information about the words and phrases to include in a Phrased Query as well as additional examples.

Note the following:

- For the purposes of this example, the example host name is `myhost.enterprise.com` and the Los Angeles office domain is `la.enterprise.com`
- The metrics displayed are filtered according to the time range you specify.
- The actual words and phrases available in your environment depend on the tags and phrases configured by your Operations Analytics administrator.

  Operations Analytics includes this information under the **Metadata** suggestions category. See "About Phrased Query Suggestions" on page 53 for more information.

  You can also view meta data information for your **collections**[1] from the **SystemMetaInfo** dashboard. See "How to View Collection Information" on page 217 for more information.

Example PQL Queries (by Task)

Task	Example Query
View the overall status metrics for myhost.enterprise.com	`status withkey myhost.enterprise.com`
View the overall status for all systems in the Los Angeles office	`status withkey *la.enterprise.com`
View the cpu speed metrics for myhost.enterprise.com	`cpu speed withkey myhost.enterprise.com`
View the cpu utilization for all systems in the IT environment	`cpu_util`
View the cpu utilization for all systems in the Los Angeles office	`cpu_util withkey *la.enterprise.com`
View memory metrics for all databases in the IT environment	`memory database`
View CPU bottlenecks for all systems in the Los Angeles office.	`cpu queue withkey *la.enterprise.com`

[1]Operations Analytics stores metrics, topology, inventory, log file, and event information in the form of collection tables. Each collection is associated with a database table in which an Operations Analytics Collector stores the data collected.

Example PQL Queries (by Task), continued

Task	Example Query
View the metrics for all servers in the IT environment that are associated with both performance and configuration tags.	`performance configuration`
View only results for database metrics for OracleService	`service withkey MyService filtering database`

For more examples, see:

"Elements in a Phrased Query (for Metrics)" on page 57

"Elements in a Phrased Query (for Log Files)" on page 64

"Filter the Scope of a Phrased Query Language (PQL) Query " on page 65

Searching for a Service Defined in Topology Manager

After you have defined a service, it can be referenced in searches and resulting dashboards. For details on defining services, see "Topology Manager" on page 172.

For example, suppose you have defined a service called MyService, as follows:

- This service is made up of the groups **MyWebServers**, **MyAppServers**, and **MyDBServers**.
- These groups are made up of **WebHost1-3**, **AppHost1-3**, and **DBHost1-3** respectively.

You can now execute the following searches:

- `Service: "MyService"`. This search returns a dashboard with information regarding the different hosts in all the groups that are part of the **MyService** service, with their events and logs.
- `Service: "MyService" Drill To: "MyWebServers"` - This search returns a dashboard with data on all the hosts that belong to the **MyWebServers** group in the service, including metrics, events and logs.

Note: You can also use a host-based search (for example `Host: "WebHost1"`) to then focus on a specific host that seems to have issues.

These different searches provide you with a drill-down capability. When you look at the service, you can pinpoint the group or in some cases the specific host that may be causing the issue. When you look at a group you can quickly focus on a specific host that exhibits problems. The final drill-down to a specific host helps you pinpoint the root cause of the problem.

To learn more about searching in Operations Analytics, see "Search Tool" on page 26.

To learn more about interpreting dashboard information, see "Dashboards and Query Panes" on page 13.

Refine Your Search Query

Note: To access the PQL described in this topic, select **Start typing:**from the Search Query field options and type a word or phrase that begins to describe the type of problem you want to resolve or the area you want to investigate. Use the list of suggestions provided by Operations Analytics for guidance to continue typing or select from the list of suggestions.

Each time you enter a Phrased Query Language (PQL) query, Operations Analytics displays the default visualizations for the metrics, topology, inventory, event, and log file information available in response to your search query. See "Out-of-the-Box Dashboards Provided by Operations Analytics" on page 13 for more information about dashboards that Operations Analytics provides. To change the dashboard results, see "Dashboards and Query Panes" on page 13.

If a search query results in too many instances to display, Operations Analytics uses the query to return only the log file messages matching any phrases in the search query. This is because it is not able to display the metrics results due to potential performance issues.

To refine your search query:

1. Return to your search query, review the list of suggestions, and select a suggestion from the list that continues to refine your query. You can also refine your query manually by filtering the query scope using the methods described in "Filter the Scope of a Phrased Query Language (PQL) Query " on page 65.

   Tip: To begin with a new search query, review the help topics described in Additional Topics that Might be of Interest. Filter the scope of your PQL query to reduce the number of entities, such as hosts, or the number of metrics returned.

2. Select an existing Dashboard from the **Dashboard** menu.

Additional Topics that Might be of Interest

Information	Topic
Basic steps for providing an Operations Analytics search query.	"Search Tool" on page 26
An overview of the requirements for Phrased Query Language as well as of the suggestions provided by Operations Analytics	"About the Phrased Query Language (PQL)" on page 50
More detailed information about Phrased Query Language Query (PQL) suggestions provided by Operations Analytics.	"About Phrased Query Suggestions" on page 53
Elements to include in a Phrased Query Language Query (PQL) to filter the query by scope, such as host name.	"Filter the Scope of a Phrased Query Language (PQL) Query " on page 65

About the Analytics Query Language (AQL)

Use the Analytics Query Language (AQL) when the Phrased Query Language (PQL) syntax is not specific enough to return the data you need. When using AQL you can be more specific about the data collected. You can also filter, group, and order the collected data in a single query.

AQL queries use a syntax similar to the ANSI Standard SQL. When using AQL, it is helpful if you have minimal knowledge of databases as well as scripting or programming skills. However, it is not mandatory to have this knowledge to get started using AQL queries.

Tip: Before you begin writing AQL queries, view the collection information that is stored in Operations Analytics to determine the kinds of data available in your environment. You will use this information as part of your AQL syntax. For details, see "How to View Collection Information" on page 217.

Note the following:

- When building AQL queries, you can also define AQL functions or expressions.
- AQL functions are a convenient way of defining and naming frequently used AQL queries for reuse. When you define the function, you define the associated AQL query as well as the argument values to pass to that AQL query. See "About Analytics Query Language (AQL) Functions" on page 146 for more information.

  Click here for more information about AQL functions.

  Operations Analytics provides a set of AQL functions for your use. You can browse them in Operations Analytics dashboard when adding or editing a query pane. See "Add / Edit Query Pane - Query Tab" on page 24 for more information.

  You can define your AQL functions using a text editor and then import them in Operations Analytics. See "Define Analytic Query Language (AQL) Functions" on page 233 and "Import Analytic Query Language (AQL) Functions" on page 234 for more information

  You can specify an AQL query, an AQL function or an AQL expression when adding or editing a dashboard query pane. See "Dashboards and Query Panes" on page 13 for more information.

- AQL expressions include one or more AQL functions. Use AQL expressions when you want the results of multiple queries to be combined in to a single query pane in a Operations Analytics dashboard. See "About Analytics Query Language (AQL) Expressions" on page 149 for more information.

  Click here for more information about using AQL functions in AQL expressions.

  You can use AQL functions in an AQL expression in any of the following ways:

  - Use a single AQL function.
  - Concatenate the results of multiple AQL functions.

  - Use the results of one AQL function as an input filter for another AQL function.

- Operations Analytics provides a number of analytic functions that encapsulate more complex SQL expressions and notations. You can include these analytic functions in AQL queries without requiring SQL and database expertise. See "Analytic Functions Provided by Operations Analytics" on page 100 for more information.

Note: Operations Analytics also generates AQL in response to a Phrased Query Language (PQL) query. See "About the Phrased Query Language (PQL)" on page 50 for more information.

About the Analytics Query Language (AQL) Syntax and Structure

The basic structure of an AQL query is a series of clauses. The clauses you include depend on the type, organization, and order of the information you want Operations Analytics to return. It also depends on the time range and type of analysis, if any, you want Operations Analytics to apply to the data.

When positioning the clauses in an AQL query, note the following

- The **from i in…** clause must be the first clause
- The **select** clause must be the last clause.
- You can include all other clauses in any order within the **from i in ...** and **select** clauses.

Click each of the choices listed for the associated clause and its required syntax.

Note: You can also specify additional optional parameters using let clauses. See **"About Parameters Used in an Analytics Query Language (AQL) let clause" on page 84** for more information.

Identify the collection of metrics and their attributes

Use the **from i in …** clause to identify the collection of metrics and the associated attributes to be queried.

The **from** clause supports the syntax described in The from clause syntax.

The from clause syntax

Syntax	Parameter	Examples
from i in (<collection_unique_id>)	<collection_unique_id> represents the unique identifier for a collection of metrics and associated attributes.	**oa_sysperf_global** is the unique identifier for the system performance metrics collected by the HP Operations Agents. To query this collection, use the following from clause: **from i in (oa_sysperf_global)**
from i in tags (<comma_separated_list_of_tags>)	<comma_separated_list_of_tags> represents the set of tags used to identify the metrics in one collection or across multiple collections	The tag **cpu** is applied to a subset of metrics in the **oa_sysperf_global** collection, which measures cpu related system behavior. Attributes such as host_name also receive this tag. **performance** is another tag applied to multiple metrics and their associated attributes in multiple collections. This tag identifies the metrics related to performance. To query these metrics with their associated attributes, use the following from clause: **from i in tags(cpu,performance)**

You can also append the **let** clause to the **from** clause to indicate either of the following

- Return metrics and attributes which are tagged with **both** tag values (for example, **cpu** and **performance**) (intersection)
- Return metrics and attributes which are tagged with **either** tag values (for example, **cpu** or **performance**) (union)

When used with the from clause, the let clause supports the syntax described in the let clause syntax to specify tag intersection or union in a from clause.

The let clause syntax to specify tag intersection or union in a from clause

Syntax	Parameter	Examples
`let tags_intersect=Y|N`	Y - intersection N - union	from i in tags(cpu, performance) let tags_intersect=Y from i in tags(cpu,performance) let tags_intersect=N

Note: If you do not include `let tags_intersect` in the from clause, Operations Analytics returns the union.

Specify a time filter

Use the **let** clause to specify the window of time or time range filter to use.

Note: The time filter is mandatory for any query that returns metrics or includes an analytic function. The time filter is not required for queries that return only attribute values.

When used to specify a time window or time range filter, the let clause supports the syntax described in the table below.

The let clause syntax to specify a time filter

Syntax	Description	Examples
`let analytic_interval=since(<time_period_in_seconds>)`	Specify a time filter that goes back a specified time period from the current time	To specify a time filter that goes back by 24 hours from the current time, use the following let clause: `let analytic_interval=since(86400)`
`let analytic_interval=between (<starttime_as_seconds_since_epoch>, <endtime_as_seconds_since_epoch>)`	Specify a time filter that has an absolute start and end time To use the time range or time line specified in the Operations Analytics console, use $starttime, $endtime, or both as shown in the following example: `let analytic_interval=between ($starttime,$endtime)` See "Dashboards and Query Panes" on page 13 for more information.	To specify 1375197291 seconds since epoch to be the start time and 1375370091 seconds since epoch to be the end time, use the following let clause: `let analytic_interval=between (1375197291,1375370091)`

Specify a set of other data filters

Use the **where** clause to apply a set of data filters.

The where clause supports the syntax described in The where clause syntax.

The where clause syntax

Syntax	Parameter	Examples
`where (filter_expression)`	filter_expression is ([left_operand] relation_operator right_operand) left_operand or right_operand can be any of the following: • any attribute or metric • an analytic function applied on a metric • an analytic function applied on an attribute or a constant • filter_expression in which the relation_operator is on the logical operators !, ||, &&. relation_operator can be one of the comparison predicates: !=, ==, >, >=, <, <=, like, ilike **Tip:** Use ilike to indicate the filter is case insensitive.	**/*Attribute filters combined*/** from i in (omi_events_omievents) let analytic_interval = between ($starttime, $endtime) let interval = $interval **where ((i.hostinfo_dnsname like "*mydomain.com") && ((i.severity == "CRITICAL") \|\| (i.severity ilike "WARN*")))** group by i.hostinfo_dnsname, i.severity select moving_count(i) **/*Attribute filters, metric filters combined*/** from i in (oa_sysperf_global) let analytic_interval = between ($starttime, $endtime) let interval = $interval **where ((i.host_name like "*.mydomain.com") && (i.cpu_util >= 0.80))** select i.host_name, i.timestamp, i.cpu_util **/*Attribute filters, metric analytic filter combined*/** from i in (oa_oraperf_graph) let analytic_interval = between ($starttime, $endtime) let interval = $interval **where (((i.host_name like "*.mydomain.com") && (i.db_instance_name like "*")) &&** **(moving_max(i.max_num_sessions) > 10))** group by i.host_name, i.db_instance_

The where clause syntax, continued

Syntax	Parameter	Examples
		name select moving_avg(i.max_num_ sessions)

Choose a set of attributes to group query results

Use the **group by** clause to group the query results

Note the following:

- When querying raw metrics or attributes,Operations Analytics ignores any group by clause.
- Specify group by attributes list only when your query includes an analytic function.
- Operations Analytics ignores the following group by fields:
  - Any field defined as a metric in the meta data.
  - The mandatory **timestamp** field defined in the meta data.
- Operations Analytics automatically selects any attribute specified in the group by clause and includes those attribute values in the query results.

The group by clause supports the syntax described in group by clause syntax.

The group by clause syntax

Syntax	Parameter	Examples
`group by i.attribute1uniqueid, i.attribute2uniqueid, …`	Group the results by the attributes whose unique identifiers (for example atribute1uniqueid) are specified. See "About Meta Data" on page 178 for more information.	**/*group by one attribute in queried collection*/** from i in (oa_sysperf_global) let analytic_interval = between ($starttime, $endtime) let interval = $interval where (i.host_name like “*.mydomain.com”) **group by i.host_name** select moving_avg(i.cpu_run_queue) **/*group by multiple attributes in queried collection*/** from i in (oa_sysperf_global) let analytic_interval = between ($starttime, $endtime) let interval = $interval where (i.host_name like “*.mydomain.com”)

The group by clause syntax, continued

Syntax	Parameter	Examples
		group by i.host_name , i.source select moving_ avg(i.cpu_run_ queue)
group by i	Group the results by all attributes in the collection identified in the **from i in..** clause	**/*group by all attributes in the queried collection*/** from i in (oa_ sysperf_global) let analytic_ interval = between ($starttime, $endtime) let interval = $interval where (i.host_ name like “*.mydomain.co m”) **group by i** select moving_ avg(i.cpu_run_ queue)

Select a set of attributes, metrics, or both to be queried

Use the select clause to specify the attributes and metrics to be selected

Note the following:

- If the select clause contains only attributes, Operations Analytics returns the distinct values of the specified attributes.

  When using this type of query, the **let analytic_interval …** clause is optional.

- If the select clause contains a set of metrics, Operations Analytics returns the raw time series of metric values. If attributes are also included in the select clause, Operations Analytics also returns the values of those attributes.

- Do not specify group by to select a set of attributes or metrics that do not have an analytic function applied to them.

The select clause supports the syntax described in select clause syntax.

The select clause syntax

Syntax	Parameter	Examples
`select i.property1uniqueid, i.property2uniqueid`	The property1uniqueid, property2uniqueid… are the unique meta data identifiers of the attributes or metrics in the collection identified in the **from i in…** clause.	**/*selecting attributes only to get distinct values of attributes from a topology collection and without time filter*/** from i in (opsa_topology) let limit = $limit where (i.service_name like "*opsaservice") **select i.opsa_server_name, i.vertica_node, i.collector_server_name, i.logger_server_name** **/*selecting attributes only to get distinct values of attributes from an events collection and with time filter*/** from i in (omi_events_omievents) let limit =$limit let analytic_interval=between ($starttime, $endtime) where ((i.hostinfo_dnsname like "*mydomain.com") && (i.severity=="CRITICAL")) **select i.hostinfo_dnsname, i.timestamp, i.state, i.category, i.title**

Select a set of analytic functions to apply to the metrics

Also use the select clause to specify one or more analytic functions to apply to the **metrics**[1] and the **attributes**[2] specified in your AQL search query. See "Analytic Functions Provided by Operations Analytics" on page 100 for more information about analytic functions.

See Guidelines for using the select clause with analytic functions for requirements when using the select clause with one or more analytic functions.

Guidelines for using the select clause with analytic functions

Guideline	Examples
Do not mix overall aggregate analytic functions, moving aggregates analytic functions, or **raw metrics**[3] in the same select clause.	**Note:** The following examples are **invalid** AQL select clauses. select **moving_avg**(i.cpu_run_queue), **aggregate_avg**(i.cpu_run_queue), **moving_total**(i.active_processes) select **moving_avg**(i.cpu_run_queue), **aggregate_avg**(i.cpu_run_queue), **moving_total**(i.active_processes), **i.disk_io_rate**

[1]Typically a measurement stored in a collection. For example, CPU utilization.
[2]A descriptor stored in a collection for an entity, such as host_name.
[3]Metrics to which an overall aggregate or moving aggregate analytic function is applied.

Guidelines for using the select clause with analytic functions, continued

Guideline	Examples
When using the analytic functions that are applied to overall aggregate analytic functions, moving aggregate analytic functions, or raw metrics, note the following: • If you apply an analytic function to one or more overall aggregate analytic functions, do not also apply the analytic function to moving aggregate analytic functions or raw metrics • If you apply an analytic function to one or more moving aggregate analytic functions, do not also apply the analytic function to overall aggregate analytic functions or raw metrics. • If you apply an analytic function to raw metrics, do not also apply the analytic function to overall aggregate or moving aggregate analytic functions.	**Note:** The following examples are **invalid** AQL select clauses. select **moving_avg**(i.cpu_run_queue)**, topN**(i.cpu_run-queue) select **aggregate_avg**(i.cpu_run_queue), **topN(moving_avg(i.cpu_run-queue)**) select i.cpu_run_que, **topN(aggregate_avg**(i.cpu_run-queue))
You can combine multiple moving aggregate analytic functions in a single select clause. Similarly, you can combine multiple overall aggregate analytic functions in a single select clause.	**Note:** The following examples are **valid** AQL select clauses: select **moving_avg**(i.cpu_run_queue), **moving_max**(i.cpu_run_queue), **moving_avg**(i.cpu_util), **moving_min**(i.cpu_util), moving_total(i.active_processes) select **aggregate_avg**(i.cpu_run_queue), **aggregate_max**(i.cpu_run_queue), **aggregate_avg**(i.cpu_util), **aggregate_min**(i.cpu_util), **aggregate_total**(i.active_processes)

Guidelines for using the select clause with analytic functions, continued

Guideline	Examples
Do not mix the following analytic functions with any other type of overall aggregate analytic functions, moving aggregate analytic functions, or raw metrics: • aggregate_count • moving_count • aggregate_distinct_count • moving_distinct_count	
When using the analytic functions that are applied to overall aggregate and moving aggregate analytic functions, apply the analytic function to only one of the following in the same select clause: • aggregate_count • moving_count • aggregate_distinct_count • moving_distinct_count	For example, if you use topN(aggregate_ distinct_count(…)) in a select clause, do not include any of the following moving count or aggregate count analytic functions in the same clause: • topN(aggregate_count(i)) • topN(moving_count(i)) • topN(moving_distinct_count(…))
Use only one of the following analytic functions in a single select clause: • aggregate_count • moving_count • aggregate_distinct_count • moving_distinct_count	

Guidelines for using the select clause with analytic functions, continued

Guideline	Examples
When using moving aggregates, specify the time interval parameter using the following **let interval = … clause** syntax: let interval=<*interval_size_in_seconds*> **Tip:** When you specify the time interval in an AQL query, make sure that the interval size is less than the number in seconds in the overall window of time specified using the **let analytic_interval=…** clause. Operations Analytics computes moving aggregates for each of the intervals.	To specify a fixed moving interval sizes of 1 hour, use the following **let** clause: let interval = 3600 To use the time interval selected in the Operations Analytics console, use the following let clause: let interval = $interval

About Parameters Used in an Analytics Query Language (AQL) let clause

Operations Analytics enables you to specify parameters in the let clause of an AQL query. These parameters are used to set limits on data returned as well as to shift or change start times, end times, or both.

Note: Operations Analytics uses default values for each of these optional parameters unless you specify a value.

Optional parameters used in the let clause describes the parameters you can choose to include in a let clause. See "About the Analytics Query Language (AQL) Syntax and Structure" on page 72 for more information about the let clause syntax as well as additional parameters used in AQL queries.

Tip: You can precede any parameter with $ (dollar sign) to specify that you want Operations Analytics to use the parameter value specified when a user adds or edits a dashboard query pane. See "Dashboards and Query Panes" on page 13 for more information.

Optional parameters used in the let clause

Parameter	Syntax	Description	Examples
limit	`let limit=<limit>`	Limits the number of rows returned in response to an AQL query. Note the following: • Use the limit parameter only in AQL queries that return distinct attribute values from events, topology, structured log, or inventory collections. • The limit parameter has no effect on all other types of queries. • The number specified as <limit> restricts the number of rows returned.	The following example query returns the 100 most recent distinct combinations of the following OMi event attributes matching the filter condition: • hostinfo_dnsname • timestamp • state • category • title attributes These attributes are stored in the omi_events_omievents collection. **from i in (omi_events_ omievents) let limit =100** let analytic_interval=between ($starttime, $endtime) where ((i.hostinfo_dnsname like “*mydomain.com”) && (i.severity==”CRITICAL”)) select i.hostinfo_dnsname, i.timestamp, i.state, i.category, i.title

Optional parameters used in the let clause, continued

Parameter	Syntax	Description	Examples
offset	`let offset=<offset>`	Determines the row at which to begin returning query results. Note the following: • Use the offset parameter only in AQL queries that return distinct attribute values from events, topology, structured log, or inventory collections. • The offset parameter has no effect on all other types of queries. • The number specified as <offset> determines the row at which to begin returning query results.	The following example query returns the 100 most recent distinct combinations of the following OMi event attributes matching the filter condition: • hostinfo_dnsname • timestamp • state • category • title These attributes are stored in the omi_events_omievents collection. The offset value specifies that Operations Analytics should start at the first row of the distinct combinations when returned the first 100 values. **from i in (omi_events_ omievents) let limit =100 let offset=0** let analytic_interval=between ($starttime, $endtime) where ((i.hostinfo_dnsname like “*mydomain.com”) && (i.severity==”CRITICAL”)) select i.hostinfo_dnsname, i.timestamp, i.state, i.category, i.title As a next step you could request for the next set of 100 distinct attribute values combinations, using this example query:

Optional parameters used in the let clause, continued

Parameter	Syntax	Description	Examples
			from i in (omi_events_ omievents) let limit =100 let offset=100 let analytic_interval=between ($starttime, $endtime) where ((i.hostinfo_dnsname like “*mydomain.com”) && (i.severity==”CRITICAL”)) select i.hostinfo_dnsname, i.timestamp, i.state, i.category, i.title

Optional parameters used in the let clause, continued

Parameter	Syntax	Description	Examples
time_offset	`let time_offset=` `<time_offset_in_seconds>`	Specifies the offset to apply to the window of time specified using the **let analytic_interval=…** clause. This parameter is useful when you want to compare metrics or their trends across a pair of time periods. **Tip:** To compare metrics or other trends, use the same AQL query for each dashboard query pane, changing only the time_offset value in each AQL query. Note the following: • A positive value of time_offset_in_seconds moves the start and end time specified in the associated let	This example query moves the time window to a week back then what is specified in **let analytic_interval=…** clause: from i in (oa_sysperf_global) **let time_offset=-604800** let analytic_interval = between($starttime, $endtime) let interval = $interval where (i.host_name like “*.mydomain.com”) group by i.host_name select moving_max(i.cpu_run_queue), moving_max(i.cpu_util), moving_total(i.active_processes) .

Optional parameters used in the let clause, continued

Parameter	Syntax	Description	Examples
		analytic_interval=… clause forward by the number of seconds used in time_offset_in_seconds. • A negative value in time_offset_in_seconds moves the times back. • The time_offset parameter is applied to both the start and end time of the time window if the absolute time filter is specified using the following let analytic_interval… clause syntax: let analytic_interval=between(<start_time_in_seconds_since_epoch>, <end_time_in_seconds_since_epoch>) • The time_offset parameter is	

Optional parameters used in the let clause, continued

Parameter	Syntax	Description	Examples
		applied only to the start time of a time window if a relative time filter is specified using the following let clause: let analytic_interval = since (<relative_time_in_seconds>)	
start_time_offset	`let start_time_offset = <start_time_offset_in_seconds>`	Moves (offsets) only the start time of the time window. **Note:** If let time_offset = <time_offset_in_seconds> is used in the same AQL query, the end time of the time window continues to be offset by the time_offset and is not affected by the start_time_offset parameter.	This example query specifies that start time should begin one week earlier than what is specified in the **let analytic_interval=…** clause. from i in (oa_sysperf_global) where (i.host_name like "*.mydomain.com") **let start_time_offset=-604800** **let analytic_interval = between($starttime, $endtime) let interval = $interval** where (i.host_name like "*.mydomain.com") group by i.host_name select moving_max(i.cpu_run_queue), moving_max(i.cpu_util), moving_total(i.active_processes)

Optional parameters used in the let clause, continued

Parameter	Syntax	Description	Examples
end_time_offset	`let end_time_offset =` `<end_time_offset_in_seconds>`	Moves (offsets) only the end time of the time window. Note the following: • If the let time_offset = <time_offset_in_seconds> is used in the same AQL query, the start time of the time window continues to be offset by the time_offset and is not affected by the end_time_offset parameter. • The end_time_offset parameter has no effect if the relative time filter is specified using the following let clause: let analytic_interval = since (<relative_time_in_seconds>)	In this example query, end time is shifted to one week later than what is specified in the **let analytic_interval=…** clause. from i in (oa_sysperf_global) **let end_time_offset=604800** **let analytic_interval = between($starttime, $endtime) let interval = $interval** where (i.host_name like “*.mydomain.com”) group by i.host_name select moving_max(i.cpu_run_queue), moving_max(i.cpu_util), moving_total(i.active_processes)

About Log File Queries Using Analytics Query Language (AQL)

Examples in this topic use Analytics Query Language (AQL) to return the information collected by log files configured using HP ArcSight Logger.

Note: These queries do not apply to **structured log files**[1].

You can use three types of AQL functions to search log file information. Click each type for more information.

Search for text strings. Use aqlrawlog.

Use the aqlrawlog function to search the log file entries stored in HP ArcSight Logger servers.

The aqlrawlog query returns the following attributes for each matching log file message entry: timestamp, message text, host name, and source host name.

Syntax: aqlrawlog(<aqllit><text_to_search></aqllit>, <starttime_as_seconds_since_epoch>, <end_time_as_seconds_since_epoch>, “”|”<comma_separated_list_of_logger_host_names>” [,<limit>])

Click here for a description of the **aqlrawlog** arguments.

[let timeout=<timeout_in_seconds>]

[let limit=<limit>]

<text_to_search> is the text string that must match in the log file entries.

Note: The <text_to_search> argument must be enclosed by the <aqllit> keyword, for example **<aqllit>severity</aqllit>**.

<starttime_as_seconds_since_epoch> is the start time of the time window within which to look for matching log file entries.

Note: To use the value selected in the Operations Analytics console, enter $starttime as the value for this argument.

<endtime_as_seconds_since_epoch> is the end time of the time window within which to look for log file entries.

Note: To use the value selected in the Operations Analytics console, enter $endtime as the value for this argument.

[1]Fragments of log file data that are stored as collections in HP Operations Analytics. Structured logs are log files that are configured as collections. These collections are created so that users can perform analytics on the log file contents. For example, you might want to query for all outliers by host name and application for a particular time range.

<comma_separated_list_of_logger_host_names> is a comma separated list of host names that identify the HP ArcSight Logger servers to query.

Tip: To query all of the HP ArcSight Logger servers configured for the current tenant, specify “” as this parameter value.

<limit> is an optional parameter that overrides the default maximum number of log file entries to return.

Note: If you do not use this parameter or the optional let limit=<limit> clause, Operations Analytics returns up to a maximum of 2000 log file messages matching the search text. You can also specify $limit for this value.

<timeout_in_seconds> is the timeout for the search operation. This parameter is specified when using the optional let timeout=… clause.

Note: If you do not specify this parameter, Operations Analytics uses the default timeout value.

Click here for examples.

/* Returns a maximum of 500 log file entries that include "error" */

aqlrawlog(<aqllit>error</aqllit>, $starttime, $endtime, “”, 500)

/*Returns the default maximum number of log file entries that include "error". This query searches log file entries only on the following servers: mylogger1.mydomain.com and mylogger2.mydomain.com logger servers*/

aqlrawlog(<aqllit>error</aqllit>, $starttime, $endtime, “mylogger1.mydomain.com,mylogger2.mydomain.com”)

/* Returns the default maximum number of log file entries that include "error" . It uses the timeout value of 5 minutes */

aqlrawlog(<aqllit>error</aqllit>, $starttime, $endtime, “”) let timeout=300

/* Returns a maximum number of 500 log file entries that include "error". It uses the timeout value of 5 minutes */

aqlrawlog(<aqllit>error</aqllit>, $starttime, $endtime, “”) let timeout=300 let limit=500

Count the number of log file entries. Use aqlrawlogcount.

Use the aqlrawlogcount function to count the log file entries stored in HP ArcSight Logger servers that contain the search text string.

Syntax: aqlrawlogcount(<aqllit><text_to_search></aqllit>, <starttime_as_seconds_since_epoch>, <end_time_as_seconds_since_epoch>, “” | ”<comma_separated_list_of_logger_host_names>”,”” | ”<comma_separated_list_of_group_by_fields>” [,<granularity_in_seconds>])

Click here for a description of each of the **aqlrawlogcount** arguments.

[let timeout=<timeout_in_seconds>]

[let limit=<limit>]

<text_to_search> is the text string that must match in each log file entry returned.

Note: The <text_to_search> argument must be enclosed by the <aqllit> keyword, for example **<aqllit>severity</aqllit>**.

<starttime_as_seconds_since_epoch> is the start time of the time window within which to look for matching log file entries.

Note: To use the value selected in the Operations Analytics console, enter $starttime as the value for this argument.

<endtime_as_seconds_since_epoch> is the end time of the time window within which to look for matching log file entries.

Note: To use the value selected in the Operations Analytics console, enter $endtime as the value for this argument.

<comma_separated_list_of_logger_host_names> is a comma separated list of host names that identify the HP ArcSight Logger servers to query.

Tip: To query all of the HP ArcSight Logger servers configured for the current tenant, specify “” as this parameter value.

<limit> is an optional parameter that overrides the default maximum number of log file entries to return.

Note: If you do not use this parameter or the optional let limit=<limit> clause, Operations Analytics returns up to a maximum of 2000 log file messages matching the search text. You can also specify $limit as the value.

<timeout_in_seconds> is the timeout for the search operation specified using the optional let timeout=… clause.

Note: If you do not specify this parameter, Operations Analytics uses the default timeout value.

The aqlrawlog query returns the following attributes for each matching log file entry: timestamp, message text, host name, and source host name.

<comma_separated_list_of_group_by_fields> is a comma separated list of the HP ArcSight Logger attributes in which to group the results.

Tip: If you do not want Operations Analytics to group the results, specify “” as the parameter value.

Note: If you specify “” as this parameter and do not specify <granularity_in_seconds>, Operations Analytics computes the moving counts without any group by criteria.

The window of time between <starttime_as_seconds_since_epoch> and <endtime_as_seconds_since_epoch> is divided into multiple intervals. Operations Analytics calculates counts at each of these intervals. Operations Analytics automatically computes the optimal length of time for each interval.

<time_interval_in_seconds> specifies the value Operations Analytics should use to subdivide the window of time between <starttime_as_seconds_since_epoch> and <endtime_as_seconds_since_epoch>. Operations Analytics computes the moving counts at each of these intervals.

Note: To use the value selected in the Operations Analytics console, enter $interval as the value for this argument. See "Dashboards and Query Panes" on page 13 for more information about how to specify the $interval parameter value in the Operations Analytics console.

<limit> is an optional parameter that overrides the default maximum number of log file entries to return.

Note: If you do not use this parameter or the optional let limit=<limit> clause, Operations Analytics returns up to a maximum of 2000 log file messages matching the search text. You can also specify $limit as the value.

<timeout_in_seconds> is the timeout for the search operation specified using the optional let timeout=… clause.

Note: If you do not specify this parameter, Operations Analytics uses the default timeout value.

Click here for examples.

/* Returns the time series counts of log file entries that contain "error" at 5 minute intervals*/

aqlrawlogcount(<aqllit>error</aqllit>, $starttime, $endtime, “”, “”, 300)

/*Returns the time series counts of log file entries that contain "error" for each combination of deviceHostName and agentSeverity at 5 minute intervals. The function queries only the mylogger1.mydomain.com server*/

aqlrawlogcount(<aqllit>error</aqllit>, $starttime, $endtime, “mylogger1.mydomain.com”, ”deviceHostName,agentSeverity”, 300)

/*Returns overall aggregate counts of log file entries that contain "error" for each combination of deviceHostName and agentSeverity. This AQL function queries only the mylogger1.mydomain.com server*/

aqlrawlogcount(<aqllit>error</aqllit>, $starttime, $endtime, “mylogger1.mydomain.com”, ”deviceHostName,agentSeverity”)

/*Returns the time series of counts of log file entries that contain "error" for each combination of deviceHostName and agentSeverity at 5 minute intervals. This AQL function queries only mylogger1.mydomain.com, uses the timeout value of 10 minutes, and queries a maximum of 1000 entries */

aqlrawlogcount(<aqllit>error</aqllit>, $starttime, $endtime, “mylogger1.mydomain.com”, ”deviceHostName,agentSeverity”, 300) let timeout=600 let limit=1000

Enter a query supported by HP ArcSight Logger. Use aqlrawlogarbitrary.

Note: A supported query is any query that is configured for use on an HP ArcSight Logger server.

Use aqlrawlogarbitrary function to run any other query supported by your HP ArcSight Logger server.

Operations Analytics displays aqlrawlogarbitrary results table format.

Syntax: aqlrawlogarbitrary(<aqllit><query_string></aqllit>, <starttime_as_seconds_since_epoch>, <end_time_as_seconds_since_epoch>, “”|”<comma_separated_list_of_logger_host_names>” [,<limit>])

Click here for a description of each of the **aqlrawlogarbitrary** function arguments.

[let timeout=<timeout_in_seconds>]

[let limit=<limit>]

<query_string> is the query string that is supported by your HP ArcSight Logger server.

Note: The <query_string> argument must be enclosed by the <aqllit> keyword, for example **<aqllit>severity</aqllit>**.

<starttime_as_seconds_since_epoch> is the start time of the time window within which to look for matching log file entries.

Note: To use the value selected in the Operations Analytics console, enter $starttime as the value for this argument.

<endtime_as_seconds_since_epoch> is the end time of the time window within which to look for matching log file entriess.

Note: To use the value selected in the Operations Analytics console, enter $endtime as the value for this argument.

<comma_separated_list_of_logger_host_names> is a comma separated list of host names of the HP ArcSight Logger servers to query.

Tip: To query all of the HP ArcSight Logger servers configured for the current tenant, specify “” as this parameter value.

<limit> is optional parameter and if specified it overrides the default maximum rows of information returned by logger to consider for returning back to Operations Analytics console.

Note: If you do not use this parameter or the optional let limit=<limit> clause, Operations Analytics returns up to a maximum of 2000 log file messages matching the search text. You can also specify $limit for this value.

<timeout_in_seconds> is the timeout for the search operation specified using the optional let timeout=… clause.

Note: If you do not specify this parameter, Operations Analytics uses the default timeout value.

Click here for examples.

/* Returns a maximum of 500 log file entries that contain the text string "error" */

aqlrawlogarbitrary(<aqllit>error</aqllit>, $starttime, $endtime, “”, 500)

/*Returns up to the default maximum number of log file entries that contain the text string "error". This AQL function queries only the mylogger1.mydomain.com and mylogger2.mydomain.com logger servers*/

aqlrawlogarbitrary(<aqllit>error</aqllit>, $starttime, $endtime, “mylogger1.mydomain.com,mylogger2.mydomain.com”)

/* Returns the default maximum number of log file entries that contain "error". This AQL function uses a timeout value of 5 minutes */

aqlrawlogarbitrary(<aqllit>error</aqllit>, $starttime, $endtime, “”) let timeout=300

/* Returns a maximum of 500 log file entries that contain "error". This AQL function uses a timeout value of 5 minutes */

aqlrawlogarbitrary(<aqllit>error</aqllit>, $starttime, $endtime, “”) let timeout=300 let limit=500

You can add a **let** clause to your aqlrawlog, aqlrawlogcount or aqlrawlogarbitrary query to define a variable that contains a list of entities returned from an AQL function. The variable can then be used in the <aqllit><*text_to_search*></aqllit> string. This feature is useful when you want to search for a set of entities, such as hosts, applications, Business Service Management transactions, or database instances without needing to enter the entire list of values. Click here for more information:

- To add a let clause to your aqlrawlog, aqlrawlogcount or aqlrawlogarbitrary query, use the following syntax:

let <*variable_name*>=<*AQL_function*>

For example, you could define the variable $myhosts to contain the list of servers returned from the AQL function named oaSysperfHosts. The oaSysperfHosts AQL function uses the following arguments to return hosts that have performance metrics collected:

oaSysperfHosts (hostFilter, numHostsLimit)

To define a variable to store the results returned from the oaSysperfHosts AQL function, use the following syntax:

let <*variable_name*>=oaSysperfHosts (hostFilter, numHostsLimit)

For example, to pass the first 50 hosts that have performance metrics collected in the enterprise.com domain to the myhosts variable, add the following let clause to your aqlrawlog, aqlrawlogcount, or aqlrawlogarbitrary query:

let myhosts=oaSysperfHosts ("*enterprise.com", 50)

- The variable you define using the let clause can be used in a text search or with a Common Event Field (CEF) field that was configured using the Operations Analytics Log File Connector for ArcSight Logger. See "Configuring the Operations Analytics Log File Connector for ArcSight Logger " in the HP Operations Analytics Configuration Guide for more information.
- To use the variable in a text search, use the following syntax:

  <aqllit><$variable></aqllit>

  For example:

  <aqllit><$myhosts></aqllit>

- To use the variable with a CEF, use the following syntax in place of <aqllit><$variable></aqllit>:

  <aqllit><CEF> in [$<variable_name>]</aqllit>

  For example:

  sourcehostName in [$myhosts]

  The previous example searches for all log file messages that contain any of the host names stored in the $myhosts variable. These host names would be the first 50 hosts that have performance metrics collected in the enterprise.com domain.

About Analytic Functions

Operations Analytics provides a set of analytic functions to analyze the metrics, topology, inventory, event, and log file data that is collected.

These analytic functions are divided into the types described in Types of Analytic Functions Provided by Operations Analytics .

Types of Analytic Functions Provided by Operations Analytics

Analytic Function Types	Description	Valid Visualizations
Overall Aggregate (Summary)	Computes a single aggregate value for a set of raw data over a specified window of time.	Table, pie chart or bar chart
Moving Aggregate (Time Series)	Computes one aggregate value at each time interval within the selected window of time. *Pie charts only*. Each moving aggregate value displayed represents a re-computed value using each data points per interval within the specified time segment. For example, the moving_avg analytic function calculates the average of all average values returned for the specified time frame and metric or attribute. Operations Analytics displays each of these re-calculated values, one per pie chart segment.	Line chart, bar chart, heat map, and pie chart
Analytic Functions applied to Overall Aggregate and Moving Aggregate Analytic Functions	Use analytic functions in AQL queries to apply analysis to the following sets of data: • Overall aggregate values for **metrics**[1] or **attributes**[2] • Moving aggregate values for metrics or attributes **Note:** These analytic functions can also be applied to **raw metrics**[3].	**Tip:** When using the topN or bottomN analytic function, Operations Analytics displays a table by default. Bar charts and pie charts can also be used to visualize topN and bottomN results.

See "Analytic Functions Provided by Operations Analytics" on the next page for more information.

Also see "Working with Query Panes" on page 154 for more information about interpreting dashboard results.

[1]Typically a measurement stored in a collection. For example, CPU utilization.
[2]A descriptor stored in a collection for an entity, such as host_name.
[3]Metrics to which an overall aggregate or moving aggregate analytic function is applied.

Analytic Functions Provided by Operations Analytics

Operations Analytics provides a set of analytic functions described in the Types of Analytic Functions Provided by HP Operations Analytics table.

Tip: These functions are used in an AQL query or AQL function. See "About the Analytics Query Language (AQL) Syntax and Structure" on page 72 for more information about the required AQL syntax. See "Examples of Using Analytic Functions in AQL Queries" on page 121 for examples of using these functions in AQL queries.

Types of Analytic Functions Provided by Operations Analytics

Analytic Function Type	Description	Valid Visualizations
Overall Aggregate (Summary)	Computes a single aggregate value for a set of raw data over a specified window of time.	Table, bar chart, and pie chart
Moving Aggregates (Time Series)	Computes one aggregate value at each time interval within the selected window of time.	Line charts, heat maps, bar charts, and pie charts

Types of Analytic Functions Provided by Operations Analytics, continued

Analytic Function Type	Description	Valid Visualizations
Analytic Functions applied to Overall Aggregate and Moving Aggregate Functions	Use analytic functions in AQL queries to apply analysis to the following sets of data: • Overall aggregate values for **metrics**[1] or **attributes**[2] • Moving aggregate values for metrics or attributes **Note:** These analytic functions can also be applied to **raw metrics**[3].	**Tip:** When using the topN or bottomN analytic function, Operations Analytics displays a table by default. Bar charts and pie charts can also be used to visualize topN and bottomN results.

[1]Typically a measurement stored in a collection. For example, CPU utilization.
[2]A descriptor stored in a collection for an entity, such as host_name.
[3]Metrics to which an overall aggregate or moving aggregate analytic function is applied.

Click here for a description of the overall aggregate (summary) analytic functions provided.

Overall Aggregate (Summary) Functions Provided by HP Operations Analytics

Function	Syntax	Description
aggregate_avg	aggregate_avg(i.<metric_unique_id>|i) <metric_unique_id> is the unique meta data identifier for the chosen metric Use aggregate_avg(i.<metric_unique_id>) to select a specific metric. **Note:** If this analytic function is applied to an attribute, Operations Analytics ignores the analytic function. Use aggregate_avg(i) to select all metrics in the collections identified in **from i in …** clause. When using the (i) notation, note the following: • Operations Analytics returns the aggregate_avg of only metrics. Attributes, such as host_name, are ignored. • If no matching metrics are found, Operations Analytics ignores the analytic function.	Identifies the average value for the metric or metrics selected.

Overall Aggregate (Summary) Functions Provided by HP Operations Analytics, continued

Function	Syntax	Description
aggregate_min	aggregate_min(i.<metric_unique_id>|i) <metric_unique_id> is the unique meta data identifier for the chosen metric Use aggregate_min(i.<metric_unique_id>) to select a specific metric use syntax. **Note:** If this analytic function is applied to an attribute, Operations Analytics ignores the analytic function. Use aggregate_min(i) to select all metrics in the collections identified in **from i in …** clause. When using the (i) notation, note the following: • Operations Analytics returns the aggregate_min of only metrics. Attributes, such as host_name, are ignored. • If no matching metrics are found, Operations Analytics ignores the analytic function.	Identifies the minimum value for the metric or metrics selected.

Overall Aggregate (Summary) Functions Provided by HP Operations Analytics, continued

Function	Syntax	Description
aggregate_max	aggregate_max(i.<metric_unique_id>|i) <metric_unique_id> is the unique meta data identifier for the chosen metric Use aggregate_max(i.<metric_unique_id>) to select a specific metric **Note:** If this analytic function is applied to an attribute, Operations Analytics ignores the analytic function. Use aggregate_max(i) to select all metrics in the collections identified in **from i in …** clause. When using the (i) notation, note the following: • Operations Analytics returns the aggregate_max of only metrics. Attributes, such as host_name, are ignored. • If no matching metrics are found, Operations Analytics ignores the analytic function.	Identifies the maximum value for the metric or metrics selected.

Overall Aggregate (Summary) Functions Provided by HP Operations Analytics, continued

Function	Syntax	Description
aggregate_total	aggregate_total(i.<metric_unique_id>|i) <metric_unique_id> is the unique meta data identifier for the chosen metric Use aggregate_total(i.<metric_unique_id>) to select a specific metric **Note:** If this analytic function is applied to an attribute, Operations Analytics ignores the analytic function. Use aggregate_total(i) to select all metrics in the collections identified in **from i in …** clause. When using the (i) notation, note the following: • Operations Analytics returns the aggregate_total of only metrics. Attributes, such as host_name, are ignored. • If no matching metrics are found, Operations Analytics ignores the analytic function.	Identifies the total value or cumulative sum for the metric or metrics selected.
aggregate_count	aggregate_count(i.<attribute_unique_id>|i) <attribute_unique_id> is the unique meta data identifier for the chosen attribute **Note:** If this analytic function is applied to a metric, Operations Analytics ignores the analytic function. Use aggregate_count(i.<attribute_unique_id>) to count only the rows with values of a specified attribute. Use aggregate_count(i) for counting all rows in a collection.	Computes the total count of rows with values of an attribute or total count of all rows in a collection table.

Overall Aggregate (Summary) Functions Provided by HP Operations Analytics, continued

Function	Syntax	Description
aggregate_distinct_count	aggregate_distinct_count(i.<attribute_unique_id>) **Note:** The (i) notation is not permitted. <attribute_unique_id> is the unique meta data identifier of the attribute for which a count of distinct values is to be calculated. **Note:** If this analytic function is applied to a metric, Operations Analytics ignores the analytic function.	Computes the total count of distinct values of an attribute.

Example AQL Queries that use Overall Aggregate Functions

Topic	Description
"Return Summary Information on Metrics (Example AQL Queries)" on page 121	Use these examples to view summary information for metrics.
"Return Summary Information on Events (Example AQL Queries) " on page 123	Use these examples to view summary information for event collections.
"Return Inventory Information Using Overall Aggregate Analytic Functions (Example AQL Queries)" on page 125	Use these examples to view a summary of inventory information for a collection.

Click here for a description of the moving aggregate (time series) functions provided.

Moving Aggregate (Time Series) Functions Provided by HP Operations Analytics

Function	Syntax	Description
moving_avg	moving_avg(i.<metric_unique_id>|i) <metric_unique_id> is the unique meta data identifier for the chosen metric Use moving_avg(i.<metric_unique_id>) to select a specific metric. **Note:** If this analytic function is applied to an attribute, Operations Analytics ignores the analytic function. Use moving_avg(i) to select all metrics in the collections identified in **from i in …** clause. When using the (i) notation, note the following: • Operations Analytics returns the moving_avg of only metrics. Attributes, such as host_name, are ignored. • If no matching metrics are found, Operations Analytics ignores the function.	Computes the average values at each time interval within the specified time window for one or more metrics. **Note:** The time interval is specified using the **let interval= …** clause.

Moving Aggregate (Time Series) Functions Provided by HP Operations Analytics, continued

Function	Syntax	Description
moving_min	moving_min(i.<metric_unique_id>|i) <metric_unique_id> is the unique meta data identifier for the chosen metric Use moving_min(i.<metric_unique_id>) to select a specific metric. **Note:** If this analytic function is applied to an attribute, Operations Analytics ignores the analytic function. Use moving_min(i) to select all metrics in the collections identified in **from i in …** clause. When using the (i) notation, note the following: • Operations Analytics returns the moving_min of only metrics. Attributes, such as host_name, are ignored. • If no matching metrics are found, Operations Analytics ignores the function.	Computes the minimum values at each time interval within the specified time window for one or more metrics. **Note:** The time interval is specified using the **let interval= …** clause.

Moving Aggregate (Time Series) Functions Provided by HP Operations Analytics, continued

Function	Syntax	Description
moving_max	moving_max(i.<metric_unique_id>|i) <metric_unique_id> is the unique meta data identifier for the chosen metric. Use moving_max(i.<metric_unique_id>) to select a specific metric. **Note:** If this analytic function is applied to an attribute, Operations Analytics ignores the analytic function. Use moving_max(i) to select all metrics in the collections identified in **from i in …** clause. When using the (i) notation, note the following: • Operations Analytics returns the moving_max of only metrics. Attributes, such as host_name, are ignored. • If no matching metrics are found, Operations Analytics ignores the function.	Computes the maximum values at each time interval within the specified time window for one or more metrics. **Note:** The time interval is specified using the **let interval= …** clause.

Moving Aggregate (Time Series) Functions Provided by HP Operations Analytics, continued

Function	Syntax	Description
moving_total	moving_total(i.<metric_unique_id>|i) <metric_unique_id> is the unique meta data identifier for the chosen metric. Use moving_total(i.<metric_unique_id>) to select a specific metric. **Note:** If this analytic function is applied to an attribute, Operations Analytics ignores the analytic function. Use moving_total(i) to select all metrics in the collections identified in **from i in …** clause. When using the (i) notation, note the following: • Operations Analytics returns the moving_total of only metrics. Attributes, such as host_name, are ignored. • If no matching metrics are found, Operations Analytics ignores the function.	Computes the totals at each time interval within the specified time window for one or more metrics. **Note:** The time interval is specified using the **let interval= …** clause.

Moving Aggregate (Time Series) Functions Provided by HP Operations Analytics, continued

Function	Syntax	Description
moving_ count	moving_count(i.<attribute_unique_id>|i) <attribute_unique_id> is the unique meta data identifier for the chosen attribute Use moving_count(i.<attribute_ unique_id>) to count only the rows with values of the specified attribute. **Note:** If this analytic function is applied to a metric, Operations Analytics ignores the analytic function. Use moving_count(i) for counting all rows in a collection.	Computes the total counts of rows with values of an attribute or total count of all rows within a collection table at each time interval within the specified time window. **Note:** The time interval is specified using the **let interval= ...** clause.
moving_ distinct_ count	moving_distinct_count(i.<attribute_ unique_id>) **Note:** The (i) notation is not permitted. <attribute_unique_id> is the unique meta data identifier of the attribute for which a count of distinct values is to be calculated. **Note:** If this analytic function is applied to a metric, Operations Analytics ignores the analytic function.	Computes the total counts of distinct values of an attribute at each time interval within the specified time window. **Note:** The time interval is specified using the **let interval= ...** clause.

See the topics described in Example AQL Queries that use Moving Aggregate Functions for example AQL queries that use moving aggregate (time series or trend) functions.

Example AQL Queries that use Moving Aggregate Functions

Topic	Description
"Return Time Series (Trend) Information on Metrics (Example AQL Queries)" on page 127	Use these examples to view time series (trend) analysis for metrics.

Example AQL Queries that use Moving Aggregate Functions, continued

Topic	Description
"Return Time Series (Trend) Information on Events (Example AQL Queries) " on page 129	Use these examples to view time series (trend) analysis for event collections.
"Return Inventory Information Using Moving Aggregate (Time Series) Analytic Functions (Example AQL Queries)" on page 130	Use these examples to view time series (trend) analysis of inventory information for a collection.
Return Time Series (Trend) Information on Structured Log Data (Example AQL Queries)	Use these examples to view a time series (trend) analysis on structured log data.

Click here for a description of the analytic functions provided.

Analytic Functions provided by Operations Analytics

Function	Syntax	Description
bottomN	bottomN(<inner_analytic_expression>|i.<metric_unique_id> [,<N>]) <inner_analytic_expression>is the use of an overall aggregate or moving aggregate function executed on an attribute or metric in a collection. <metric_unique_id> is the unique meta data identifier of a metric in the queried collection. <N> is an optional parameter that determines the number of values returned. **Note:** Use the $N variable in a bottomN analytic function when you want Operations Analytics to use the $N value specified when you add or edit a dashboard query pane. For example: bottomN(aggregate_disntinct_count (i.ciid),$N). See "Dashboards and Query Panes" on page 13 for more information about providing parameter values when creating a dashboard.	Uses the rank (ascending order) analytic function to identify the lowest N values. Operations Analytics returns the bottom N values with their associated rank. Note the following: • If you do not specify an N value in the AQL query, Operations Analytics displays the bottom five values. • The bottomN analytic function is not permitted in the **where** clause.

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
inverse_ pctile	inverse_pctile(<inner_analytic_ expression>|i.<metric_unique_id>, <pctile>) <inner_analytic_expression> is the use of an overall aggregate or moving aggregate function executed on an attribute or metric in a collection. <metric_unique_id> is unique meta data identifier of a metric property in a queried collection. <pctile> is the nth percentile parameter. Valid values are 1 to 100. **Note:** Use the $pctile variable when you want Operations Analytics to use the $pctile value specified when you add or edit a dashboard query pane. For example: inverse_pctile (moving_max(i.cpu_util),$pctile). See "Dashboards and Query Panes" on page 13 for more information about providing parameter values when creating a dashboard.	Calculates the inverse percentile distribution values for the set of raw metric, overall aggregate or moving aggregate values. Each inverse percentile distribution value is the nth percentile value among the set of values. For example, if you specify 50 as the <pctile> value, inverse_ pctile finds the 50th percentile value (or median value) for the data set of raw metric, overall aggregate or moving aggregate values. **Note:** While identifying the nth percentile value,Operations Analytics uses an ascending ranking order. Use this analytic function to filter the underlying data set of values above or below a specified percentile value. Click here for more information. When applied to raw metrics, the nth percentile value is determined for the entire set of raw metric values without any grouping. When applied to overall aggregate values, the nth percentile value is calculated for the entire set of overall aggregate values. Because the overall aggregates data is already

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
		computed in groups using the group by clause, Operations Analytics does not apply any additional grouping of this data set before identifying the single nth percentile value for the entire data set of overall aggregates. When applied to moving aggregate values, the inverse_pctile value is calculated once for each time series of moving aggregate values.

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
pctile	pctile(<inner_analytic_expression>|i.<metric_ unique_id>)) <inner_analytic_expression> is the use of an overall aggregate or moving aggregate function executed on an attribute or metric in a collection. <metric_unique_id> is unique meta data identifier of a metric property in a queried collection.	Calculates the percentile rank value expressed as a fraction for the data set of raw metric, overall aggregate or moving aggregate values. Use this analytic function on a data set of raw metrics, overall aggregate values or moving aggregate values to view the relative distribution score for each value along with the value itself. **Note:** The ranking order is ascending while calculating percentile or relative distribution values. Click here for more information. When applied to raw metrics, the percentile ranking is calculated for each of the raw metric values within the data set. Each metric value receives a percentile rank. When applied to overall aggregate values,Operations Analytics calculates the percentile ranking for each of the overall aggregate values in the data set. Each overall aggregate value receives a percentile rank. Because the overall aggregates data is already computed in groups using the group by clause, Operations Analytics does not apply any additional grouping of this data set before identifying the

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
		percentile ranks of values within the data set of overall aggregates. When applied to moving aggregate values, each moving aggregate value is assigned a percentile rank score relative to other intervals in the same time series. If the query includes multiple moving aggregate time series (trends), Operations Analytics calculates a percentile for each moving aggregate time series.

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
outlier	outlier(<inner_analytic_expression>|i.<metric_unique_id> [, <upper_pctile_limit>, <lower_pctile_limit>]) <inner_analytic_expression> is the use of an overall aggregate or moving aggregate function executed on an attribute or metric in a collection. <metric_unique_id> is unique meta data identifier of a metric property in a queried collection. <upper_pctile_limit> and <lower_pctile_limit> indicate the upper and lower boundaries of the normal range. Valid values are 1 to 100. Note the following: • The <lower_pctile_limit> must be lower than <upper_pctile_limit>. • Use the $upper_pctile_limit and $lower_pctile_limit variables in an outlier function when you want Operations Analytics to use the $upper_pctile_limit and $lower_pctile_limit values specified when you add or edit a dashboard query pane. For example: outlier(moving_count(i), $upper_pctile_limit, $lower_pctile_limit)). See "Dashboards and Query Panes" on page 13 for more information about providing parameter values when creating a dashboard.	Highlights the data that is outside of the normal range identified using the <upper_pctile_limit> and <lower_pctile_limit>. **Note:** The outlier analytic function is not permitted in the **where** clause. The outlier analytic function uses the inverse_pctile analytic to filter and identify the values above the high percentile value and below the low percentile value. By default, if you do not specify the upper and lower pctile limits, Operations Analytics uses the 1st and 95th percentile range. Any value outside of this range is an outlier. Click here for more information. When the outlier analytic function is applied to a set of moving aggregate values representing multiple time-series (trends), Operations Analytics returns only the outlier time intervals within each time series. The returned set of time series (trend) data is sorted in descending order based on the number of outlier time intervals for each time series. Click here for more information. When the outlier analytic is

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
		applied to raw metric values returned as a time series, the outlier analytic function is applied on each raw metric time series without any grouping. When the outlier analytic is applied to overall aggregate values, only the overall aggregate values that are considered to be outliers are returned. Because the overall aggregates data is already computed in groups using the group by clause, Operations Analytics does not apply any additional grouping of this data set before identifying the outlier values within the data set of overall aggregates.

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
rank	rank(<inner_analytic_expression>|i.<metric_ unique_id>)[,<order>]) <inner_analytic_expression> is the use of an overall aggregate or moving aggregate function executed on an attribute or metric in a collection. <metric_unique_id> is unique meta data identifier of a metric property in a queried collection. <order> is optional. Valid values are **"asc"** or **"desc"** .	Use this analytic on a data set of raw metrics values, overall aggregate values or moving aggregate values to view the relative rank expressed as an integer for each value along with the value itself. Use the <order> parameter to specify either ascending or descending ranking. **Note:** If you do not specify an <order>, the default order is descending. Click here for more information. When the rank analytic is applied to raw metrics, Operations Analytics assigns a rank to each metric value. When the rank analytic is applied to overall aggregate values, Operations Analytics assigns a rank to each overall aggregate value. When the rank analytic is applied to moving aggregate values that return multiple time series (trend) results, Operations Analytics assigns a rank to each interval within each time series. Each interval receives a rank score relative to the other intervals in the same time series.

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
topN	topN(<inner_analytic_expression>|i.<metric_unique_id> [,<N>]) <inner_analytic_expression>is the use of an overall aggregate or moving aggregate function executed on an attribute or metric in a collection. <metric_unique_id> is the unique meta data identifier of a metric in the queried collection. <N> is optional parameter that determines the number of values returned. **Note:** Use the $N variable in a topN analytic function when you want Operations Analytics to use the $N value specified when you add or edit a dashboard query pane. For example: topN(aggregate_distinct_count(i.ciid), $N). See "Dashboards and Query Panes" on page 13 for more information about providing parameter values when creating a dashboard.	Uses the rank (descending order) analytic function to identify the highest N values. Operations Analytics returns the top N values with their associated rank. Note the following: • If you do not specify an N value in the AQL query, Operations Analytics displays the top five values. • The topN analytic function is not permitted in the **where** clause.

See the topics described in Example AQL Queries that Apply Analytic Functions for example AQL queries that use these analytic functions.

Example AQL Queries that Apply Analytic Functions

Topic	Description
"Return Inverse Percentile Values (Example AQL Queries)" on page 140	Use these examples to view inverse percentile values.
"Return Percentile Values (Example AQL Queries)" on page 141	Use these examples to view percentile values.
"Return Outlier Values (Example AQL Queries)" on page 142	Use these examples to view outlier values.
"Return Values by Rank (Example AQL Queries)" on page 144	Use these examples to view values by rank.
"Return the Top N Values (Example AQL Queries)" on page 145	Use these examples to view top N values. **Tip:** Also use these examples to assist you in constructing AQL queries that use the bottomN analytic function.

Examples of Using Analytic Functions in AQL Queries

This topic points to examples of using the analytic functions provided by Operations Analytics in Analytic Query Language (AQL) queries. These examples describes some of the more common Analytics Query Language (AQL) queries that you might find useful. Use these examples as a guideline for creating other AQL queries you might want to use.

"Return Summary Information on Metrics (Example AQL Queries)" below

"Return Summary Information on Events (Example AQL Queries) " on page 123

"Return Inventory Information Using Overall Aggregate Analytic Functions (Example AQL Queries) " on page 125

"Return Time Series (Trend) Information on Metrics (Example AQL Queries)" on page 127

"Return Time Series (Trend) Information on Events (Example AQL Queries) " on page 129

"Return Inventory Information Using Moving Aggregate (Time Series) Analytic Functions (Example AQL Queries)" on page 130

"Apply Additional Analytic Functions to Overall Aggregates, Moving Aggregates and Raw Metrics" on page 132

Return Summary Information on Metrics (Example AQL Queries)

Operations Analytics provides a set of analytic functions that return overall aggregate (summary) information.

Click here to view the description for each overall aggregate analytic function provided.

Overall Aggregate (Summary) Analytic Functions Provided by HP Operations Analytics

Function	Syntax	Description
aggregate_avg	aggregate_avg (i.<metric_unique_id>|i) <metric_unique_id> is the unique meta data identifier for the chosen metric	Identifies the average value for the metric or metrics selected.
aggregate_min	aggregate_min (i.<metric_unique_id>|i)	Identifies the minimum value for the metric or metrics selected.

Overall Aggregate (Summary) Analytic Functions Provided by HP Operations Analytics, continued

Function	Syntax	Description
aggregate_max	aggregate_max (i.<metric_unique_id>|i)	Identifies the maximum value for the metric or metrics selected.
aggregate_total		Identifies the total value for the metric or metrics selected.
aggregate_count	aggregate_countl (i.<attribute_unique_ id>|i) <attribut_unique_id> is the unique meta data identifier for the chosen attribute	Identifies the total count of rows with values of an attribute or total count of all rows in a collection table.
aggregate_distinct_count	aggregate_distinct_ countl(i.<attribute_ unique_id>) <attribute_unique_id> is the unique meta data identifier of the attribute for which a count of distinct values is to be calculated	Identifies the total count of distinct values of an attribute.

Click each query of interest from the following list to view the example AQL query.

Note: Each of the examples queries data from the oa_sysperf_global collection. This collection uses HP Performance Agent to collect system metrics. Each example queries data for only the hosts in the **mydomain.com** domain.

Return the average CPU utilization and CPU run queue size.

The following AQL query returns the average CPU utilization and CPU run queue size for each host matching the filter criteria.

from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime)

where (i.host_name like “*.mydomain.com”) group by i.host_name

select aggregate_avg(i.cpu_util), aggregate_avg(i.cpu_run_queue)

Return the average for each of the metrics collected by the oa_sysperf_global collection.

The following AQL query returns the average for each of the metrics collected by oa_sysperf_global for each host matching the filter criteria:

from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime)

where (i.host_name like “*.mydomain.com”) group by i.host_name

select aggregate_avg(i)

Return the maximum, minimum, and average values for CPU utilization and CPU run queue size.

The following AQL query returns the maximum, minimum, average for CPU utilization and CPU run queue size for each host matching the filter criteria:

from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime)

where (i.host_name like “*.mydomain.com”) group by i.host_name

select aggregate_min(i.cpu_util), aggregate_max(i.cpu_util), aggregate_max(i.cpu_util), aggregate_min(i.cpu_run_queue), aggregate_max(i.cpu_run_queue), aggregate_avg(i.cpu_run_queue)

Return the minimum, maximum, and average for each of the metrics collected by the oa_sysperf_global collection.

The following AQL query returns the minimum, maximum and average for each of the metrics collected by oa_sysperf_global for each host matching the filter criteria:

from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime)

where (i.host_name like “*.mydomain.com”) group by i.host_name

select aggregate_min(i), aggregate_max(i), aggregate_avg(i)

Return Summary Information on Events (Example AQL Queries)

Operations Analytics provides a set of analytic functions that return overall aggregate (summary) information.

Click here to view the description for each overall aggregate analytic function provided.

Overall Aggregate (Summary) Analytic Functions Provided by HP Operations Analytics

Function	Syntax	Description
aggregate_avg	aggregate_avg (i.<metric_unique_id>|i) <metric_unique_id> is the unique meta data identifier for the chosen metric	Identifies the average value for the metric or metrics selected.
aggregate_min	aggregate_min (i.<metric_unique_id>|i)	Identifies the minimum value for the metric or metrics selected.
aggregate_max	aggregate_max (i.<metric_unique_id>|i)	Identifies the maximum value for the metric or metrics selected.
aggregate_total		Identifies the total value for the metric or metrics selected.
aggregate_count	aggregate_countl (i.<attribute_unique_ id>|i) <attribut_unique_id> is the unique meta data identifier for the chosen attribute	Identifies the total count of rows with values of an attribute or total count of all rows in a collection table.
aggregate_distinct_count	aggregate_distinct_ countl(i.<attribute_ unique_id>) <attribute_unique_id> is the unique meta data identifier of the attribute for which a count of distinct values is to be calculated	Identifies the total count of distinct values of an attribute.

Note: Each of the examples queries data from the omi_events_omievents collection. This collection uses HP Operations Manager i (OMi) to collect OMi events. Each example queries data for only the hosts in the **mydomain.com** domain.

Return the total count of OMi events for a specified host and severity combination.

The following AQL query calculates the total count of OMi events for each host and severity combinations matching the filter criteria:

from i in (omi_events_omievents)

let analytic_interval= between($starttime, $endtime)

where ((i.hostinfo_dnsname like “*mydomain.com”) && ((i.severity ilike ”CRITI*”) || (i.severity ilike “WARN*”)))

group by i.hostinfo_dnsname, i.severity select aggregate_count(i)

Return the total count of OMi events for a specified host and severity combination and for which the event count exceeds 100

The following AQL query does the same as the previous AQL query, but returns the counts for only those host name and severity combinations for which the event count exceeds 100:

from i in (omi_events_omievents)

let analytic_interval= between($starttime, $endtime)

where ((i.hostinfo_dnsname like “*mydomain.com”) && ((i.severity ilike ”CRITI*”) || (i.severity ilike “WARN*”)) && (aggregate_count(i) > 100))

group by i.hostinfo_dnsname, i.severity select aggregate_count(i)

Return Inventory Information Using Overall Aggregate Analytic Functions (Example AQL Queries)

Operations Analytics provides a set of analytic functions that return overall aggregate (summary) information.

Click here to view the description for each overall aggregate analytic function provided.

Overall Aggregate (Summary) Analytic Functions Provided by HP Operations Analytics

Function	Syntax	Description
aggregate_avg	aggregate_avg (i.<metric_unique_id>|i) <metric_unique_id> is the unique meta data identifier for the chosen metric	Identifies the average value for the metric or metrics selected.
aggregate_min	aggregate_min (i.<metric_unique_id>|i)	Identifies the minimum value for the metric or metrics selected.

Overall Aggregate (Summary) Analytic Functions Provided by HP Operations Analytics, continued

Function	Syntax	Description
aggregate_max	aggregate_max (i.<metric_unique_id>|i)	Identifies the maximum value for the metric or metrics selected.
aggregate_total		Identifies the total value for the metric or metrics selected.
aggregate_count	aggregate_countl (i.<attribute_unique_ id>|i) <attribut_unique_id> is the unique meta data identifier for the chosen attribute	Identifies the total count of rows with values of an attribute or total count of all rows in a collection table.
aggregate_distinct_count	aggregate_distinct_ countl(i.<attribute_ unique_id>) <attribute_unique_id> is the unique meta data identifier of the attribute for which a count of distinct values is to be calculated	Identifies the total count of distinct values of an attribute.

Return the number of distinct applications monitored by HP Business Process Monitor (BPM) per location.

Note: The following AQL query uses the bpm_application_performance collection. This collection uses HP Business Process Monitor (BPM) to gather application performance information.

The following AQL query calculates the number of distinct applications monitored by BPM on a location by location basis.

from i in (bpm_application_performance)

let analytic_interval = between($starttime, $endtime)

group by i.location

select aggregate_distinct_count(i.application)

Return the total count of distinct database instances reporting oracle metrics.

> **Note:** The following AQL query uses the oa_oraperf_graph collection. The oa_oraperf_graph collection uses HP Operations Smart Plug-in for Oracle to gather Oracle performance information.

The following AQL query returns a distinct counts of database instances reporting oracle metrics:

from i in (oa_oraperf_graph)

let analytic_interval= between($starttime,$endtime)

where (i.host_name like “*mydomain.com”)

group by i.host_name select aggregate_distinct_count(i.db_instance_name)

Return Time Series (Trend) Information on Metrics (Example AQL Queries)

Operations Analytics provides a set of analytic functions that return moving aggregate (time series) information.

Click here to view the description for each moving aggregate analytic function provided.

Moving Aggregate (Time Series) Analytic Functions Provided by HP Operations Analytics

Function	Syntax	Description
moving_ avg	moving_avg (i.<metric_ unique_id>|i)	Computes the average values at the specified time interval and within the specified time window for one or more metrics.
moving_ min	moving_min (i.<metric_ unique_id>|i)	Computes the minimum values at the specified time interval and within the specified time window for one or more metrics.
moving_ max	moving_max (i.<metric_ unique_id>|i)	Computes the maximum values at the specified time interval and within the specified time window for one or more metrics.
moving_ total	moving_total (i.<metric_ unique_id>|i)	Computes the totals at the specified time interval and within the specified time window for one or more metrics.
moving_ count	moving_countl (i.<attribute_ unique_id>|i)	Computes the total number of rows with values of an attribute or total count of all rows within a collection table at the specified time interval and within the specified time window.
moving_ distinct_ count	moving_ distinct_count (i.<attribute_ unique_id>)	Computes the total count of distinct values of an attribute at the specified time interval and within the specified time window.

Click each query of interest from the following list to view the example AQL query.

Note: Each of the examples queries data from the oa_sysperf_global collection. This collection uses HP Performance Agent to collect system metrics. Each example queries data for only the hosts in the **mydomain.com** domain.

Return the moving average CPU utilization and CPU run queue size.

The following AQL query returns the moving average CPU utilization and CPU run queue size for each host matching the filter criteria.

from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime) let interval=$interval

where (i.host_name like “*.mydomain.com”) group by i.host_name

select moving_avg(i.cpu_util), moving_avg(i.cpu_run_queue)

Return the moving average for each of the metrics collected by the oa_sysperf_global collection.

The following AQL query returns the moving average for each of the metrics collected by oa_ sysperf_global for each host matching the filter criteria:

from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime) let interval=$interval

where (i.host_name like “*.mydomain.com”) group by i.host_name

select moving_avg(i)

Return the moving maximum, minimum, and average values for CPU utilization and CPU run queue size.

The following AQL query returns the moving maximum, minimum, average for CPU utilization and CPU run queue size for each host matching the filter criteria:

from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime) let interval=$interval

where (i.host_name like “*.mydomain.com”) group by i.host_name

select moving_min(i.cpu_util), moving_max(i.cpu_util), moving_max(i.cpu_util), moving_min (i.cpu_run_queue), moving_max(i.cpu_run_queue), moving_avg(i.cpu_run_queue)

Return the moving minimum, maximum, and average for each of the metrics collected by the oa_sysperf_global collection.

The following AQL query returns the moving minimum, maximum and average for each of the metrics collected by oa_sysperf_global for each host matching the filter criteria:

from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime) let interval=$interval

where (i.host_name like “*.mydomain.com”) group by i.host_name

select moving_min(i), moving_max(i), moving_avg(i)

Return Time Series (Trend) Information on Events (Example AQL Queries)

Operations Analytics provides a set of analytic functions that return moving aggregate (time series or trends) information.

Click here to view the description for each moving aggregate analytic function provided.

Moving Aggregate (Time Series) Analytic Functions Provided by HP Operations Analytics

Function	Syntax	Description
moving_ avg	moving_avg (i.<metric_ unique_id>|i)	Computes the average values at the specified time interval and within the specified time window for one or more metrics.
moving_ min	moving_min (i.<metric_ unique_id>|i)	Computes the minimum values at the specified time interval and within the specified time window for one or more metrics.
moving_ max	moving_max (i.<metric_ unique_id>|i)	Computes the maximum values at the specified time interval and within the specified time window for one or more metrics.
moving_ total	moving_total (i.<metric_ unique_id>|i)	Computes the totals at the specified time interval and within the specified time window for one or more metrics.
moving_ count	moving_countl (i.<attribute_ unique_id>|i)	Computes the total number of rows with values of an attribute or total count of all rows within a collection table at the specified time interval and within the specified time window.
moving_ distinct_ count	moving_ distinct_count (i.<attribute_ unique_id>)	Computes the total count of distinct values of an attribute at the specified time interval and within the specified time window.

Note: Each of the examples queries data from the omi_events_omievents collection. This collection uses HP Operations Manager i (OMi) to collect OMi events. Each example queries data for only the hosts in the **mydomain.com** domain.

Return the moving total count of OMi events for a specified host and severity combination.

The following AQL query calculates the moving total count of OMi events for each host and severity combinations matching the filter criteria:

from i in (omi_events_omievents)

let analytic_interval=between($starttime,$endtime) let interval=$interval

where ((i.hostinfo_dnsname like “*mydomain.com”) && ((i.severity ilike ”CRITI*”) || (i.severity ilike “WARN*”)))

group by i.hostinfo_dnsname, i.severity select moving_count(i)

Return the moving total count of OMi events for a specified host and severity combination and for which the event count exceeds 100

The following AQL query does the same as the previous AQL query, but returns the moving counts for only those host name and severity combinations at only those intervals at which the event count exceeds 100:

from i in (omi_events_omievents)

let analytic_interval=between($starttime,$endtime) let interval=$interval

where ((i.hostinfo_dnsname like “*mydomain.com”) && ((i.severity ilike ”CRITI*”) || (i.severity ilike “WARN*”)) && (moving_count(i) > 100))

group by i.hostinfo_dnsname, i.severity select moving_count(i)

Return Inventory Information Using Moving Aggregate (Time Series) Analytic Functions (Example AQL Queries)

Operations Analytics provides a set of analytic functions that return moving aggregate (time series or trend) information.

Click here to view the description for each moving aggregate analytic function provided.

Moving Aggregate (Time Series) Analytic Functions Provided by HP Operations Analytics

Function	Syntax	Description
moving_ avg	moving_avg (i.<metric_ unique_id>|i)	Computes the average values at the specified time interval and within the specified time window for one or more metrics.
moving_ min	moving_min (i.<metric_ unique_id>|i)	Computes the minimum values at the specified time interval and within the specified time window for one or more metrics.
moving_ max	moving_max (i.<metric_ unique_id>|i)	Computes the maximum values at the specified time interval and within the specified time window for one or more metrics.
moving_ total	moving_total (i.<metric_ unique_id>|i)	Computes the totals at the specified time interval and within the specified time window for one or more metrics.
moving_ count	moving_countl (i.<attribute_ unique_id>|i)	Computes the total number of rows with values of an attribute or total count of all rows within a collection table at the specified time interval and within the specified time window.

Moving Aggregate (Time Series) Analytic Functions Provided by HP Operations Analytics, continued

Function	Syntax	Description
moving_ distinct_ count	moving_ distinct_count (i.<attribute_ unique_id>)	Computes the total count of distinct values of an attribute at the specified time interval and within the specified time window.

Note: The following two examples query data from the collection rtsm_ci_inventory. This collection uses HP Run-Time Service Model (RTSM) to collect inventory information on topology.

Return the moving number of distinct applications monitored by HP Business Process Monitor (BPM) per location.

Note: The following AQL query uses the bpm_application_performance collection. This collection uses HP Business Process Monitor (BPM) to gather application performance information.

The following AQL query calculates the moving number of distinct applications monitored by BPM on a location by location basis.

from i in (bpm_application_performance)

let analytic_interval = between($starttime, $endtime) let interval = $interval

group by i.location

select moving_distinct_count(i.application)

Return the moving total count of distinct database instances reporting oracle metrics.

Note: The following AQL query uses the oa_oraperf_graph collection. The oa_oraperf_graph collection uses HP Operations Smart Plug-in for Oracle to gather Oracle performance information.

The following AQL query returns moving total counts of the distinct database instances reporting oracle metrics:

from i in (oa_oraperf_graph)

let analytic_interval= between($starttime,$endtime) let interval = $interval where (i.host_name like “*mydomain.com”)

group by i.host_name

select moving_distinct_count(i.db_instance_name)

Apply Additional Analytic Functions to Overall Aggregates, Moving Aggregates and Raw Metrics

Operations Analytics provides a set of analytic functions that can be applied to overall aggregate and moving aggregate analytic functions.

Click here to view the description for each of these analytic functions provided.

Analytic Functions provided by Operations Analytics

Function	Syntax	Description
bottomN	bottomN(<inner_analytic_expression>|i.<metric_unique_id> [,<N>]) <inner_analytic_expression>is the use of an overall aggregate or moving aggregate function executed on an attribute or metric in a collection. <metric_unique_id> is the unique meta data identifier of a metric in the queried collection. <N> is an optional parameter that determines the number of values returned. **Note:** Use the $N variable in a bottomN analytic function when you want Operations Analytics to use the $N value specified when you add or edit a dashboard query pane. For example: bottomN(aggregate_disntinct_count (i.ciid),$N). See "Dashboards and Query Panes" on page 13 for more information about providing parameter values when creating a dashboard.	Uses the rank (ascending order) analytic function to identify the lowest N values. Operations Analytics returns the bottom N values with their associated rank. Note the following: • If you do not specify an N value in the AQL query, Operations Analytics displays the bottom five values. • The bottomN analytic function is not permitted in the **where** clause.

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
inverse_ pctile	inverse_pctile(<inner_analytic_ expression>|i.<metric_unique_id>, <pctile>) <inner_analytic_expression> is the use of an overall aggregate or moving aggregate function executed on an attribute or metric in a collection. <metric_unique_id> is unique meta data identifier of a metric property in a queried collection. <pctile> is the nth percentile parameter. Valid values are 1 to 100. **Note:** Use the $pctile variable when you want Operations Analytics to use the $pctile value specified when you add or edit a dashboard query pane. For example: inverse_pctile (moving_max(i.cpu_util),$pctile). See "Dashboards and Query Panes" on page 13 for more information about providing parameter values when creating a dashboard.	Calculates the inverse percentile distribution values for the set of raw metric, overall aggregate or moving aggregate values. Each inverse percentile distribution value is the nth percentile value among the set of values. For example, if you specify 50 as the <pctile> value, inverse_ pctile finds the 50th percentile value (or median value) for the data set of raw metric, overall aggregate or moving aggregate values. **Note:** While identifying the nth percentile value,Operations Analytics uses an ascending ranking order. Use this analytic function to filter the underlying data set of values above or below a specified percentile value. Click here for more information. When applied to raw metrics, the nth percentile value is determined for the entire set of raw metric values without any grouping. When applied to overall aggregate values, the nth percentile value is calculated for the entire set of overall aggregate values. Because the overall aggregates data is already

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
		computed in groups using the group by clause, Operations Analytics does not apply any additional grouping of this data set before identifying the single nth percentile value for the entire data set of overall aggregates. When applied to moving aggregate values, the inverse_pctile value is calculated once for each time series of moving aggregate values.

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
pctile	pctile(<inner_analytic_expression>|i.<metric_ unique_id>)) <inner_analytic_expression> is the use of an overall aggregate or moving aggregate function executed on an attribute or metric in a collection. <metric_unique_id> is unique meta data identifier of a metric property in a queried collection.	Calculates the percentile rank value expressed as a fraction for the data set of raw metric, overall aggregate or moving aggregate values. Use this analytic function on a data set of raw metrics, overall aggregate values or moving aggregate values to view the relative distribution score for each value along with the value itself. **Note:** The ranking order is ascending while calculating percentile or relative distribution values. Click here for more information. When applied to raw metrics, the percentile ranking is calculated for each of the raw metric values within the data set. Each metric value receives a percentile rank. When applied to overall aggregate values,Operations Analytics calculates the percentile ranking for each of the overall aggregate values in the data set. Each overall aggregate value receives a percentile rank. Because the overall aggregates data is already computed in groups using the group by clause, Operations Analytics does not apply any additional grouping of this data set before identifying the

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
		percentile ranks of values within the data set of overall aggregates. When applied to moving aggregate values, each moving aggregate value is assigned a percentile rank score relative to other intervals in the same time series. If the query includes multiple moving aggregate time series (trends), Operations Analytics calculates a percentile for each moving aggregate time series.

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
outlier	outlier(<inner_analytic_expression>|i.<metric_unique_id> [, <upper_pctile_limit>, <lower_pctile_limit>]) <inner_analytic_expression> is the use of an overall aggregate or moving aggregate function executed on an attribute or metric in a collection. <metric_unique_id> is unique meta data identifier of a metric property in a queried collection. <upper_pctile_limit> and <lower_pctile_limit> indicate the upper and lower boundaries of the normal range. Valid values are 1 to 100. Note the following: • The <lower_pctile_limit> must be lower than <upper_pctile_limit>. • Use the $upper_pctile_limit and $lower_pctile_limit variables in an outlier function when you want Operations Analytics to use the $upper_pctile_limit and $lower_pctile_limit values specified when you add or edit a dashboard query pane. For example: outlier(moving_count(i), $upper_pctile_limit, $lower_pctile_limit)). See "Dashboards and Query Panes" on page 13 for more information about providing parameter values when creating a dashboard.	Highlights the data that is outside of the normal range identified using the <upper_pctile_limit> and <lower_pctile_limit>. **Note:** The outlier analytic function is not permitted in the **where** clause. The outlier analytic function uses the inverse_pctile analytic to filter and identify the values above the high percentile value and below the low percentile value. By default, if you do not specify the upper and lower pctile limits, Operations Analytics uses the 1st and 95th percentile range. Any value outside of this range is an outlier. Click here for more information. When the outlier analytic function is applied to a set of moving aggregate values representing multiple time-series (trends), Operations Analytics returns only the outlier time intervals within each time series. The returned set of time series (trend) data is sorted in descending order based on the number of outlier time intervals for each time series. Click here for more information. When the outlier analytic is

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
		applied to raw metric values returned as a time series, the outlier analytic function is applied on each raw metric time series without any grouping. When the outlier analytic is applied to overall aggregate values, only the overall aggregate values that are considered to be outliers are returned. Because the overall aggregates data is already computed in groups using the group by clause, Operations Analytics does not apply any additional grouping of this data set before identifying the outlier values within the data set of overall aggregates.

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
rank	rank(<inner_analytic_expression>|i.<metric_ unique_id>)[,<order>]) <inner_analytic_expression> is the use of an overall aggregate or moving aggregate function executed on an attribute or metric in a collection. <metric_unique_id> is unique meta data identifier of a metric property in a queried collection. <order> is optional. Valid values are **"asc"** or **"desc"** .	Use this analytic on a data set of raw metrics values, overall aggregate values or moving aggregate values to view the relative rank expressed as an integer for each value along with the value itself. Use the <order> parameter to specify either ascending or descending ranking. **Note:** If you do not specify an <order>, the default order is descending. Click here for more information. When the rank analytic is applied to raw metrics, Operations Analytics assigns a rank to each metric value. When the rank analytic is applied to overall aggregate values, Operations Analytics assigns a rank to each overall aggregate value. When the rank analytic is applied to moving aggregate values that return multiple time series (trend) results, Operations Analytics assigns a rank to each interval within each time series. Each interval receives a rank score relative to the other intervals in the same time series.

Analytic Functions provided by Operations Analytics, continued

Function	Syntax	Description
topN	topN(<inner_analytic_expression>|i.<metric_unique_id> [,<N>]) <inner_analytic_expression>is the use of an overall aggregate or moving aggregate function executed on an attribute or metric in a collection. <metric_unique_id> is the unique meta data identifier of a metric in the queried collection. <N> is optional parameter that determines the number of values returned. **Note:** Use the $N variable in a topN analytic function when you want Operations Analytics to use the $N value specified when you add or edit a dashboard query pane. For example: topN(aggregate_distinct_count(i.ciid), $N). See "Dashboards and Query Panes" on page 13 for more information about providing parameter values when creating a dashboard.	Uses the rank (descending order) analytic function to identify the highest N values. Operations Analytics returns the top N values with their associated rank. Note the following: • If you do not specify an N value in the AQL query, Operations Analytics displays the top five values. • The topN analytic function is not permitted in the **where** clause.

See the following topics for examples that use these analytic functions and the expected results.

"Return Inverse Percentile Values (Example AQL Queries)" below

"Return Percentile Values (Example AQL Queries)" on the next page

"Return Outlier Values (Example AQL Queries)" on page 142

"Return Values by Rank (Example AQL Queries)" on page 144

"Return the Top N Values (Example AQL Queries)" on page 145

Return Inverse Percentile Values (Example AQL Queries)

Operations Analytics provides a set of analytic functions that can be applied to overall aggregate (summary) or moving aggregate (time series or trend) information. See"Analytic Functions Provided by Operations Analytics" on page 100 for more information.

The following examples use the inverse_pctile analytic function to return inverse percentile values for sets of data returned by the overall aggregate and moving aggregate analytic functions.

Note: The following two examples query data from the oa_sysperf_global collection. This collection uses HP Operations Agent to collect system metrics. Each example queries data for only the hosts in the **mydomain.com** domain.

Return the hosts and their overall aggregate average values of CPU utilization that are greater than the 95th percentile.

The following AQL query determines the hosts and their overall aggregate average values of CPU utilization which are greater than 95th percentile value amongst the overall aggregate average values for all hosts matching the filter criteria:

from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime)

where((i.host_name like “*.mydomain.com”) && (aggregate_avg(i.cpu_util) > inverse_pctile (aggregate_avg(i.cpu_util), 95)))

group by i.host_name select aggregate_avg(i.cpu_util)

Return the raw CPU utilization values that are greater than the 95th percentile value.

The following AQL query returns the raw CPU utilization values that are greater than the 95th percentile value:

from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime)

where((i.host_name like “*.mydomain.com”)&& (i.cpu_util > inverse_pctile(i.cpu_util, 95)))

select i.host_name, i.timestamp, i.cpu_util

Note: The following two examples query data from the omi_events_omievents collection. This collection uses HP Operations Manager i (OMi) to collect OMi events. Each example queries data for only the hosts in the **mydomain.com** domain.

Return the hosts and their overall aggregate count of events that are greater than the 95th percentile.

The following AQL query determines the hosts and their overall aggregate count of events that are greater than 95th percentile among the overall aggregate event count values for all hosts matching the filter criteria:

from i in (omi_events_omievents) let analytic_interval= between($starttime,$endtime)

where((i.hostinfo_dnsname like “*.mydomain.com”)&& (aggregate_count(i) > inverse_pctile (aggregate_count(i), 95)))

group by i.hostinfo_dnsname select aggregate_count(i)

Return Percentile Values (Example AQL Queries)

Operations Analytics provides a set of analytic functions that can be applied to overall aggregate (summary) or moving aggregate (time series or trend) information. See "Analytic Functions Provided by Operations Analytics" on page 100 for more information.

The following examples use the pctile analytic function to return percentile values for sets of data returned by the overall aggregate and moving aggregate analytic functions.

Note: The following example queries data from the oa_sysperf_global collection. This collection uses HP Operations Agent to collect system metrics. Each example queries data for only the hosts in the **mydomain.com** domain.

Return the hosts and their overall aggregate average values of CPU utilization along with the percentile rank

The following AQL query determines the hosts and their overall aggregate average values of CPU utilization along with the percentile rank for the value among the overall aggregate average values for all hosts matching the filter criteria:

from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime)

where(i.host_name like “*.mydomain.com”)

group by i.host_name select pctile(aggregate_avg(i.cpu_util))

Note: The following example queries data from the omi_events_omievents collection. This collection uses HP Operations Manager i (OMi) to collect OMi events. Each example queries data for only the hosts in the **mydomain.com** domain.

Return the hosts and their overall aggregate count of events with the percentile rank of the overall aggregate event count.

The following AQL query determines the hosts and their overall aggregate count of events along with percentile ranks of the overall aggregate event count values for all hosts matching the filter criteria:

from i in (omi_events_omievents)

let analytic_interval= between($starttime,$endtime)

where(i.hostinfo_dnsname like “*.mydomain.com”)

group by i.hostinfo_dnsname

select pctile(aggregate_count(i))

Note: The following example queries data from the bpm_application_performance collection. This collection uses HP Business Process Monitor (BPM) to gather application performance information.

Return Outlier Values (Example AQL Queries)

Operations Analytics provides a set of analytic functions that can be applied to overall aggregate (summary) or moving aggregate (time series or trend) information. See "Analytic Functions Provided by Operations Analytics" on page 100 for more information.

The following examples use the outlier analytic function to return outlier values for sets of data returned by the overall aggregate and moving aggregate analytic functions.

Note: The following set of examples query data from the oa_sysperf_global collection. This collection uses HP Operations Agent to collect system metrics. Each example queries data for only the hosts in the **mydomain.com** domain.

Return the outlier values for the hosts that have an overall aggregate average value of CPU utilization outside the specified high and low range.

The following AQL query determines only the hosts and their overall aggregate average values of CPU utilization that are outside the specified high and low range among the overall aggregate average values for all hosts matching the filter criteria:

```
from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime)

where( i.host_name like “*.mydomain.com” )

group by i.host_name

select outlier(aggregate_avg(i.cpu_util) )
```

Return only the outlier moving average values of CPU utilization and their intervals.

The following AQL query returns per host only moving average values of CPU utilization that are outside the specified high and low range and the interval for each:

```
from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime) let interval=$interval

where( i.host_name like “*.mydomain.com” )

group by i.host_name

select outlier(moving_avg(i.cpu_util))
```

Note: The following example queries data from the omi_events_omievents collection. This collection uses HP Operations Manager i (OMi) to collect OMi events. Each example queries data for only the hosts in the **mydomain.com** domain.

Return the hosts and their overall aggregate count of events that are outside the specified high and low range.

The following AQL query determines the hosts and their overall aggregate count of events that are outside the specified high and low range among all hosts matching the filter criteria:

```
from i in (omi_events_omievents)

let analytic_interval= between($starttime,$endtime)

where( i.hostinfo_dnsname like “*.mydomain.com” )

group by i.hostinfo_dnsname

select outlier(aggregate_count(i))
```

Note: The following example queries data from the bpm_application_performance collection. This collection uses HP Business Process Monitor (BPM) to gather application performance information.

Return the locations and their overall aggregate distinct count of applications monitored by BPM for those counts that are outside the specified high and low range.

The following AQL query determines the locations of the outlier overall aggregate distinct count of applications monitored by BPM, among all locations matching the filter criteria:

```
from i in (bpm_application_performance)

let analytic_interval= between($starttime,$endtime)

where(i.location like “*” )

group by i.location

select outlier(aggregate_distinct_count(i.application))
```

Return Values by Rank (Example AQL Queries)

Operations Analytics provides a set of analytic functions that can be applied to overall aggregate (summary) or moving aggregate (time series or trend) information. See "Analytic Functions Provided by Operations Analytics" on page 100 for more information.

The following examples use the rank analytic function to return the calculated rank and associated values for sets of data returned by the overall aggregate and moving aggregate analytic functions.

Note: The following example queries data from the oa_sysperf_global collection. This collection uses HP Operations Agent to collect system metrics. Each example queries data for only the hosts in the **mydomain.com** domain.

Return the hosts that have the 10 lowest values for overall aggregate average CPU utilization along with the rank of each value.

The following AQL query determines the hosts that have the 10 lowest values for overall aggregate average CPU utilization. This AQL query also returns the rank associated with each value.

```
from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime)

where(( i.host_name like “*.mydomain.com” )&& ( rank(aggregate_avg(i.cpu_util),"asc") <= 10 ))

group by i.host_name

select aggregate_avg(i.cpu_util)
```

Note: The following two examples query data from the omi_events_omievents collection. This collection uses HP Operations Manager i (OMi) to collect OMi events. Each example queries data for only the hosts in the **mydomain.com** domain.

Return the hosts that have the highest 10 values for the overall aggregate count of events.

The following AQL query determines the hosts and their overall aggregate count of events that are the highest 10 values. This AQL query also returns the rank associated with each value.

from i in (omi_events_omievents) let analytic_interval= between($starttime,$endtime) where((i.hostinfo_dnsname like “*.mydomain.com”)&& (rank(aggregate_count(i)) <=10))

group by i.hostinfo_dnsname

select aggregate_count(i)

Return the Top N Values (Example AQL Queries)

Operations Analytics provides a set of analytic functions that can be applied to overall aggregate (summary) or moving aggregate (time series or trend) information. See "Analytic Functions Provided by Operations Analytics" on page 100 for more information.

Tip: Also use these examples to assist you in constructing AQL queries that use the bottomN analytic function.

The following examples use the topN analytic function to return the top *n* values for sets of data returned by the overall aggregate and moving aggregate analytic functions.

Note: The following examples query data from the oa_sysperf_global collection. This collection uses HP Operations Agent to collect system metrics. Each example queries data for only the hosts in the **mydomain.com** domain.

Return the top five hosts and their overall aggregate average values of CPU utilization. This query also returns the associated relative ranks.

The following AQL query determines the top five hosts and their overall aggregate average values of CPU utilization among the overall aggregate average values and relative ranks for all hosts matching the filter criteria:

from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime)

where (i.host_name like “*.mydomain.com”)

group by i.host_name

select topN(aggregate_avg(i.cpu_util))

Return the top five raw CPU utilization values.

The following AQL query returns the top five raw CPU utilization values:

from i in (oa_sysperf_global)

let analytic_interval= between($starttime,$endtime)

where (i.host_name like “*.mydomain.com”)

select i.host_name, i.timestamp, topN(i.cpu_util)

Note: The following example queries data from the omi_events_omievents collection. This collection uses HP Operations Manager i (OMi) to collect OMi events. Each example queries data for only the hosts in the **mydomain.com** domain.

Return the top 10 hosts with the highest overall aggregate count of events.

The following AQL query determines the top 10 hosts with the highest overall aggregate count of events among the overall aggregate event count values for all hosts matching the filter criteria:

from i in (omi_events_omievents)

let analytic_interval= between($starttime,$endtime)

where (i.hostinfo_dnsname like “*.mydomain.com”)

group by i.hostinfo_dnsname

select topN(aggregate_count(i), 10)

Note: The following example queries data from the bpm_application_performance collection. This collection uses HP Business Process Monitor (BPM) to gather application performance information.

Return the top five locations with the highest overall aggregate distinct count of applications monitored by BPM.

The following AQL query determines the top five locations with the highest overall aggregate distinct count of applications monitored by BPM among the overall aggregate distinct application counts for all locations matching the filter criteria:

from i in (bpm_application_performance)

let analytic_interval= between($starttime,$endtime)

where(i.location like “*”)

group by i.location

select topN(aggregate_distinct_count(i.application))

About Analytics Query Language (AQL) Functions

Analytic Query Language (AQL) functions are functions that can be used in place of an associated AQL query. When you define an AQL function, you name the AQL function, define its arguments and the associated AQL query.

Note: To select an AQL Function provided by Operations Analytics, use the Add a Query Pane feature from an Operations Analytics dashboard. See "Dashboards and Query Panes" on page 13 for more information.

To create an AQL function use the following syntax:

`define` <*AQL function name*>(*argument_1*, *argument_2,...argument_n*)=<*AQL query syntax*>

Note: Arguments are those values that are passed to the associated AQL function. Any value that is used in the AQL query is known as a parameter. For example, the name of a host might be a valid parameter for an AQL query.

To use an AQL function use the following syntax:

[<*AQL function name*>(*value for argument_1*, *value for argument_2,...value for argument_n*)]

Note: The brackets ([]) are mandatory.

See AQL Function Syntax for a description of the AQL function syntax.

Note the following:

- You create AQL functions using a text editor. See"Define Analytic Query Language (AQL) Functions" on page 233 for more information.
- To make the AQL functions available to your user community, import the AQL functions using the opsa-aql-import.sh. See "Import Analytic Query Language (AQL) Functions" on page 234 for more information.
- The arguments that can be passed to an AQL function include any parameter included in an AQL query. See "About Parameters Used in an Analytics Query Language (AQL) let clause" on page 84 for more information about the parameters that can be included in a let clause. See "About the Analytics Query Language (AQL) Syntax and Structure" on page 72 for more information about additional parameters that can be defined in an AQL query.

AQL Function Syntax

Syntax	Description	Example
AQ_function_name	Name of the AQL function. **Tip:** Use a name that will help you to remember the AQL function purpose. Alphanumeric characters and underscore (_) are permitted. Spaces and other special characters (~ ! @ # $ % ^ &; * ( ) + -) are not permitted.	cpu_threshold
argument_n	The nth argument to be passed to the associated AQL query. You can enter any number of arguments. **Note:** In this example, **percent** is used to identify the cpu utilization percent threshold.	percent

AQL Function Syntax, continued

Syntax	Description	Example
AQL_ query_ syntax	Syntax for the AQL query to which the AQL function is associated. **Note:** When the AQL function is used, each argument value provided is passed.to the associated AQL query. See the **bold** text in the Example.	from i in (oa_sysperf_ global) let analytic_ interval=between ($starttime,$endtime) where (aggregate_ avg(i.cpu_util > **percent**) group by i.host_name select i.host_name, aggregate_avg (i.cpu_util)

Click here for the complete example.

`define cpu_threshold(percent) =`	from i in (oa_sysperf_global) let analytic_interval=between($starttime,$endtime) where (aggregate_avg(i.cpu_util) > percent) group by i.host_name select i.host_name, aggregate_avg(i.cpu_util)

To use the cpu_threshold AQL function to return a list of all the hosts where the average CPU utilization exceeds 80 percent, include the following parameter values:

```
[cpu_threshold(0.8)]
```

Click here for another example.

The following AQL function selects the host name that matches the value of argument **name**. The query returns the following information for the most recent **number** of OMi events that originated from the host selected:

- host name (hostinfo_dsname)
- timestamp
- message title
- severity

define `host_events (name,number)` =	from i in (omi_events_omievents) where (i.hostinfo_dnsname like **name**) let analytic_interval = between($starttime,$endtime) let offset = 0 let limit = **number** select i.hostinfo_dnsname, i.timestamp, i.title, i.severity

To use the host_events AQL function to return a list of the most recent 50 events for all hosts in the "enterprise.com" domain, include the following argument values:

```
[host_events("enterprise.com", 50)]
```

Operations Analytics provides a set of AQL functions for your use. See "Analytic Query Language (AQL) Functions Provided by Operations Analytics" below for more information.

Analytic Query Language (AQL) Functions Provided by Operations Analytics

By default, Operations Analytics provides several AQL functions to assist you with creating AQL queries and associated dashboards.

To view the AQL functions provided by Operations Analytics, use either of the following methods:

- If you are an Operations Analytics administrator, look at the .aql files in $OPSA_HOME/inventory/lib/hp/aql.
- Use the Add a Query Pane feature from an Operations Analytics dashboard. See "Dashboards and Query Panes" on page 13 for more information.

Operations Analytics also provides a set of analytic functions that you might commonly use in your AQL queries when creating AQL functions. See "Analytic Functions Provided by Operations Analytics" on page 100 for more information.

When using AQL functions, note the following:

- Your Operations Analytics administrator can specify which AQL functions to make available for your use.
- If you are an Operations Analytics administrator, you can create your own AQL functions using a text editor and then import those functions in to Operations Analytics. See "Import Analytic Query Language (AQL) Functions" on page 234 for more information.

About Analytics Query Language (AQL) Expressions

Analytic Query Language (AQL) expressions include multiple AQL functions. Use AQL expressions when you want the results of multiple queries to be combined into a single query pane in a

dashboard.

You can use AQL functions in an AQL expression in any of the following ways:

Use a single AQL function.

Syntax: [<aql_function_invocation>]

See "About Analytics Query Language (AQL) Functions" on page 146 for more information.

Concatenate multiple AQL functions.

Concatenating multiple AQL functions enables you to concatenate the results from each AQL function as if they were run individually.

Syntax: [<aql_function1>,<aql_function2>, ….<aql_function*n*>]

Click here for an example:

The following AQL function returns the concatenation of the results from the following:

- moving averages of CPU utilization
- outlier values for the data set of moving averages of CPU utilization

[oaSysperfMovingMetric(“*.mydomain.com”, cpu_util, moving_avg),
oaSysperfOutlierMovingMetric(“*.mydomain.com”, cpu_util, moving_avg)]

Click here for the definition of each AQL function used in the example expression.

/* Returns the moving aggregation analytic function results for the specified metric. Input parameters are host filter, metric and analytic function. */

define oaSysperfMovingMetric(hostFilter, metric, moving_analytic) =

from i in (oa_sysperf_global)

let analytic_interval = between($starttime, $endtime) let interval = $interval

where i.host_name like hostFilter

group by i.host_name

select moving_analytic(i.metric);

/* Returns the outlier values for the results from a moving aggregate analytic function on a metric. Input parameters are host filter, metric & analytic. */

define oaSysperfOutlierMovingMetric(hostFilter, metric, moving_analytic) =

from i in (oa_sysperf_global)

let analytic_interval = between($starttime, $endtime)

let interval = $interval where i.host_name like hostFilter group by i.host_name

select outlier(moving_analytic(i.metric));

Use multiple AQL functions so that the results from one AQL function is an input filter for another AQL function.

This type of AQL expression is known as an AQL composition.

Syntax: [do <target_function> filter by <filter_function> with <filter_criteria>]

Click here for more syntax information.

<target_function> is the AQL function to execute.

<filter_function> is the AQL function used to filter the results.

<filter_criteria> is the criteria to use for filtering the results of target function. The syntax of <filter_criteria> is:

(<filter_criteria_element1>, <filter_criteria_element2>, ….)

Each <filter_criteria_element> specifies a metric or attribute column name with its associated collection. Values for the column name specified must be returned in the target_function and filter_function results.

Note: All of the filter criteria elements must be met to successfully filter the target function results.

The syntax for any filter criteria element is:

<target_function_name>.<target_function_resultcolumn> == <filter_function_name>.<filter_function_resultcolumn>

The <target_function_resultcolumn> can be any of the expected result columns from the results of <target_function>.

<target_function_name> is the name of the target function

Similarly, <filter_function_resultcolumn> can be any of the expected result columns from the results of <filter_function> . The <filter_function_name> is the name of the filter function.

Click here for an example.

The following example AQL expression returns the moving_avg, moving_max and moving min of CPU utilization for the top five hosts with the highest aggregate_avg cpu_util values.

[do oaSysperfMovingMetricAvgMaxMin(“*”, cpu_util) filter by oaSysperfTopNAggregateMetric (“*.mydomain.com”,cpu_util,aggregate_avg,5) with (oaSysperfMovingMetricAvgMaxMin.host_name== oaSysperfTopNAggregateMetric.host_name)]

Click here for the definition of each AQL function used in the example expression.

/* Returns the moving average, maximum, and minimum values of a specific metric by host. Input parameters are the host filter and the metric. */

define oaSysperfMovingMetricAvgMaxMin(hostFilter, metric) =

from i in (oa_sysperf_global)

let analytic_interval = between($starttime,$endtime) let interval = $interval

where i.host_name like hostFilter

group by i.host_name

select moving_avg(i.metric), moving_max(i.metric), moving_min(i.metric);

/* Returns the topN of a moving aggregate analytic function on a metric. Input parameters are the host filter, metric, moving aggregate analytic function, and N. */

define oaSysperfTopNMovingMetric(hostFilter, metric, moving_analytic, N) =

from i in (oa_sysperf_global)

let analytic_interval = between($starttime, $endtime) let interval = $interval

where i.host_name like hostFilter group by i.host_name

select topN(moving_analytic(i.metric), N);

Click here for another example.

The following AQL expression returns the aggregate_avg CPU utilization for all server nodes in the Operations Analytics topology. These servers include the database server nodes. This example uses topology data to filter and return metric analysis for important entities in your topology:

[do oaSysperfAggregateMetric("*",cpu_util,aggregate_avg) filter by opsaNodes()

with (

oaSysperfAggregateMetric.host_name== opsaNodes.opsa_server_name,
oaSysperfAggregateMetric.host_name== opsaNodes.collector_server_name,
oaSysperfAggregateMetric.host_name== opsaNodes.logger_server_name,
oaSysperfAggregateMetric.host_name== opsaNodes.vertica_node

)]

Click here for the definition of each AQL function used in the example expression.

/* Returns the results of the overall aggregate analytic function applied to the specified metric. Input parameters are host filter, metric and overall aggregate analytic function. */

define oaSysperfAggregateMetric(hostFilter,metric,aggregate_analytic) =

from i in (oa_sysperf_global) let analytic_interval = between($starttime, $endtime)

where i.host_name like hostFilter

group by i.host_name

select aggregate_analytic(i.metric);

/* Returns the host names of Operations Analytics application servers, logger servers, collector servers and vertica nodes in an Operations Analyticsdeployment */

define opsaNodes() = from i in (opsa_topology) select i.opsa_server_name, i.logger_server_name, i.collector_server_name, i.vertica_node;

Operations Analytics provides a set of AQL functions for your use. See "About Analytics Query Language (AQL) Functions" on page 146 and "Analytic Query Language (AQL) Functions Provided by Operations Analytics" on page 149 for more information.

Chapter 4: Working with Query Panes

This section describes the different types of charts and visualizations used to display data in Query Panes.

Learn About

Data Types

Moving Aggregate Data Visualizations

Operations Analytics presents moving aggregate (time series) data as line charts, heat maps, bar charts and pie charts. Moving aggregate (time series) data is data that is displayed according to a time interval within a specified time range.

This data might include the total, average, minimum, or maximum values calculated at each interval over the specified time range. It might also include the count of unique instances or values. For example, you might want to view CPU utilization for each unique hosts in a specified domain at 1 hour intervals for the last 24 hours.

Operations Analytics displays time series (moving aggregate) data as a line chart by default.

Moving Aggregate Data Visualizations

Operations Analytics presents moving aggregate (time series) data as line charts, heat maps, bar charts and pie charts. Moving aggregate (time series) data is data that is displayed according to a time interval within a specified time range.

This data might include the total, average, minimum, or maximum values calculated at each interval over the specified time range. It might also include the count of unique instances or values. For example, you might want to view CPU utilization for each unique hosts in a specified domain at 1 hour intervals for the last 24 hours.

Operations Analytics displays time series (moving aggregate) data as a line chart by default.

Overall Aggregate Data Visualizations

Operations Analytics presents overall aggregate data as bar charts, pie charts, or tables.

Overall aggregate data is data that is grouped by total, average, minimum, or maximum values within a specified time range.

Operations Analytics displays overall aggregate (summary of totals, counts, averages, maximum values, or minimum values) data in table format by default.

Default Visualizations

If you select a visualization that is not supported by your Analytics Query Language (AQL) search query, Operations Analytics uses the default visualizations described in the following **Default Visualizations** tables. See "Dashboards and Query Panes" on page 13 for more information about selecting a visualization in a dashboard query pane. See "About the Analytics Query Language (AQL) " on page 71 for more information about AQL search queries.

Default Visualizations by Types of Analytic Functions

AQL Query	Default Visualization	Valid Visualizations
Includes a Moving Aggregate (Time Series) Analytic Function	Line Chart	Line chart, heat map, bar chart, and pie chart
Includes an Overall Aggregate (Summary) Analytic Function	Table	Table, bar chart and pie chart

Tip: When using the topN or bottomN analytic function, Operations Analytics displays a bar chart by default. You can also use topN and bottomN analytic functions to visualize pie charts and tables.

About Bar Charts

You can use both moving aggregate (time series) and overall aggregate (summary) analytic functions to display your results as a bar chart.

Group the Results and Select the Items to Display

- You can group the items in a bar chart by entities or metrics. Entities are defined as any items that are measured by your metrics. To do so, select **Group by Entity** or **Group by Metric**.
- Select the entities or metrics to display by using the drop down menu.
- Select the group to display by using the **Go To Page** menu or the arrows at the bottom of the pane.

About Heat Maps

You can use moving aggregate (time series) analytic functions to display your results as a heat map.

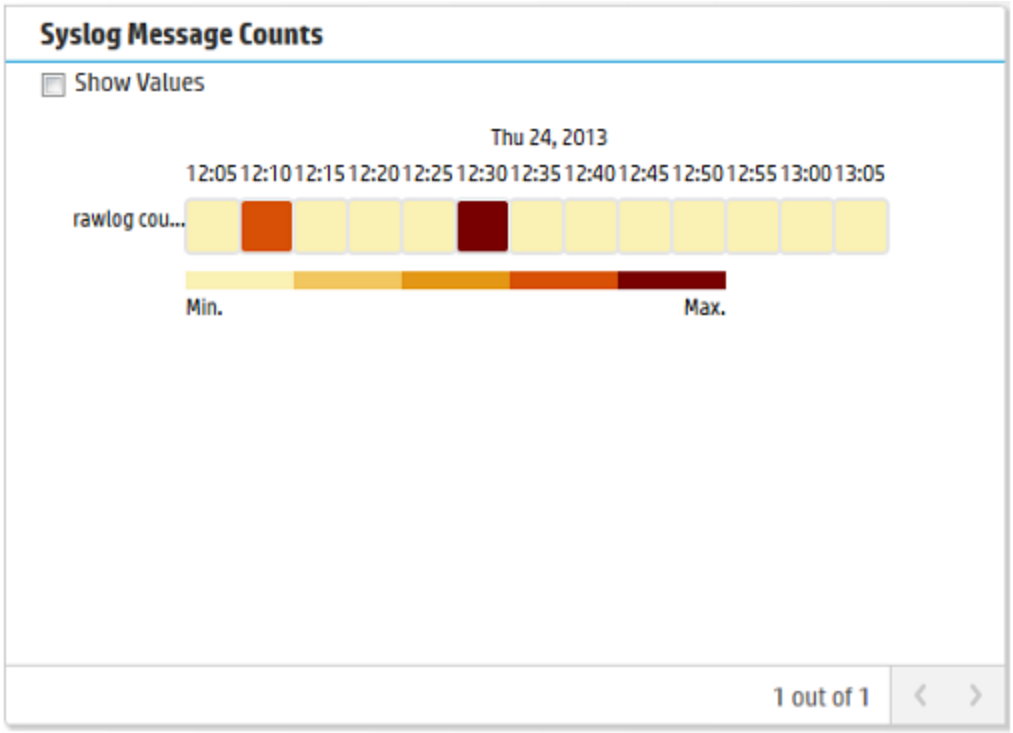
Moving aggregate (time series) analytic functions display results according to a time interval within a specified time range. This data might include actual metric values or total, average, minimum, or maximum values calculated at each interval over the specified time range. For example, you might want to view CPU utilization for each unique host in a specified domain at 1 hour intervals for the last 24 hours.

Heat maps use a series of color-coded rectangles to map returned values to a scale based on the minimum and maximum values. Each cell color is determined as follows:

- Operations Analytics identifies the minimum and maximum value per the group by entity for the selected metric. The minimum and maximum values are identified in the available results for the selected duration.
- Operations Analytics calculates the percentage of each cell value in relation to the minimum and maximum value.

- The calculated percentage value is associated with a pre-determined color shade. For example, a value of 50 percent might be associated with a medium shade of orange.

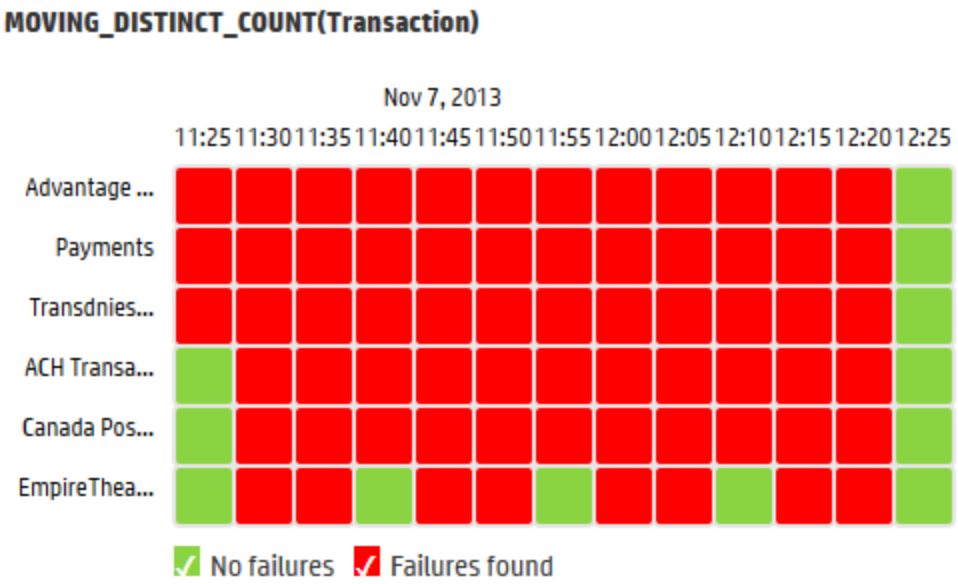
The following heat map example displays the number of syslog log file messages generated over a specified time period:



When using the heat map legend, note the following:

- The legend describes the minimum to maximum value ranges represented by each color used in the map.
- A clear rectangle indicates no data is available.
- Some dashboards provided by Operations Analytics use heat maps to display metrics that indicate some type of failure. Operations Analytics uses green to indicate **No failures** and red to indicate **Failures found**.

For example:



You can perform the following operations on heat maps:

Display the value within each heat map cell

You can display the first few characters of the value that is represented within each heat map cell by clicking ✔**Show Values**.

Calculate the percentage values using the minimum and maximum values for the entire matrix, per row, or per column

To re-calculate percentage values in a heat map:

1. Mouse over the query pane toolbar for the query pane you want to change.
2. Click to edit the query pane.
3. Navigate to the **Visualization** tab.
4. Select **Heat**.
5. Do either of the following:
   a. Select **Matrix** to calculate the heat percentages using the minimum and maximum values of the entire data set (matrix).
   b. Select **Row** to calculate the heat percentages using the minimum and maximum values per row.
   c. Select **Column** to calculate the heat percentages using the minimum and maximum values per column.
6. Click **OK**.

Operations Analytics recalculates the heat colors based on the new minimum and maximum values.

View additional heat maps in a query pane

Operations Analytics enables you to navigate through a series of heat maps by using the › and ‹ buttons.

About Line Charts

You can use moving aggregate (time series) analytic functions to display your results as a line chart.

When using line charts, note the following:

- Operations Analytics displays multiple line charts in a single query pane when the Analytic Query Language (AQL) search query requests in multiple line charts.
- Operations Analytics displays time series information in line chart format by default.
- When creating BPM line charts, if you want to see data gaps (for when an application status was unavailable), add `i.status` to the AQL query.

  Example: In the following example, add the bold text to the AQL Query.

  from i in (bpm_application_performance) let analytic_interval=between($starttime, $endtime) let interval=$interval select i.application, moving_avg(i.transaction_response_time), **i.status**

You can perform the following operations on line charts:

- To change the order that items are displayed in the list, select **Group by Entity** or **Group by Metric**.
- To display different entities or metrics, select the check boxes next to the items in the list.
- To view a prediction line for an entity or metric, click the Predict button. For details, see "Predictive Analytics" on page 165.

About Pie Charts

You can use both moving aggregate (time series) and overall aggregate (summary) analytic functions to display your results as a pie chart.

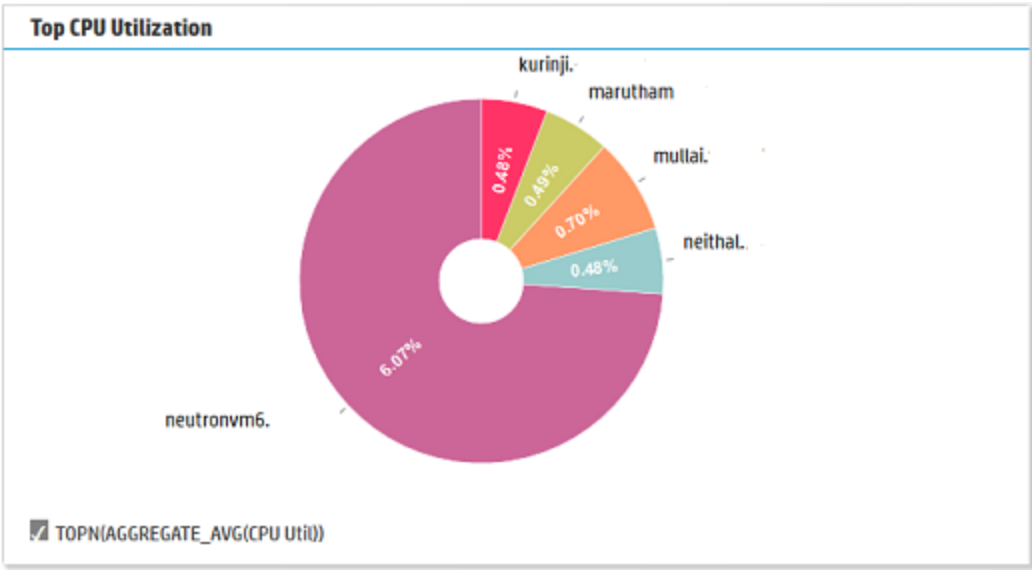
Moving aggregate (time series) analytic functions display results according to a time interval within a specified time range. This data might include the total, average, minimum, or maximum values calculated at each interval over the specified time range. For example, you might want to view CPU utilization for each unique hosts in a specified domain at 1 hour intervals for the last 24 hours.

Each moving aggregate value displayed represents a re-computed value using each data points per interval within the specified time segment. For example, the moving_avg analytic function calculates the average of all average values returned for the specified time frame and metric or

attribute. Operations Analytics displays each of these re-calculated values, one per pie chart segment.

Overall aggregate (summary) data is data that is grouped by total, average, minimum, or maximum values within a specified time range. For example, you might want to view the total number of log messages generated by each host within a specified domain within the last hour.

Operations Analytics displays the values for each pie segment as shown in the following example:



ing example: Select items in the chart to drill down. This generates a new dashboar focusing on the item you sele

About Sunburst Charts

Sunburst charts display the hierarchy you defined using the topology manager. They display services, their associated groups, their associated hosts, and the top metrics for each host.

To interpret the data in a sunburst chart, note the following:

- The root or center of a sunburst chart does not represent an object.
- Sunburst charts use color ranges to show the relative weight of a metric among the set of objects rather than to show status. Operations Analytics uses a darker color to indicate there is more of a particular value and a lighter shade of the same color to indicate there is less of a value.
- Gray indicates no values are available.
- Operations Analytics calculates the color fill for each parent node using the average color of all child nodes. When determining the average, It ignores any node with a fill color of gray.

You can perform the following operations on a sunburst chart:

- To select a metric to display, use the dropdown menu.
- To return the sunburst chart to its orignial detail, click the center of the chart.
- To drill down into any of the elements in the chart, click the element.

About Table Data

Operations Analytics presents overall aggregate data as bar charts, pie charts, or tables. Overall aggregate data is data that is grouped by total, average, minimum, or maximum values within a specified time range.

Operations Analytics displays overall aggregate (summary of totals, counts or averages) data in table format by default.

Note: Operations Analytics also displays log file information in table format by default.

When viewing table data, note the following:

- You can use an AQL query to specify the column names to be displayed. Operations Analytics displays each column name in the order in which it appears in the AQL query.
- If you do not specify column names in your query, Operations Analytics initially displays a maximum of eight columns.
- If more than eight columns are returned from the search, Operations Analytics displays the set of columns that are determined to be of the most value. Examples of these "preferred" columns include **raw**, **message**, **title**, **severity** and **host**.
- Operations Analytics does not display identification columns that are for internal use only.

You can perform the following operations on table data:

- To filter the results, enter a string in the text field.
- To restore the original column settings, select the **Columns** drop down menu and select **Restore original**.
- To sort the data, use the up and down buttons at the top of each column.
- To get more details about a row, click . To hide the details, click .
- To show or hide columns, select the **Columns** drop down menu and use the check boxes next to the column names.

Chapter 5: Filter Search Query Results

Operations Analytics enables you to filter your search query results using the following methods:

Tables only. Use the **Filter** option to filter the results by words or phrases.

The **Filter** option enables you to filter the results according to a word or phrase.

> **Note:** The word or phrase you enter must be an exact match in the results displayed.

Use the Time Line to fine tune the Time Range selected.

Operations Analytics enables you to focus on a specified time segment using the slide bar that appears above the metrics, log file and event data displayed. For example, you might want to focus on a particular day or a particular peak period.

> **Note:** The time range attribute that appears next to the search query initially defines the x-axis for the bar, line or plot diagram displayed as well as the time frame for the log file and event information that is displayed.

Changing the Time Line segment, changes the information displayed in visualizations and tables for all metric and log file and event data.

To filter your analysis by time segment:

Slide each end of the time line to the beginning and end point of the time you want to use:

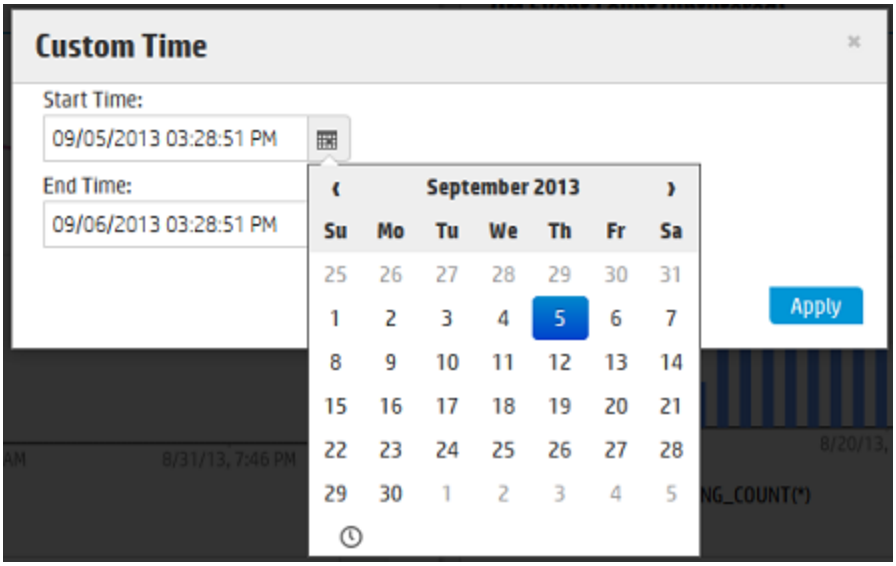
Operations Analytics filters the information available to focus only on the time segment you selected in each of the metric visualizations displayed. The log file and event information is also filtered based on the time segment you specify.

Use the Time Range option to filter the results by a specified time period.

To change the time range for the data displayed, by doing either of the following:

- Refine your search query to narrow the information presented.
- Change the time range value from the Time Range drop-down menu to narrow or broaden the time range for which the data is displayed:

Use the **Custom Time** option when you want to specify a start and end date using the Operations Analytics calendar:



See "Search Tool" on page 26 for more information.

You can also specify a filter in your Phrased Query search using collection attribute values stored in a collection. See "Filter the Scope of a Phrased Query Language (PQL) Query " on page 65 for more information.

Chapter 6: Play Back History

Operations Analytics enables you to play back your dashboard results using the ▶ Play feature.

Use this feature when you want to view the most recent changes in data over time or when you want to note the point at which a problem began to occur.

When using this feature, note the following:

- Operations Analytics uses the start and end time specified in the time line.
- Operations Analytics selects the optimum time segment within the specified start and end time in which to display the results. For example, if the time line specifies 1 day, Operations Analytics might choose a time interval of 1 hour. If the time line specifies 1 hour, Operations Analytics might choose a time interval of 5 minutes.

> **Note:** If you provide an $interval parameter value in a **query pane**[1], Operations Analytics uses the $interval value you specify for the time segment for only that query pane . See "Dashboards and Query Panes" on page 13 for more information.

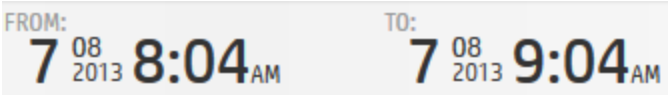
To play back your search query results:

1. Click « Playback.
2. Click ▶ (Play).
3. Do any of the following:
   - To pause the recording, click ❚❚ (Pause) or press the spacebar. To unpause press the spacebar again.
   - To fast forward to a new location, click ❚❚ (Pause), then ▶▶ (Fast Forward).
   - To rewind to a new location, click ❚❚ (Pause), then ◀◀ (Rewind).
   - To reverse play, click ◀ (Back).

> **Note:** If a query pane shows multiple pages of data, Operations Analytics replays only the results for the current query pane.

As Operations Analytics replays the results, it indicates each point in time for which data is displayed as shown in the following example:

[1]Displays the results of an Analytic Query Language (AQL) query, AQL function, or AQL expression. If you use the Phrased Query Language (PQL) in your search, HP Operations Analytics converts the PQL query to one or more AQL queries and subsequent query panes.



When you finish viewing the playback results, click (Pause).

To restore the dashboard to its original state before using the play back feature, click « Playback.

Chapter 7: Predictive Analytics

OpsA's predictive analytics enables you to generate a prediction line for one or more metrics based on past behavior and seasonal trends.

To Access

To turn on predictive analytics in a Metric Data query pane, click the **Predict** button.

Click  to specify the length of the prediction line. By default, the prediction line runs for one day.

Learn About

About Predictive Analytics

OpsA can predict the future behavior of some metrics and display this information in a query pane. The prediction line is displayed as a dashed line, with the option of adding a prediction sleeve to show the margin of error.

Typically it takes about 2-3 hours to gather enough information to enable the prediction feature. The prediction confidence indicates the strength of the prediction and can be viewed in the tooltip over the  icon. Confidence increases as more data is collected for a given metric.

The tooltip also displays the trend of the prediction over time. For example, if the prediction is that the value will decrease from the current time until the end of the prediction time, the tooltip will indicate that there is a descending trend line.

To calculate the prediction, OpsA makes use of the following items:

- Previous metric data and trends. For example, the data is steadily increasing or decreasing over time.
- Seasonal patterns (up to one month). For example, every morning at 8:30 there is a peak as employees arrive at the office.

Predictive analytics presents different displays depending on whether you are viewing one metric or more than one metric.

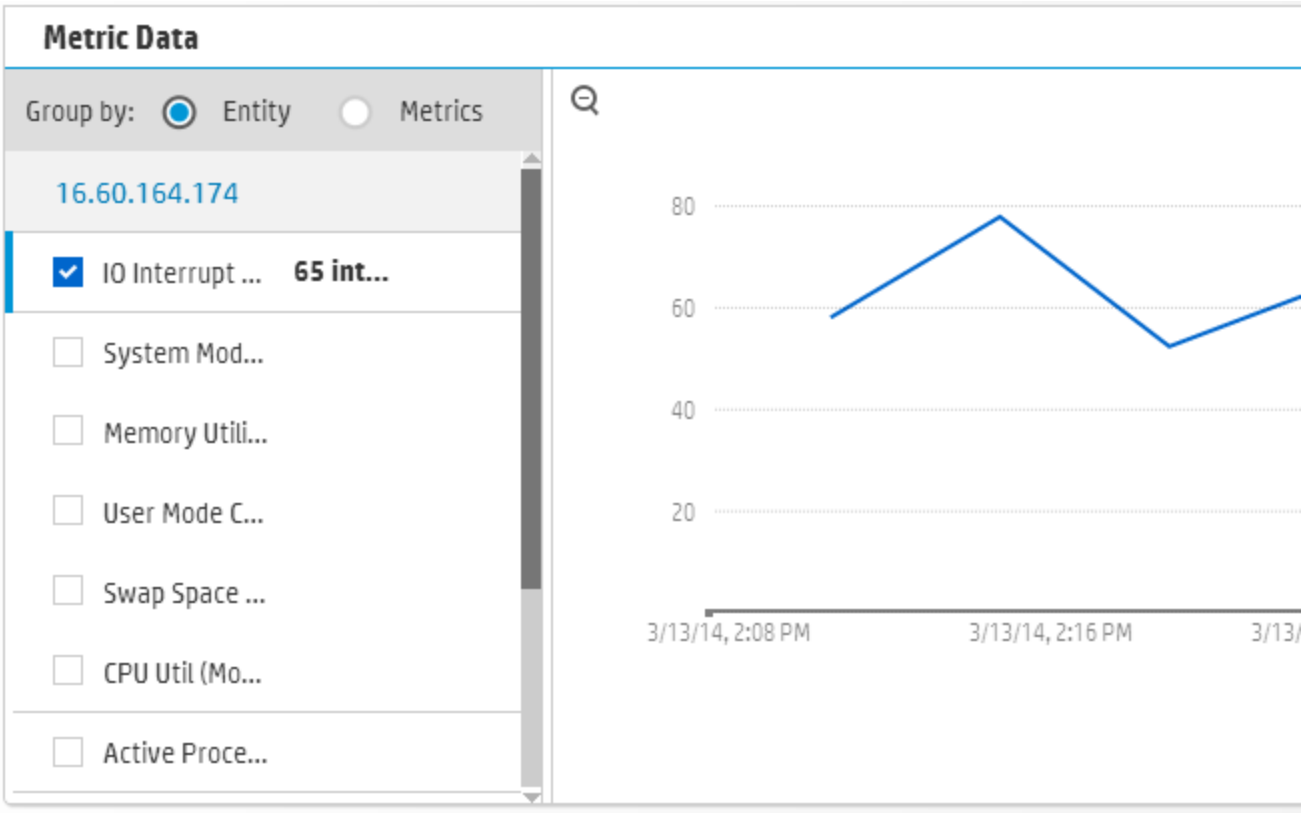
Limitations:

- While the prediction feature is generally accurate, inaccurate predictions can occur at times due to unexpected events.
- Most AQL statements are supported with the prediction feature, but a limited number are not.

Tasks

Using Predictive Analytics

1. Select the check boxes next to the items you wish to view.



2. Click the **Predict** 

button or click  to specify the time period. By default, predictive analytics is activated for one day.

3. To view the strength of the prediction, mouse over the icon. To view the prediction sleeve which displays the margin of error, click  .

4. To remove the prediction lines, click **Stop Predict** 

.

Chapter 8: Log Analytics

Log Analytics is a forensic tool that scans your log messages over a given time range and generates a list of the most significant ones.

To Access:

Search for a host, group of hosts, or service using the search tool. Locate the **Log Analytics - Top Unusual Log Messages** Query Pane.

Learn About

About Log Analytics

Searching for the root cause of a problem can be daunting. Even using OpsA's PQL searching technology, knowing where to start can be difficult. OpsA has designed a powerful Log Analytics algorithm that creates a list of the top suspected log messages. This algorithm runs over a user-defined time range for a host or a user defined group of hosts (a service). The Log Analytics algorithm uses a number of different parameters to calculate message significance, such as:

- Distance from problem time (user defined)
- Severity
- Specific keywords (for example: Exception)
- Repetition and seasonality (to identify insignificant messages)
- User feedback

The results can be viewed as a graph or in a list format.

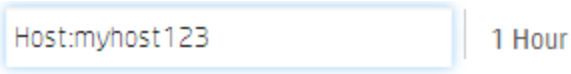
About Message Groups

Operations Analytics automatically analyzes your messages and creates message groups. Message groups are comprised of messages with very similar texts. These groups can later be liked, ignored, and analyzed as one unit. For details, see the tasks below.

Tasks

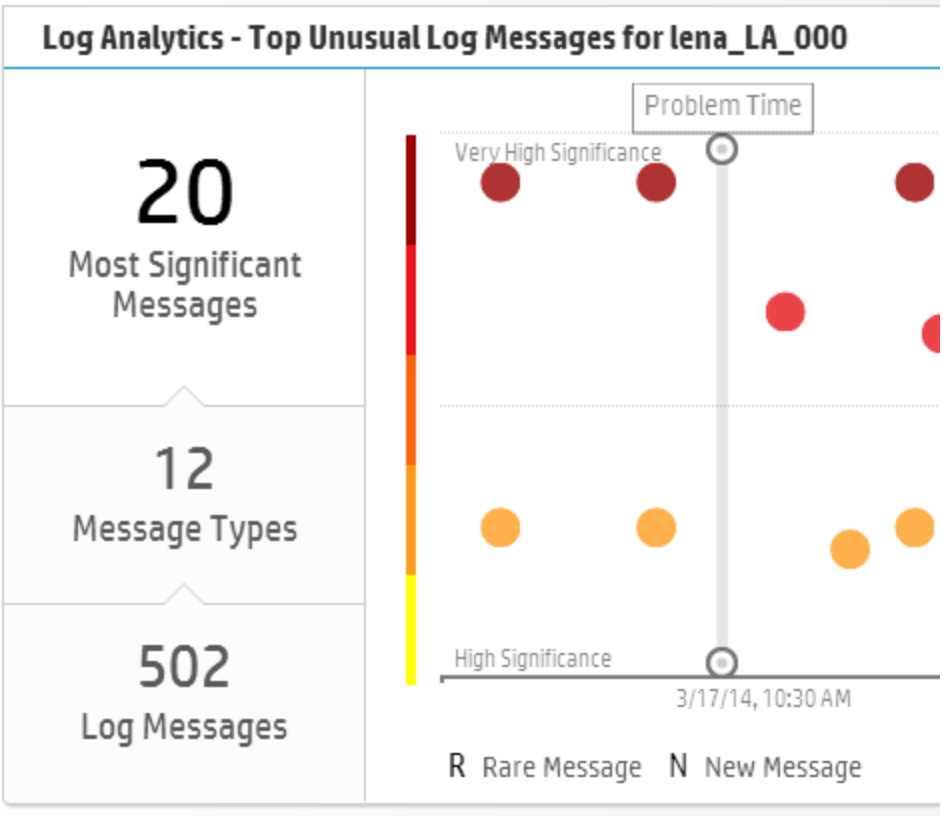
Log Analytics Workflow

1. Prerequisite: Make sure that you have peformed the steps for configuring log analytics in the Operations Analytics Quick Start Guide. The log analytics feature will not work without performing those steps.
2. Search for a host, groups of hosts, or a service using the search tool.



Alternatively , you can add the Top Unusual Log Messages query pane to a custom dashboard. For details, see below.

3. Locate the **Top Unusual Log Messages** query pane and define the time the problem started in the query pane by sliding the **Problem Time** indicator to the appropriate time. OpsA then recalculates the most significant messages based on the problem time you select.



4. Hover over the bubbles in the graph to view the tooltips. At this point in the procedure, all additional steps are optional and you can stop as soon as you have located the root cause of your problem.

5. Click a circle or the area labelled **X Most Significant Messages** on the left to open the log viewer.

> **Note:** To open the log viewer in the general log messages tab, select the area labelled **X Log Messages.**

6. Use the filtering capabilities of the log viewer to locate the root of your problem.

a. Use the fields at the top of each column to filter the results. For example, if you type "error" in field at the top of the Message Text column, the results will be limited to items that have the string "error" in the log message text. Alternatively, you can double-click a word in the message text column to filter the results by that word.

> **Note:** You can use a variety of custom expressions in the Message Text field. For details, see below.

b. Select **Show liked only** to display message groups that you have previously liked using the  button.

c. You can manually ignore individual message groups by using the ignore  button. You can later restore these items by using the **Ignored Messages** button.

7. You can view the distribution of all messages that are similar to a specific message by viewing the graphs on the bottom right of the log viewer. When you select a message, the distribution of messages with the same group ID is displayed. When you select underlined text in the message text field, the distribution of messages with different values for the underlined text is displayed.

> **Example:** The message text is "Processing error on server 1234" You can click the string "1234" to view the distribution of server names for all messages that have the same text and the same group ID.

8. To view all messages, including non-significant messages, select the **Log messages** tab.

Modifying the Significant Message Calculation Model

Log analytics uses a number of different criteria to calculate which messages are significant. You can affect this calculation in the following ways:

- **Problem time**

  In the **Log Analytics - Top Unusual Log Messages** Query Pane, move the problem time indicator to the location that you believe the problem occurred. The significance of messages is calculated based on proximity to this time.

- **Keywords**

  OpsA uses certain keywords such as **Exception** to determine significance. You can add and remove additional keywords and set their importance.

  a. Click the Settings  button and select **Log Analytics Settings**.

  b. Enter a display name and your keyword in the **Expressions** field.

Note: You can use a variety of custom expressions in this field. For details, see below.

c. Indicate the relative importance of this expression in the **Importance** drop down menu.

d. Click **Add**.

- **Likes**

  In the log viewer, click the like  button to indicate that this message group is significant to you. This information is used in future calculations to determine message significance.

- **Ignore**

  In the log viewer, click the ignore  button to ignore a message group. This removes the message group from the log viewer list and the Top Unusual Messages chart. You can later restore these items by using the **Ignored Messages** button.

How to Add a Log Analytics Query Pane to a Custom Dashboard

Add a query pane with the following AQL query to a custom dashboard:

aqllogsummary(<aqllit>*</aqllit>, $starttime, $endtime, $problemtime)

For details about creating custom query panes, see "How to Add or Edit a Query Pane" on page 20.

How to Search for Strings in Log Analytics

You can improve your expressions when searching for strings by using the tips in this section. They are applicable in both the the Text field of the Log Viewer and the *Expressions field of the Log Analytics Settings user interface.

Expression	Meaning
and	Search for both strings before and after the expression. **For example:** *one and two* means search for the strings "one" and "two".
a space between strings	Spaces are interpretted as and expressions **For example:** *one two* means search for the strings "one" and "two".
or	Search for either the string before or after the or expression. **For example:** *one or two* means search for "one" or "two".

"expression"	Search for the exact expression (whole word only). **For example:** *one and two* means search for the exact expression "one and two". **Tip:** To search for a string that contains any of the other expressions described in this table such as ( ),? and, or, * put them in quotation marks " ". **Limitation:** You cannot search for a string in which the string itself contains quotation marks " ".
( )	Groups expressions **For example:** *(one or two) and three* means search for the string "three" and either "one" or "two". Parentheses can be used multiple times and can be stacked in expressions like *four and (three or (one and two))*.
*	The wildcard expression is intrepreted as any number of characters (including 0). **For example:** *User*23* means search for the string "UserX23" where X is any string including empty. **Exception:** If you use * at the end of a word and no other wildcard expressions are used, the results will only return strings that occur at the beginning of a message.
?	The limited wildcard expression is interpreted as one unknown character. **For example:** *User?23* means search for the string "UserX23" where X is any one character.

Chapter 9: Topology Manager

The Topology Manager enables you define a logical hierarchy for monitored hosts. You can group hosts together based on their function, their location, or any other grouping that is meaningful to you when organizing your services.

Learn About

Services, Groups, and Hosts

Hosts are organized into **groups** and **services**. A **service** is a collection of **groups**, and a **group** is a collection of **hosts**.

For example, you might create a service that includes web servers, applications servers, and database servers. In order to easily reference all these hosts and get a holistic view of the service, you would create groups for web servers and so on. The groups will correspond to the groups you want to look at in Operations Analytics. A subsequent search for this service will return results for all the underlying hosts, providing a single pane of glass for all hosts that make up the service.

Tasks

How to define a service:

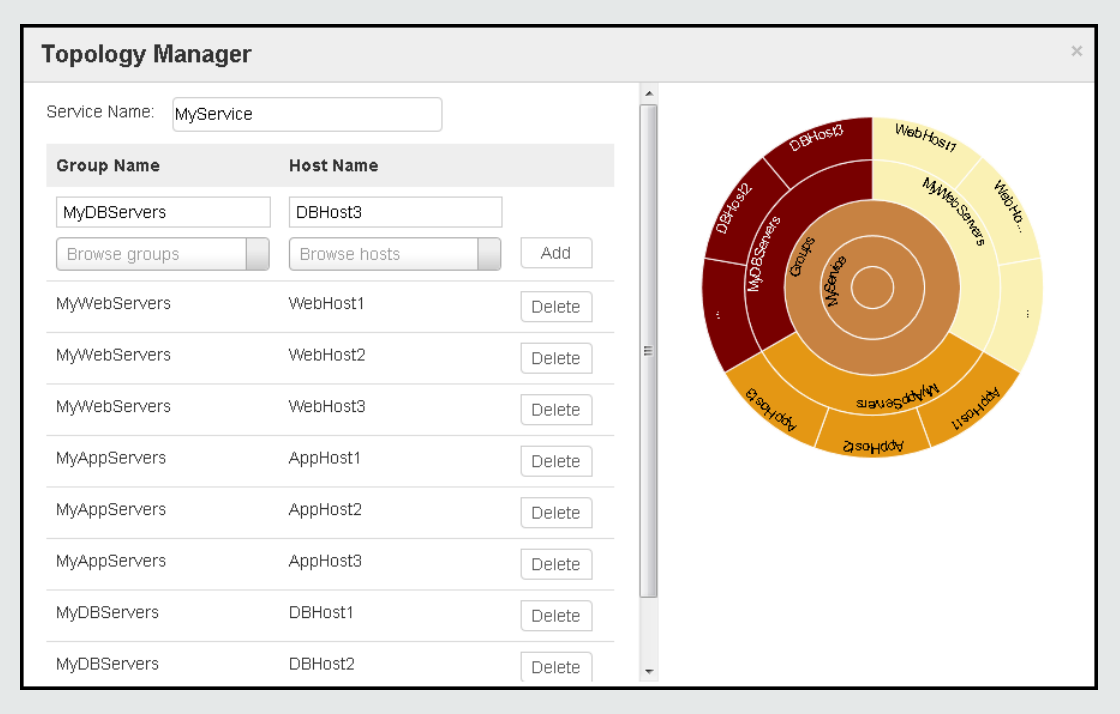
1. Click **Settings** and select **Topology Manager**.
2. Select **New**, and enter a name for your service.
3. Enter a group name and a host, then click **Add**.

   Tip: You can define a dynamic set of hosts by using the * symbol. For example, if you enter **dbhost*** as your host name, OpsA will add all hosts that begin with the string **dbhost** to the specified group. The group definition will be updated automatically if additional hosts are defined with the string **dbhost**.

   You can select the host from a list; as you type the first letters of the host, the list filters automatically. When adding a host, you can add it to an existing group or to a new one.
4. Continue defining groups and their hosts until you are done, and then click **Save**.

As a simple example, you can define a service called MyService, as follows:

- This service is made up of the groups **MyWebServers**, **MyAppServers**, and **MyDBServers**.
- These groups are made up of **WebHost1-3**, **AppHost1-3**, and **DBHost1-3** respectively.



After you define a service, you can then search for it and view metrics, events and logs that are relevant to all the hosts in that serv

Chapter 10: Collections

This topic describes the terms and procedures related to data collection sources.

Learn About

About Keys and Link Tags

Keys identify a column in a collection that you want Operations Analytics to use to do either of the following:

- Narrow a search within a single collection
- Match metrics for one entity (collection row) to the same or related entity (collection row) across collections

Typically, key columns uniquely identify an entity instance.

When using a key column to narrow a search within only one collection, Operations Analytics returns only those metrics for the specified key column value. For example, if the **host_name** column is defined as a key in a cpu metrics collection, the host_name key column enables you to search for cpu metrics for a specific host name.

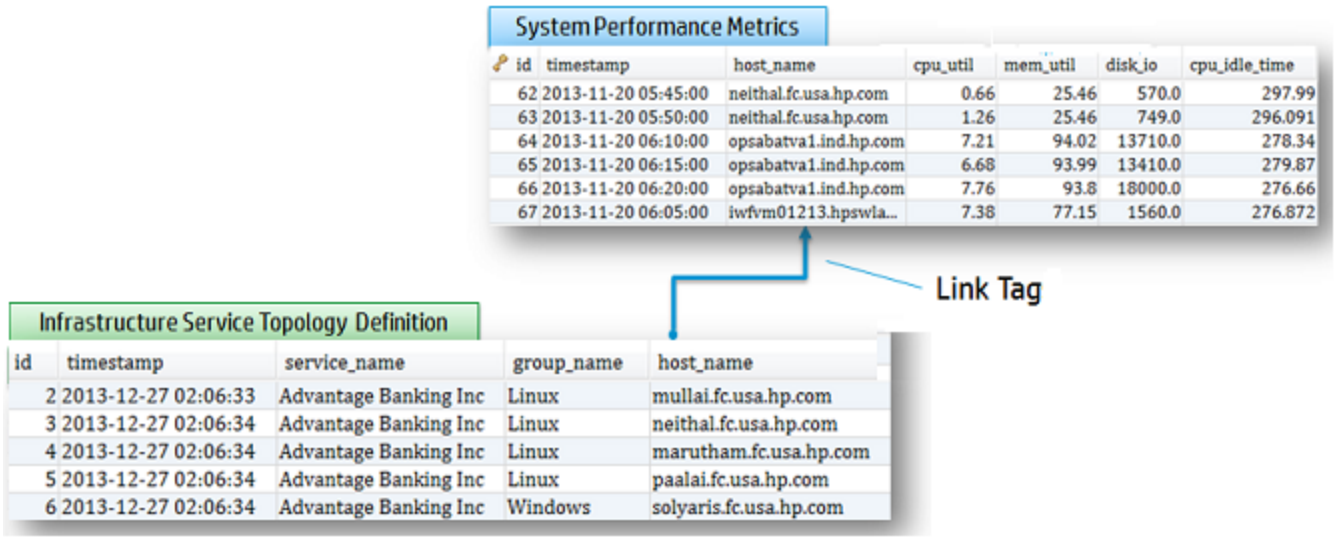
When using keys to identify a column in a collection that you want Operations Analytics to use to match metrics for a specific entity across collections make sure the required column is configured in each collection. For example, you might find that host_name is an attribute that identifies the host in most of your collections. However, perhaps in one or two collections, server_name is the attribute used to identify the host. In this scenario, you specify **host_name** as a key column in the collections that include the host_name attribute and **server_name** as a key column in the collections that include server_name. When a user enters a host_name value in a PQL search query, Operations Analytics looks for that value in all key columns across collections.

Keys also enable you to filter the scope of a search using link tags.

Link tags are special tags that associate two collections. The link between collections is based on one or more columns configured as keys . Values contained in a key column can then be used to filter one collection by the instances in another collection.

Note: You can link only two collections together in a single link tag.

The following diagram illustrates using the **host_name** key column to link an example Infrastructure **Service Topology Definition** collection with its associated system performance metrics stored in the **System Performance Metrics** collection.



Note the following:

- Operations Analytics includes link tags for the collections it provides. If you are an Operations Analytics administrator, see "Configuring Collections using Predefined Templates" in the HP Operations Analytics Configuration Guide for more information about these collections.
- When you define a service using the Topology Manager, Operations Analytics configures the link tags to establish the relationships between the collections for your service. You can then search for information using these relationships. See "Topology Manager" on page 172 for more information.

About Tags

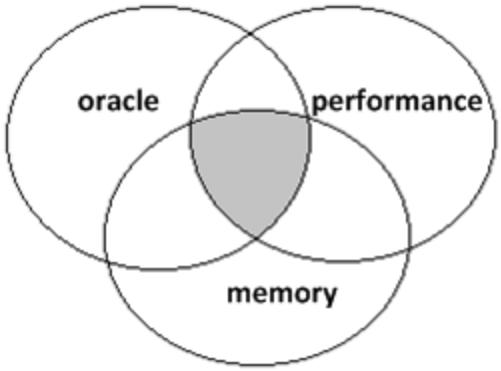
A tag is a word that is associated with a collection or with a metric or attribute that is stored as part of a collection.

Tags are used in the Operations Analytics Phrased Query Language (PQL) to create an Operations Analytics dashboard. They help to define the following:

Note: Tags are not limited to these example uses.

- Entities for which you want information, such as **host**, **database**, and **application**
- Hardware and software components, such as **cpu**, **memory**, **disk**, **interface**, **tablespace**, **process**, and **threads**
- Metrics or problem areas, such as **utilization**, **availability**, **performance**, and **change**

Operations Analytics returns results based on an intersection of the tags used in the search query. For example, the query **oracle memory performance** returns only the metrics that are associated with all three tags (**oracle memory performance**) as represented in the following diagram:



Note: If you include a hostname in your query, Operations Analytics refines the search to include only those metrics associated with the host name you specify.

As an Operations Analytics administrator, you might want to add, edit, or remove tags after they are initially configured. See opsa-tag-manager.sh and "Configure Your Collections" in the HP Operations Analytics Configuration Guide for more information.

To view the tags available for a collection, see "How to View Collection Information" on page 217 or use the opsa-tag-manager.sh command.

See the Uses for Tags table for examples of how you as the Operations Analytics administrator might use tags.

Uses for Tags

Use	Example	Result
Represent the data for an entire collection	If you have configured an HP NNM iSPI Performance for Metrics collection, the tag **performance** might be used for that collection.	When you type **performance** in your phrased search query, the value for all attributes in the NNM iSPI Performance for Metrics collection are considered for use in the metrics displayed.
Provide one or more synonyms for an attribute stored in a collection	The tag **host** might be used as a synonym for the attribute **host_name**	When you type **host** in your search query, Operations Analytics uses the value stored for **host_name** in each collection table for which the tag is defined.

Uses for Tags, continued

Use	Example	Result
Group attributes that provide similar information	The tag **cpu utilization** might be used to represent the following CPU attributes: • cpu_idle_time • cpu_sys_mode • cpu_util_time • cpu_util • cpu_user_mode • cpu_context_switch_rate • cpu_run_queue	When you type **cpu utilization** in your search query, Operations Analytics uses the values stored for the CPU attributes in each collection in which the tag **cpu utilization** is defined.
Focus on attributes that are prototypical	The tag **primary** might be used to tag the most important metric attributes for a specific area, such as cpu). This means that when the user enters **cpu primary** in the search query, the results focus on only a few important metrics, which are tagged as **primary**.	When you type **<*hostname*>cpu** in your search query, Operations Analytics uses the following metrics in its results. • cpu_idle_time • cpu_sys_mode • cpu_util_time • cpu_util • cpu_user_mode • cpu_context_switch_rate • cpu_run_queue When you type <***hostname***>**cpu primary** in your search query, Operations Analytics might use only the following metrics in its results. • cpu_util • cpu_user_mode

Uses for Tags, continued

Use	Example	Result
Group attributes across collections	The tags **performance primary** could be used for the attributes that assist with identifying performance problems across collections. As another example, you might tag all metrics that are useful for identifying status or health information across collections.	When you type **performance primary**, Operations Analytics returns performance metrics from both the HP Operations Smart Plug-in for Oracle and HP Operations Agent collections.
Dynamically extend your collections	Use the same tag name for more than one collection. For example, you might use the tag name **event** and **events** for the following collections: • HP Operations Manager (OM) • HP Operations Manager i (OMi)	When you type <*host name*> **events** in your search query, both the Operations Manager i events and Operations Manager events data is used to return your results.

About Meta Data

Operations Analytics stores collections information as meta data (descriptors). Example meta data information includes:

- Collection table names.

  Note: Operations Analytics stores metrics, topology, inventory, log file, and event information in the form of collection tables. These collection tables are also known as property groups. The columns that represent the metrics collected and that store values within these tables are also known as properties. A property can be either an **attribute**[1] or a **metric**[2].

- Metrics, attributes, and tags per collection.
- The length of time the data is retained per collection.
- Data type information per collection.

About Collectors

A Collector is responsible for collecting data from one or more data sources. The data collected is organized by collections.

Each collector is configured to run in an Operations Analytics Collector Agent.

[1]A descriptor stored in a collection for an entity, such as host_name.
[2]Typically a measurement stored in a collection. For example, CPU utilization.

Each server that is running the Operations Analytics Collector agent is configured as a Collector Appliance.

See "Adding a New HP Operations Analytics Collection" the HP Operations Analytics Configuration Guide for more information.

About Collections

Operations Analytics stores metrics, topology, inventory, log file, and event information in the form of collections. Each collection is associated with a database table in which an Operations Analytics Collector stores the data collected.

Note: These collection tables are identified in the Operations Analytics database as **property_group_uid.** The columns that represent the metrics collected and that store values within these tables are stored in the database as **property_uid**. This is important to know when using the SystemMetaInfo dashboard to identify text strings to include in your search queries.

As the Operations Analytics administrator, you configure one or more data sources per Operations Analytics collection.

See "Collections" on page 174 for more information about possible data sources.

See "Configure Collections" in the HP Operations Analytics Configuration Guide for information about how to configure collections.

Collection Data Sources

Operations Analytics gathers metrics, topology, inventory, event, and log file data from a diverse set of possible sources. The table below describes the details of these sources.

- The Operations Analytics administrator configures the collection data sources.
- Operations Analytics data sources marked with an asterisk (*) indicate the data sources for which Operations Analytics provides configuration templates.

Collection Data Sources

Data Source	Details		Metrics	Inventory	Topology	Log	Events
Business Process Monitor (BPM)*	**Description:**	Collects metrics data from HP Business Process Monitor. Click here for examples of the metrics collected by default: • Application • Application_Id • Transaction • Transaction_Id • Location • Location_Id • Status • Transaction_ Response_Time • Transaction_SSL_ Time • Transaction_ Network_Time • Transaction_ BackEnd_Time • Transaction_Client_ Time	✔				
	Required Software:	HP Business Process Monitor (BPM)					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
Custom CSV files	**Description:**	Collects data that resides in a CSV file.	✔	✔	✔	✔	✔
	Required Software:	No requirements. Many applications export data, such as topology and metrics information, into CSV files. In addition, your network administrator might have written customized scripts to export data to CSV files.					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
Log Files	**Description:**	Collects **raw log**[1] file information. **Note:** These log files must be configured in HP ArcSight Logger or Splunk. If you are an Operations Analytics administrator, see the HP Operations Analytics Configuration Guide for more information. Examples of the types of log files that can be configured for use with Operations Analytics include: • syslog • database • applications • network device log files				✔	
	Required Software:	HP ArcSight Logger					

[1]Log files that contain messages as they appear in the log source from which they are collected. These log files must be configured using the log file management software supported by HP Operations Analytics. See the HP Operations Analytics Support Matrix for more information.

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
Log Files (Structured)	**Description:**	Collects **structured log** file information. **Note:** These log files must be configured as collections. If you are an Operations Analytics administrator, see the HP Operations Analytics Configuration Guide for more information. Examples of the types of log files that can be configured for use with Operations Analytics include: • syslog • database • applications • network device log files				✔	
	Required Software:	HP Operations Analytics or Splunk					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
HP Operations Agent*	**Description:**	Collects global system information. Examples of the type of metric collected by default include host name, time stamp, and global metrics such as CPU total utilization, and disk input and output rate. Click here for examples of the metrics collected by default: • HOSTNAME • AGENTTIMESTAMP • SOURCE • SOURCEID • GBL_ACTIVE_PROC • GBL_ALIVE_PROC • GBL_CPU_CLOCK • GBL_CPU_IDLE_TIME • GBL_CPU_SYS_MODE_UTIL • GBL_CPU_TOTAL_TIME • GBL_CPU_TOTAL_UTIL • GBL_CPU_USER_MODE_UTIL	✔				

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		• GBL_CSWITCH_RATE • GBL_DISK_PHYS_BYTE_RATE • GBL_DISK_PHYS_IO • GBL_DISK_PHYS_IO_RATE • GBL_DISK_PHYS_READ_BYTE_RATE • GBL_DISK_PHYS_WRITE_BYTE_RATE • GBL_DISK_REQUEST_QUEUE • GBL_INTERRUPT_RATE • GBL_MEM_FREE • GBL_MEM_PAGEOUT_RATE • GBL_MEM_UTIL • GBL_NET_ERROR_RATE • GBL_NET_IN_PACKET • GBL_NET_IN_PACKET_RATE • GBL_NET_OUT_PACKET • GBL_NET_OUT_					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		PACKET_RATE • GBL_NET_ PACKET_RATE • GBL_RUN_QUEUE • GBL_STARTED_ PROC • GBL_SWAP_ SPACE_UTIL • GBL_SYSTEM_ UPTIME_ SECONDS See *HP Operations Agent User's Guide* for more information about each of these attributes.					
	Required Software:	HP Operations Manager					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
HP Operations Smart Plug-in for Oracle*	**Description:**	Collects global Oracle database information. Click here for examples of the database information that is collected by default: • HOSTNAME • AGENTTIMESTAMP • SOURCE • SOURCEID • INSTANCENAME • E007_TBLSPCSTATUSCNT • E006_TBLSPFREEPCTCNT • E008_TSBREADRATIOCNT • E009_TSTMPEXNTPCTCNT • E011_TBLSPCFRGMNTCNT • E016_SEGMNTEXTENDCNT • E017_SEGMAXEXTENTCNT	✔				

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		• E018_ SEGEXTRAPIDCNT • E067_ RBSEGMNTSTATCNT • E004_ USERSTMPDFLTCNT • E005_ OBJCTSFORIGNCNT • E019_ SORTDISKRATE • E020_ SORTMEMORYPCT • E052_ SORTTOTALRATE • E021_ BUFFERBUSYPCT • E022_ TOTBUFCACHITPCT • E023_ CURBUFCACHITPCT • E026_ DICTCACHEHITPCT • E027_ LIBCACHRELODPCT • E039_					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		LIBCACGETHITPCT • E040_ LIBCACPINHITPCT • E045_ SHRDPOOLFREEPCT • E059_ CURSORCACHEPCT • E030_ FULLGTBLSCNRATE • E042_ UNLYZTBLINDXPCT • E046_ ROWFETCBYIDXPCT • E070_ PQSERVRSBUSYPCT • E071_ PQSRVHIGHWTRPCT • E074_ PQQUERYRATE • E076_ PQRANGESCANPCT • E044_ COMMITRATE • E058_ ARCHVFREESPCP					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		CT • E060_ REDOUNARCHVDCNT • E062_ BKGRDUMPSPCEPCT • E063_ TRACEFILEADDCNT • E064_ USERDUMPSPACPCT • E065_ COREDUMPSPACPCT • E037_ USERLOGONCNT • E082_ SESSHIGHWATRCNT • E002_ PROCESSSTATUS • E051_ SORTROWSAVGCNT • E041_ FULSHTBLSCNRATE • E047_ TABLESCACHEDCNT • E048_ CHANDROWFTCH					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		PCT • E054_ ROLLBACKRATE • E085_ TRANSACTIONPCT • E056_ ARCHVFREESPCCNT • E057_ ARCHIVEFREQRATE • E066_ ALERTLOGSIZE • E028_ LOCKSUSEDPCT • E029_ SESSWAITLCKCNT • E097_ DISBLDTBLLCKNUM • E032_ REDOLGSPCREQCNT • E033_ REDOALOCLTCHPCT • E034_ REDOCOPYLTCHPCT • E024_ EQWAITSREQPCT					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		• E038_ LTCHOVRLIMITCNT • E043_ EQTIMEOUTREQPCT • E089_ ENQUEUEPCT • E049_ USERCALLRATE • E050_ RCSVUSRCALRATIO • E075_ RCRSVCURSRRATIO • E090_ DSPTCHRBUSYPCT • E091_ NUMDSPTCHRCLNTS • E092_ SHRSRVRREQWTPCT • E093_ SHAREDSERVERPCT" • E094_ SESUGAMEMCURPCT • E095_ SESUGAMEMMAXPCT					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		• E096_ SHRDSRVHWMPCT • E086_ PHYSREADSRATE • E088_ LOGICREADSRATE • E035_ BCKGNDCKPTRATE • E083_ DBWRCKPTRATE • E068_ RBSGMNTSHRNKCNT • E069_ RBSEGWAITPCTCNT • E031_ OPENCRSRPCTCNT • E087_ PROCESSPCT" See *HP Operations Smart Plug-in for Oracle Reference Guide* for more information about each of these attributes.					
	Required Software:	HP Operations Manager					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
HP Network Node Manager i Software (NNMi) Custom Poller*	**Description:**	Collects numeric metrics from any NNMi Custom Poller MIB expressions. Click here for the example metrics collected for Custom Poller by default: • Node Name • Time Stamp (ms) • SOURCE • Node UUID • IP Address • MIB Expression • Poll Interval (ms) • MIB Instance • Metric Value • Display Attribute • Filter Value See the *NNMi Help for Operators* for more information about each of these attributes.	✔				
	Required Software:	HP Network Node Manager i Software (NNMi)					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
HP Network Node Manager iSPI Performance for Metrics*	**Description:**	Collects interface and node component performance information from HP NNM iSPI Performance for Metrics. Examples of the information collected include: • Interface health extension pack metrics • Component health extension pack metrics Click here for examples of the metrics collected for interface health by default: • Node Name • Source • Interface Name • Interface Description • Interface Index • Interface Type • Interface Physical Address • Interface Speed • Interface ID • Interface UUID • Node Contact • Node Location	✔				

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		• Node Family • Node Vendor • Node ID • Node UUID • SONET Line ESs Rate • LAN FCS Error Rate • Interface Utilization • Interface Utilization In • Multicast Packets Out • SONET Far End Path ESs Rate • Queue Drop Ouput Rate • Broadcast Packets Out • Nonunicast Packets In • Multicast Paackets • DSx 1UASs Rate • Nonunicast Packets Out • SNMP Respone Time • Packet Size Bytes In • DSx 1CSSs Rate					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		• Error Rate Out • Sucessful Retry Count • Throughput Out bps • DSx 1LCVs/sec • Unresponsive Target • LAN Collision Rate • DSx 3LESs Rate • Unicast Packets Out • Discard Packets In • Volume Bytes • SONET Section SESs Rate • WLAN FCS Error Rate • Volume Packets out • Unicast Packets In • DSx 1SESs Rrate • Error Packets In • DSx 3UASs Rate • DSX 3CSESs Rate • Error Rate • SONET Far End Line SESs Rate • Packet Size					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		• Utilization Out • Volume Packets In • SONET Path SESs Rate • Discard Rate Out • Volume Packets • Broadcast Packets In • Nonunicast Packets • DSx 3PSESs Rate • SONET Section CVs/sec • Error Rate In • Volume Bytes In • Broadcast Packets • SONET Far End Path SESs Rate • SONET Far End Path CVs/sec • DSx 1BESs Rate • DSx 3SEFSs Rate • DSx 1SEFSs Rate • Discard Rate In • DSx 3CESs Rate • Multicast Packets In • SONET Path					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		CVs/sec • DSx 3PCVs/sec • SONET Path UASs Rate • DSx 3CVVs/sec • SONET Line SSESs Rate • SONET Section SEFSs Rate • DSx 1 LESs Rate • SONET Far End line SUASs Rate • SONEt Line SUASs Rate • DSx 1ESs Rate • DSx 3PESs Rate • Errors Packets Out • SONET Far End Line SESs Rate • Throughput in bps • Volume Bytes Out • Unicast Packets • SONET Path ESs Rate • Discard Packets • DSx 1DMs Rate • Throughput bps					

Collection Data Sources, continued

Data Source	Details	Metrics	Inventory	Topology	Log	Events
	• SONET Section ESs Rate • SONET Far End Line CVs/sec • Discard Rate • Error Packets • DSx 1PCVs/sec • Packet Sze Byte Out • Queue Drops Input Rate Threshold Exception Rate • Availablity Threshold Exception Rate • Queue Drops Output Rate Threshold Exception Rate • Utilization In Threshold Exception Rate • Discard Rate Threshold Exception Rate • Discard Rate Out Threshold Exception Rate • Utilization Threshold Exception Rrate • Error Rate In Threshold Exception Rate • WLAN FCS Error					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		Rate Threshold Exception Rate • LAN FCS Error Rate Threshold Exception Rate • Error Rate Threshold Exception Rate • Discard Rate In Threshold Exception Rate • Error Rate Out Threshold Exception Rate • Utilization Out Threshold Exception Rate • Interface Alias • Unknown Protocol - Packets • DSx 3LCVs/sec • Reboot Flag • Discards - Packets Out • LAN Deferred Frames • LAN Collision Count • RTS Success Count • Maxed Out Transmit Attempts • ACK Failure Count					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		• RTS Failure Count • Stations Deauthenticated • WLAN FCS Error Count • Undecryptable Frames • Queue Drops - Input Packets • Frame Duplicate Count • Stations Associated • Queue Drops - Input Rate • Stations Roamed Away • Num Active Repeaters • LAN Alignment Errors • Stations Roamed In • Queue Drops - Output Packets • Num Active Bridges • Num Active Wireless Clients • Stations Disassociated • Received Fragment Count					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		• LAN FCS Error Count • Transmitted Fragment Count • Sonet Far End Path UASs Rate • Stations Authenticated • Utilization Out - Forecast Baseline (4 week) • Utilization In - Forecast Baseline (4 week) • Utilization Out - Forecast Baseline (8 week) • Utilization In - Forecast Baseline (8 week) • Utilization Out - Forecast Baseline (12 week) • Utilization In - Forecast Baseline (12 week) • Utilization Out - Baseline Exception Count • Utilization In - Baseline Exception Count • Utilization Out -					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		Baseline Exception Rate • Utilization In - Baseline Exception Rate • Utilization Out - Threshold Exception Count • Discard Rate Out - Threshold Exception Count • LAN FCS Error Rate - Threshold Exception Count • Discard Rate In - Threshold Exception Count • Error Rate - Threshold Exception Count • WLAN FCS Error Rate - Threshold Exception Count • Error Rate Out - Threshold Exception Count • Utilization In - Threshold Exception Count • Queue Drops - Input Rate - Threshold Exception Count • Queue Drops - Output Rate -					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		Threshold Exception Count • Error Rate In - Threshold Exception Count • Availability - Threshold Exception Count • Utilization - Threshold Exception Count See the HP Network Node Manager iSPI Performance for Metrics online help for more information about these attributes. Click here for examples of the metrics collected for component health by default: • Node Name • SOURCE • Component Name • Component Type • Component ID • Component UUID • Node Contact • Node Location • Node Family • Node Vendor					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		• Node ID • Node UUID • Disk Space Free - MB • SNMP Response Time (msecs) • Buffer Miss Rate • ICMP ResponseTime (Milliseconds) • CPU 5min Utilization • Backplane Utilization • Disk Space Utilization • Buffer Failure Rate • Disk Space Used - MB • Free Memory • Memory Utilization • Buffer Utilization • Disk Space Total - MB • ICMP ResponseTime (Milliseconds) - Threshold Exception Rate • Buffer Utilization - Threshold Exception					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		Rate • Backplane Utilization - Threshold Exception Rate • Buffer Failure Rate - Threshold Exception Rate • Disk Space Utilization - Threshold Exception Rate • Memory Utilization - Threshold Exception Rate • CPU 5min Utilization - Threshold Exception Rate • Buffer Miss Rate - Threshold Exception Rate • Reboot • Node Reachability • Buffer Hits • Buffer NoMemory Rate • Buffer Failures • Node Availability • CPU 5sec Utilization • Buffer Misses • CPU 1min Utilization					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		• Buffer Used • CPU 1min Utilization - Forecast Baseline (4 week) • CPU 5sec Utilization - Forecast Baseline (4 week) • CPU 5min Utilization - Forecast Baseline (4 week) • Memory Utilization - Forecast Baseline (4 week) • Buffer Utilization - Forecast Baseline (4 week) • ICMP ResponseTime (Milliseconds) - Forecast Baseline (4 week) • Backplane Utilization - Forecast Baseline (4 week) • Disk Space Utilization - Forecast Baseline (4 week) • CPU 1min Utilization - Forecast Baseline (8 week) • CPU 5sec Utilization - Forecast Baseline (8 week) • CPU 5min Utilization					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		- Forecast Baseline (8 week) • Memory Utilization - Forecast Baseline (8 week) • Buffer Utilization - Forecast Baseline (8 week) • ICMP ResponseTime (Milliseconds) - Forecast Baseline (8 week) • Backplane Utilization - Forecast Baseline (8 week) • Disk Space Utilization - Forecast Baseline (8 week) • CPU 1min Utilization - Forecast Baseline (12 week) • CPU 5sec Utilization - Forecast Baseline (12 week) • CPU 5min Utilization - Forecast Baseline (12 week) • Memory Utilization - Forecast Baseline (12 week) • Buffer Utilization - Forecast Baseline (12 week)					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		• ICMP ResponseTime (Milliseconds) - Forecast Baseline (12 week) • Backplane Utilization - Forecast Baseline (12 week) • Disk Space Utilization - Forecast Baseline (12 week) • Buffer Utilization - Baseline Exception Count • Disk Space Utilization - Baseline Exception Count • CPU 1min Utilization - Baseline Exception Count • Memory Utilization - Baseline Exception Count • ICMP ResponseTime (Milliseconds) - Baseline Exception Count • CPU 5min Utilization - Baseline Exception Count • Backplane Utilization - Baseline Exception Count • CPU 5sec Utilization					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		- Baseline Exception Count • Backplane Utilization - Baseline Exception Rate • CPU 1min Utilization - Baseline Exception Rate • Memory Utilization - Baseline Exception Rate • CPU 5sec Utilization - Baseline Exception Rate • ICMP ResponseTime (Milliseconds) - Baseline Exception Rate • Disk Space Utilization - Baseline Exception Rate • CPU 5min Utilization - Baseline Exception Rate • Buffer Utilization - Baseline Exception Rate • Disk Space Utilization - Threshold Exception Count • CPU 1min Utilization - Threshold Exception Count					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		• CPU 5sec Utilization - Threshold Exception Count • Buffer NoMemory Rate - Threshold Exception Count • ICMP ResponseTime (Milliseconds) - Threshold Exception Count See the HP Network Node Manager iSPI Performance for Metrics online help for more information about these attributes.					
	Required Software:	HP Network Node Manager iSPI Performance for Metrics					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
HP Operations Manager (OM) events*	**Description:**	Collect events generated by HP Operations Manager (OM). Click here for the OM event metrics collected by default: • EventID • TimeReceivedTimeStamp • TimeCreatedTimeStamp • Severity • NodeName • State • EventText • MessageGroup • EventObject • MsgSource • Application • AutoState • AutoAcknowledge • OperatorAcknowledgeFlag • Service		✔			✔
	Required Software:	HP Operations Manager					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
HP Operations Manager i (OMi) events*	**Description:**	Collects events generated by HP Operations Manage i Software. Click here for examples of the event information collected by default: • EVENT • ID • DATE_CREATED • DATE_RECEIVED • TIME_STATE_ CHANGED • TITLE • DESCRIPTION • PRIORITY • STATE • SEVERITY • TYPE • CATEGORY • SUBCATEGORY • APPLICATION • ASSIGNED_ GROUP • ASSIGNED_USER • CIREF_ID • HOSTREF_ID					✔

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
		• HOSTINFO_IPADDRESS • HOSTINFO_DNSNAME • ORIGINATING_IPADDRESS • ORIGINATING_DNSNAME • SENDER_IPADDRESS • SENDER_DNSNAME • PARENT_ID • RC_FLAG • POLICY_TYPE • POLICY_NAME • CORRELATION_TYPE • CORRELATION_RULE_ID • LOG_ONLY See *HP Operations Manager Administrator's Reference* for more information about each of these attributes.					
	Required Software:	HP Business Service Management (BSM)					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
	Required Software:	HP Business Service Management (BSM)					
HP Run-Time Service Model (RTSM)*	**Description:**	Collects Configuration Item (CI) inventory information that is stored in BSM. Click here for examples of the inventory information collected by default: • CiId • CiType • display_label • name • description		✔			
	Required Software:	HP Business Service Management (BSM)					

Collection Data Sources, continued

Data Source	Details		Metrics	Inventory	Topology	Log	Events
HP SiteScope	**Description:**	Collects monitoring information, such as: • CPU utilization • memory utilization • pages per second • memory pool size **Note:** The list of metrics varies depending on the collection you create. See *HP SiteScope Monitor Reference* for more information about the monitoring attributes available.	✔				
	Required Software:	HP SiteScope					

Tasks

How to View Collection Information

Operations Analytics stores metrics, topology, inventory, log file, and event information in the form of collection tables. Becoming familiar with the data collected is useful to help determine the type of queries you might want to perform. For example, you can include the collection name (AQL search query) or its associated tag name (PQL search query) to return all data from a specified collection. Because each collection is stored as part of a database table, you might also specify a collection column name (for example, cpu_util) to return a subset of data across one or more collections. See "Search Tool" on page 26 for more information.

When viewing collection information, use the mapping described in the table below to determine the collection information to include in your queries.

Note: As shown in the mapping table, collection tables are also known as property groups. The columns that represent the metrics collected and that store values within these tables are also

known as properties. A property can be either an **attribute**[1] or a **metric**[2]. The property groups are uniquely identified by **property group uid** and properties are uniquely identified within a property group by **property uid**. When specifying a collection name or column name in your search query, use the **property group uid** and **property uid** values.

Column Descriptions for Meta Data Tables

Information	Table Column Name Displayed in the Dashboard
Collection names	property group uid
Columns (metrics or attributes) per collection	property uid
Tag names, if any, per collection or column	tag name
Columns defined as keys.	Look for rows in which the iskey value is true

Note: You can also use opsa-tag-manager.sh to view tag information.

To view collection information:

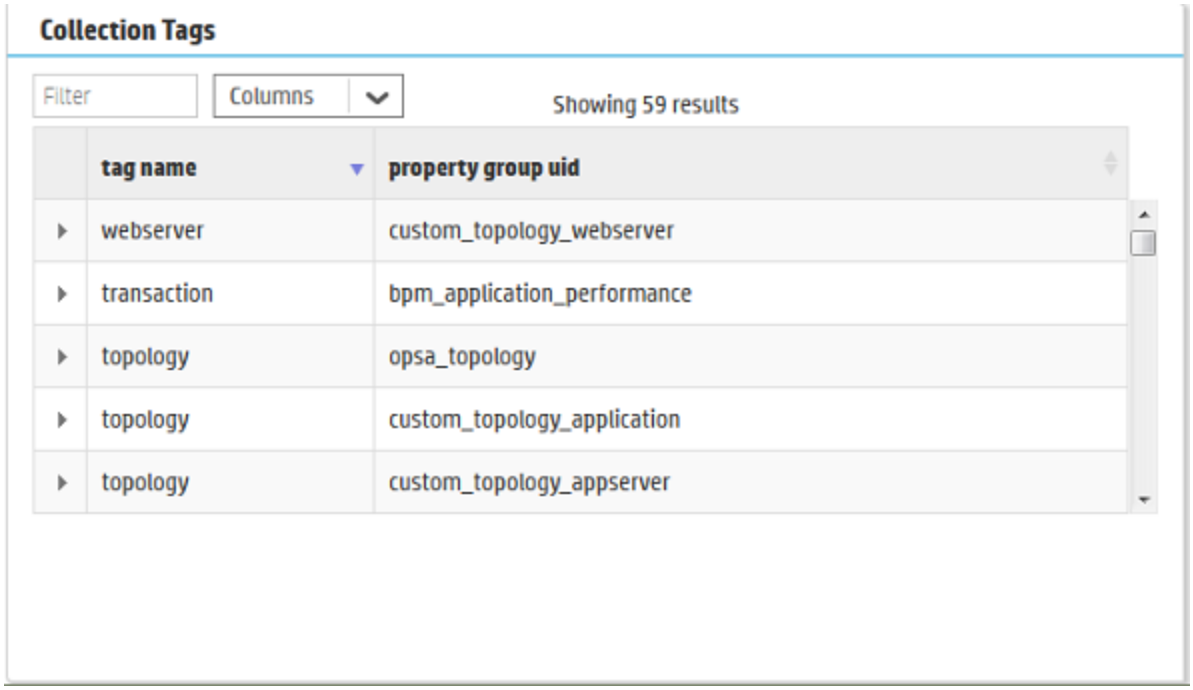
1. Navigate to the **Dashboards** menu.
2. Select **SystemMetaInfo**.

   Tip: You can also access this dashboard using the **Show SystemMetaInfo** option when adding or editing a query pane.

   Operations Analytics displays tables that include the following information:

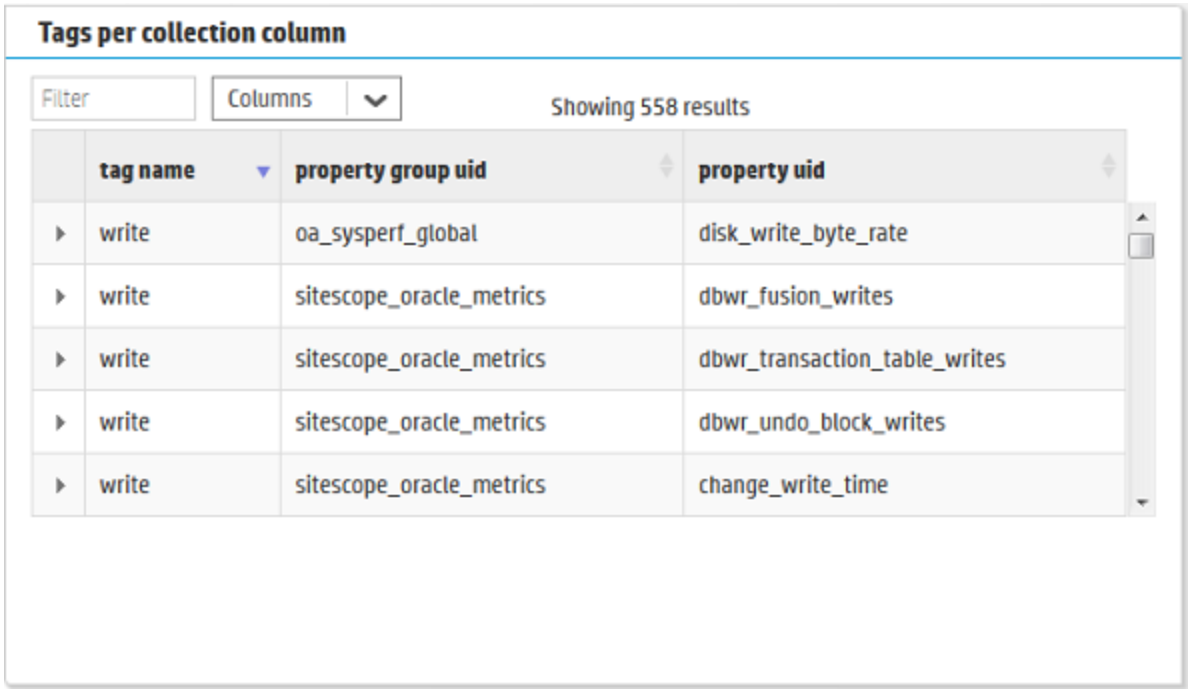
[1]A descriptor stored in a collection for an entity, such as host_name.
[2]Typically a measurement stored in a collection. For example, CPU utilization.

- Tags, if any, assigned to each collection (property group uid):

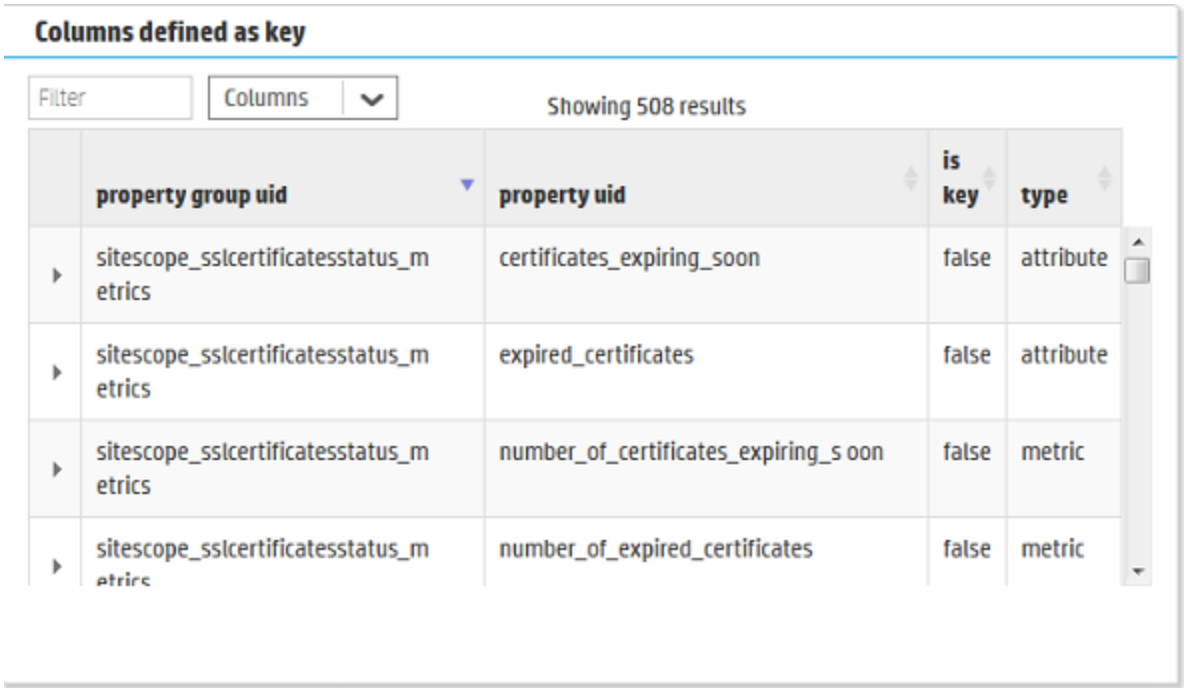


- Tags associated with columns (property uid) within each collection.

Tip: You can also access this table using the **Show Tags** option when adding or editing a query pane. See "Dashboards and Query Panes" on page 13 for more information.



- Columns that are configured as key columns (iskey) in each collection. Key columns are used to filter metrics across collections.



See "About Table Data" on page 160 for more information about working with tables.

Chapter 11: Manage Users and Tenants

This topic defines user accounts, user groups, and tenants and contains the procedures required to work with them.

To access

Click **Settings** and select **User Management**.

Learn About

About User Accounts

As an Operations Analytics administrator, you must configure a User Account for each user who needs to access the Operations Analytics graphical user interface.

Note the following:

- User Accounts must be unique across all Tenants.

  Tip: To ensure the user name is globally unique, enter a user's email address as the user name.

- Each User Account must be assigned to a User Group.

To create a user account, see "Add a User Account" on page 224, opsa-user-manager.sh, and "Configuring Tenants and Collections" in the HP Operations Analytics Configuration Guide.

The first time you log on, you will need to change the default password. User account passwords must be a minimum of eight characters long, and must include at least one uppercase character, one lowercase character, one digit, and one special character.

After ten failed attempts to access Operations Analytics from a specific user account, Operations Analytics denies access to users attempting access with this user account. This account restriction lasts for ten minutes. If you have any Operations Analytics access problems, discuss them with your Operations Analytics administrator.

About User Groups

User Groups are pre-defined in Operations Analytics and determine which tasks each User Account that is assigned to the User Group can perform.

- User Accounts must be unique across all tenants.
- All User Groups have access to the Operations Analytics graphical user interface.
- You cannot add a new User Group to Operations Analytics.

- A User Account was assigned to the **Super Admin** User Group when Operations Analytics was installed.
- See opsa-tenant-manager.sh and "Configuring Tenants and Collections" in the HP Operations Analytics Configuration Guide for information about assigning a user to a User Group.

Pre-defined User Groups

User Group	Description	Supported Tasks
Super Admin	**Note:** Operations Analytics permits only one Super Admin user. The user account assigned to this user group has access to the following information for each tenant defined: • User Accounts • User Groups	Add, modify, and delete tenants. Add, modify, and delete user accounts assigned to the Tenant Admin user group.
Tenant Admin	User accounts assigned to this User Group have access to the following information only for the tenant to which they are assigned: • Collectors • Collections • Meta Data • Tags • User Accounts • User Groups	Add, modify, and delete user accounts. Manage the collectors, collections, meta data, and tags for a specified tenant.

Pre-defined User Groups, continued

User Group	Description	Supported Tasks
User	User accounts assigned to this User Group have access to the Operations Analytics graphical user interface and to only the meta data and data for the tenant to which they are assigned.	Access and perform tasks using the Operations Analytics Dashboards. **Note:** Users assigned to this user group can also add and delete tags from a collection. See opsa-tag-manager.sh and "Configuring Tenants and Collections" in the HP Operations Analytics Configuration Guide for more information.

New users are automatically assigned to a predefined user group. The user group to which a new user is assigned depends on the user group to which you are assigned when adding a new user.

User Groups Assigned to New Users

Your User Group	User Group Automatically Assigned to the New User
Super Admin	Tenant Admin
Tenant Admin	User

About Tenants

Operations Analytics supports multi-tenancy. This means one instance of Operations Analytics can serve multiple customers. Tenants ensure isolation of meta data and data across customers. The meta data includes the following:

- Collections
- Database schema
- Tags
- Dashboards
- User Accounts

For example, if you are a Manage Service Provider or Software as a Service Provider with multiple customers, tenants enable you to ensure that each customer accesses only the data for its data center or network.

When you install Operations Analytics, by default Operations Analytics creates the **opsa_default** tenant.

To create one or more tenants, see opsa-tenant-manager.sh and "Configuring Tenants and Collections" in the HP Operations Analytics Configuration Guide.

Tasks

Add a User Account

1. Click **Settings** and select **User Management**.

   Operations Analytics displays the **User Management** form.

   Note: You must belong to either the Super Admin or Tenant Admin User Group to access the **User Management** option.

2. Click .

   Operations Analytics displays the **Add User** form.

3. If you belong to the Super Admin User Group, in the **Tenant** attribute, do one of the following:

   a. Select a tenant name to which you want to assign the user account.

   b. Enter the name of a tenant you want to create. Click **No matches found - Click to Add** . In the **Add Tenant** dialog, click **OK.**

      The new user is assigned to this tenant.

4. In the **User Name** attribute, enter the user account name of at least four characters.

   Tip: If you are using Public Key Infrastructure (PKI) authentication, the user name must be an email address.

5. In the **Password** attribute, enter a password that contains at least four characters.

6. In the **Confirm Password** attribute, re-enter the password for this user account.

7. Click **Save**.

   Operations Analytics lists the new user account in the **User Management** table with its associated user group and tenant.

You can also add a user account using the opsa-user-manager.sh script. Run the following command:
$OPSA_HOME/bin/opsa-user-manager.sh -add -loginUser <*Super Admin or Tenant Admin User Name*> -loginPassword <*password*> -newUser <*new username*> -newUserPassword <*new user password*>

Note: See the opsa-user-manager.sh reference page (or the Linux manpage) for more information.

After creating a new user use the opsa-user-manager.sh script, to show a list of users run the commands shown in the following examples:

- **To list Tenant Admin users**: $OPSA_HOME/bin/opsa-user-manager.sh -list -loginUser opsaadmin -loginPassword <*opsaadmin password*>
- **To list users by Tenant**: $OPSA_HOME/bin/opsa-user-manager.sh -list -loginUser <*Tenant Admin User*> -loginPassword <*Tenant Admin Password*>

You can delete a user account using the opsa-user-manager.sh script. Run the following command: $OPSA_HOME/bin/opsa-user-manager.sh -delete -loginUser <*Tenant Admin User*> -loginPassword <*Tenant Admin Password*> -user <*username*>

Change Your User Account Password

You can change your user account password at any time.

To change your user account password:

1. In the upper right-hand corner of the Operations Analytics console, click your user account name.
2. Select **Change Password**.

   The **Change Password** dialog box appears.
3. In the **Current Password** field, enter the current password.
4. In the **New Password** field, enter the password value.

   The password must be at least four characters.
5. In the **Confirm Password** field, re-enter the new password.
6. Click **Update**.

You can also modify the password for a user account using the opsa-user-manager.sh script. Run the following command:
$OPSA_HOME/bin/opsa-user-manager.sh -modify -loginUser <*username*> -loginPassword <password> -newUserPassword <*new user password*>

Note: Run the opsa-user-manager.sh command as an opsa user, not as a root user. Running opsa-user-manager.sh as a root user is not supported.

Note: See the opsa-user-manager.sh reference page (or the Linux manpage) for more information.

Add a Tenant

As an Operations Analytics administrator, if you belong to the **Super Admin** User Group, you can add one or more tenants.

- You can also use opsa-tenant-manager.sh to add tenants to Operations Analytics.
- If you do not configure one or more tenants, Operations Analytics stores all of the meta data, collection and query information in the **opsa_default** tenant.
- User account names must be unique across all tenants.

To add a tenant:

1. Click **Settings** and select **User Management**.

   Operations Analytics displays the **User Management** form.

   Note: You must belong to either the Super Admin or Tenant Admin User Group to access the **User Management** option.

2. Click .

   Operations Analytics displays the **Add User** form.

3. If you belong to the Super Admin User Group, in the **Tenant** attribute, enter the name of a tenant you want to create. Tenant names cannot begin with a number. The initial alpha character can be followed by alphanumeric characters (including an underscore).

4. Click **No matches found - Click to Add** .

5. In the **Add Tenant** dialog, click **OK.**

Chapter 12: Administrator Tasks

As an Operations Analytics administrator, you perform the tasks described in the table below to enable Operations Analytics users to proactively manage and troubleshoot IT operations problems.

For example, after you have initially installed and configured Operations Analytics, you might find that you want to use additional data sources and configure the associated collections.

Administrator Tasks

Category	Task	Sections or Help Topic	Command
Maintain Collections and Collectors	Plan for each new data source and subsequent collection configuration.	"Planning Your Deployment" in the HP Operations Analytics Installation Guide.	
	In multiple Operations Analytics server environments only. Designate the Operations Analytics server from which to configure all collections.	"Configuring Tenants and Collections" in the HP Operations Analytics Configuration Guide.	
	Create the collection template for each additional collection.	"Adding a New HP Operations Analytics Collection" in the HP Operations Analytics Configuration Guide.	opsa-collection-config.sh
	Configure your collection templates to match your IT environment.	"Configuring Tenants and Collections" in the HP Operations Analytics Configuration Guide.	opsa-collection-config.sh
	Optional. Add one or more tenants	"Add a Tenant" on page 225 "Creating a Tenant" in the HP Operations Analytics Configuration Guide.	opsa-tenant-manager.sh
	Optional. Delete one or more tenants. **Note:** Be sure to remove a collection registration for any tenant that will be removed.	"Deleting a Tenant" and "Remove a Collection Registration for a Tenant" in the HP Operations Analytics Configuration Guide.	opsa-tenant-manager.sh
	Optional. Associate each collection with a tenant. **Note:** You must first create the tenant to which you want to associate a collection.	"Configuring Tenants and Collections" in the HP Operations Analytics Configuration Guide.	opsa-tenant-manager.sh

Administrator Tasks , continued

Category	Task	Sections or Help Topic	Command
	Optional. For each tenant, create a user account for the **Tenant Admin** and **User** User Groups.	"Manage Users and Tenants" on page 221 "Configuring Tenants and Collections" in the HP Operations Analytics Configuration Guide.	opsa-tenant-manager.sh
	Configure a collector for each new collection.	"Configuring Tenants and Collections" in the HP Operations Analytics Configuration Guide.	opsa-collection-config.sh
	Configure additional collectors for one or more existing collections.	"Installing and Configuring the Operations Analytics Collector Appliance using the VMware vSphere Client" in the HP Operations Analytics Installation Guide and "Configuring Tenants and Collections" in the HP Operations Analytics Configuration Guide.	opsa-collection-config.sh
	Back up your collection configuration on the Operations Analytics server. The collection configuration directory is: `/opt/HP/opsa/conf/collection`	"Configuring Tenants and Collections" in the HP Operations Analytics Configuration Guide.	
	Troubleshoot collection problems	"Troubleshooting Operations Analytics Collections" in the HP Operations Analytics Configuration Guide.	
	Communicate collection names and meta data information to your users.	See "Communicating Collection Names and Meta Data Information to your Users" in the HP Operations Analytics Configuration Guide.	

Administrator Tasks , continued

Category	Task	Sections or Help Topic	Command
Define a Service	Set collection retention periods.	See "Setting Collection Retention Periods" in the HP Operations Analytics Configuration Guide.	
	View the collection information stored in Operations Analytics.	"How to View Collection Information" on page 217	
	Topology Manager enables you to group together hosts that are of interest to you, and view them in Operations Analytics as a **service**. You can group hosts together based on their function, their location, or any other grouping that is meaningful to you when organizing your services.	"Topology Manager" on page 172	
Create AQL Functions	*Optional*. Write Analytic Query Language (AQL) functions using a text editor.	"About Analytics Query Language (AQL) Functions" on page 146	
Import AQL Functions	*Optional.* Import your AQL functions.	"Import Analytic Query Language (AQL) Functions" on page 234	opsa-aql-module-manager.sh
Maintain User Accounts	Add, modify, or delete one or more user accounts.	"Manage Users and Tenants" on page 221 "Maintaining User Accounts" in the HP Operations Analytics Configuration Guide.	

Administrator Tasks , continued

Category	Task	Sections or Help Topic	Command
Maintain HP Operations Analytics	Check the system health of Operations Analytics.	"Check the Health of Operations Analytics" below "Checking Operations Analytics System Health" in the HP Operations Analytics Configuration Guide.	
	Back up the Operations Analytics database.	"Maintaining the HP Operations Analytics Database" in the HP Operations Analytics Configuration Guide.	
	View license information.	Access Help and About Information from the help menu.	

Check the Health of Operations Analytics

Operations Analytics provides two methods for checking the health of servers running the Operations Analytics service:

Command Line Interface

The table below describes the commands used to check the status of Operations Analytics:

Command	Description
opsa-server status	Check the status of the Operations Analytics service **Note:** The opsa-server command must be run on the Operations Analytics server.
opsa-collector status	Checks the status of the collector service on the Collector Appliance. **Note:** The opsa-collector command must be run on the Operations Analytics Collector Appliance.
opsa-loader status	Checks the status of the loader service on the Collector Appliance. **Note:** The opsa-loader command must be run on the Operations Analytics Collector Appliance.

OpsaSystemHealth Dashboard

Use the OpsaSystemHealth dashboard to investigate the health of the Operations Analytics servers. The table below describes the query panes available.

Note: If you view the message that no data is available, this might mean you do not have the required software to collect the expected data. See the **Required Software** column of the table below. Also see "Checking Operations Analytics System Health" in the HP Operations Analytics Configuration Guide for the configuration steps required to display this dashboard information.

Query Pane	Description	Required Software
System Metrics Trends	Use this visualization to determine server health for the Operations Analytics servers. Displays the average value over time for the following metrics for each server running the Operations Analytics service: • System up time • CPU utilization	HP Operations Agent
Service components with max CPU utilization of nodes	Use this visualization to determine the servers running Operations Analytics software. Displays topology information for the Operations Analytics service, including the following servers: • Operations Analytics server • Operations Analytics collector servers • HP logger servers • HP Vertica database servers Also displays the CPU utilization for each of the Operations Analytics servers.	Operations Analytics only
Recent Logs from OPSA service components	Use this visualization to troubleshoot any Operations Analytics log file error messages. Displays all log file messages for servers running the Operations Analytics service.	Operations Analytics only

Define Analytic Query Language (AQL) Functions

By default, Operations Analytics provides several Analytic Query Language (AQL) functions to assist you with creating AQL queries, AQL functions, and associated dashboards.

You can write your own AQL functions using a text editor and then import these functions into Operations Analytics. Each text file you create can contain any number of AQL functions. A set of AQL functions that reside in a single file are known as an AQL module.

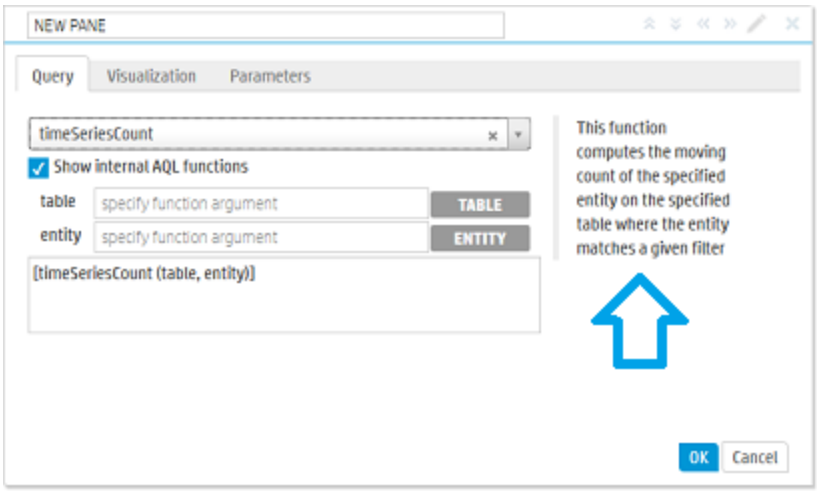
Tip: Use the bpm_functions.aql module as an example. This AQL module contains several AQL functions that can be used as a template for creating your own. They reside in the `$OPSA_HOME/inventory/lib/hp/aql` directory.

You can also view these AQL functions when you use the Add A Query Pane option from an Operations Analytics dashboard. See "Dashboards and Query Panes" on page 13 for more information.

Note: To view the AQL query associated with each AQL function provided by Operations Analytics, look at the .aql files in $OPSA_HOME/inventory/lib/hp/aql or use the opsa-aql-module-manager.sh command.

When creating AQL functions to be imported, note the following:

- The comment preceding each AQL function is displayed as the description for the AQL function selected as shown in the following example:



- As a best practice, name your file using a .aql extension.
- As a best practice, use the opsa-aql-module-manager.sh validate option to ensure your module will import.

- As a best practice, place your file in the `$OPSA_HOME/inventory/lib/user/aql` directory before it is imported. This helps to ensure that the file is not overwritten when upgrading to a new version of Operations Analytics.
- To make your AQL functions available to your user community, use the opsa-aql-module-manager.sh command. This command imports the AQL functions defined in your module into the Operations Analytics database and makes them available to your user community by default.

Also see "Import Analytic Query Language (AQL) Functions" below.

Import Analytic Query Language (AQL) Functions

By default, Operations Analytics provides several Analytic Query Language (AQL) functions to assist you with creating AQL queries, AQL functions, and associated dashboards. See "Define Analytic Query Language (AQL) Functions" on the previous page for more information about the AQL functions that Operations Analytics provides.

You can write your own AQL functions using a text editor and then import these functions into Operations Analytics. Each text file you create can contain any number of AQL functions. Each set of AQL functions contained in a single file is known as a **module**.

Use the opsa-aql-module-manager.sh command to manage the AQL functions. that you create.

When using the opsa-aql-module-manager.sh command, note the following:

- You must specify the tenant name for which the AQL functions should be available.
- Use file names that identify the types of AQL functions contained in each file.
- You define the <module_name> in the first line of each file; for example:

  module <module_name>;

- You validate, list, and delete modules using the module name.

Use the opsa-aql-module-manager.sh command to perform the following tasks:

Validate the AQL functions included in an module file

Enter the following command:

```
opsa-aql-module-manager.sh -t <tenant_name> -v <file_name>
```

```
Note:The opsa-aql-module-manager.sh script does not currently detect some sy
ntax errors,
 such as unbound variables referenced within the body of an AQL function. Ta
ke extra care
when creating and editing your AQL functions.
```

Import an AQL module

Enter the following command:

```
opsa-aql-module-manager.sh -t <tenant_name> -i <file_name>
```

When importing AQL functions, note the following:

- After importing your AQL functions, all functions are available to the user community in the specified tenant.
- To replace or redefine AQL functions, you must make the appropriate changes to the .aql module and re-import the file.

List all AQL modules that have been imported into Operations Analytics

Enter the following command:

```
opsa-aql-module-manager.sh -t <tenant_name> -l modules
```

List the AQL functions contained in a module that has been imported into OpsA

Enter the following command:

```
opsa-aql-module-manager.sh -t <tenant_name> -l <module_name>
```

See opsa-aql-module-manager.sh for more information.

Glossary

A

attribute

A descriptor stored in a collection for an entity, such as host_name.

C

category

A folder that is used to organize your AQL modules.

collections

Operations Analytics stores metrics, topology, inventory, log file, and event information in the form of collection tables. Each collection is associated with a database table in which an Operations Analytics Collector stores the data collected.

D

Database schema

Table, column, attribute, and data type information per collection.

K

Knowledge Content

An xml file that configures a predefined dashboard. Each Knowledge Context includes a name, the entities for which the dashboard displays information, phrases to help identify the Knowledge Context, as well as the queries that return the dashabord and any filters to use before the data is returned.

M

metric

Typically a measurement stored in a collection. For example, CPU utilization.

O

outlier

A data point that is outside of the normal range based on the data collected to date.

Q

query pane

Displays the results of an Analytic Query Language (AQL) query, AQL function, or AQL expression. If you use the Phrased Query Language (PQL) in your search, HP Operations Analytics converts the PQL query to one or more AQL queries and subsequent query panes.

R

raw logs

Log files that contain messages as they appear in the log source from which they are collected. These log files must be configured using the log file management software supported by HP Operations Analytics. See the HP Operations Analytics Support Matrix for more information.

raw metrics

Metrics to which an overall aggregate or moving aggregate analytic function is applied.

S

structured log files

Fragments of log file data that are stored as collections in HP Operations

Analytics. Structured logs are log files that are configured as collections. These collections are created so that users can perform analytics on the log file contents. For example, you might want to query for all outliers by host name and application for a particular time range.

T

tag

A word or phrase that is associated with a metric, topology, event, or log file attribute that is stored as part of a collection in HP Operational Analytics. These tags can be used in the HP Operational Analtyics search query as synonyms for the attributes stored in HP Operational Analytics collection tables. They are also used to make metrics display names more meaningful. Tags are provided by HP Operational Analytics and can also be defined by the HP Operational Analytics administrator.

We appreciate your feedback!

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on HP Operations Analytics Help (Operations Analytics 2.10)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to sw-doc@hp.com.