Quest™

Foglight™ APM 5.9.11

Reference Guide

owners.

Legend

WARNING: A WARNING icon indicates a potential for property damage, personal injury, or death.

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

IMPORTANT NOTE, **NOTE**, **TIP**, **MOBILE**, or **VIDEO:** An information icon indicates supporting information.

Foglight APM Reference Guide
Updated - July 2017
Foglight Version - 5.7.5.8
Cartridge Version - 5.9.11

Contents

1

Foglight APM views reference

Foglight™ APM for Real User Experience (hereafter called Foglight APM) is an appliance-based application performance management solution. This *Reference Guide* describes the monitoring-type dashboards installed with Foglight APM, the details collected from live web traffic, and the Foglight-generated metrics that populate the dashboards and views.

This guide is intended for Foglight users with the role of APM Operator or APM Administrator.

This section describes the monitoring dashboards and views included with Foglight APM Appliances. Each view description contains links to the metrics shown within the view.

This section describes the following monitoring dashboards and views:

- Web Sites and Endpoints dashboard
- Web Site: *SelectedWebSite dashboard*
- Endpoints: *SelectedEndpoint dashboard*
- Appliance Health dashboard
- Appliance: *SelectedAppliance dashboard*
- Sniffer: *SelectedSniffer dashboard*
- Archiver: *SelectedArchiver dashboard*
- Relayer: *SelectedRelayer dashboard*
- Alarms View
- FAQts View

Web Sites and Endpoints dashboard

The Web Sites and Endpoints dashboard displays information about page and hit performance by web sites or by endpoints. To open this dashboard, from the navigation panel, under Dashboards, click **APM > Web Sites and Endpoints**.

The tiles across the top of the dashboard summarize the health of the web sites and endpoints in your environment. If a tile shows a web site or endpoint in a non-normal state, click the tile to display more information in the quick view below. The possible quick views are:

- Web Sites Quick View
- Endpoints Quick View

You can use the FAQts View and related FAQts tab to select from a list of common questions about your web sites and endpoints.

Web Sites Quick View

The Web Sites list on the left determines which view is displayed on the right. Select **All Web Sites** (default) to display the Summary - All Web Sites View, which contains web site names, end-to-end times, client times, network

delays, back-end times, and hit volumes. Select a web site to display the *SelectedWebSite View*, which displays graphical views showing how a web site page is performing, providing resources such as page performance, hit performance, errors and warnings metrics for that selection.

Table 1. Summary - All Web Sites View

Embedded Views	Purpose	Metrics
Status	Lists all the web sites and their status. For each web site, displays total time and volume metrics.	• End to End Time • Back End Time • Network Delay • Client Time • Volume
Alarms View	Displays alarms for all web sites.	
Where to go next	Click a web site. Opens the Web Site: *SelectedWebSite dashboard*.	

Table 2. *SelectedWebSite* View

Embedded Views	Purpose	Metrics
Page Performance	Monitors time spent on the server-side and the client-side for pages retrieved from this web site.	• Back End Time • Network Delay • Client Time
Hit Performance	Monitors time spent on the server-side and the client-side for hits retrieved from this web site.	• Back End Time • Network Delay • Client Time
Errors	Monitors errors occurring for pages and for hits.	See Status metrics
Warnings	Monitors warnings occurring for pages and for hits.	See Status metrics
Alarms View	Displays alarms for the selected web site.	
Where to go next	Click **Explore**. Opens the Web Site: *SelectedWebSite dashboard*.	

Endpoints Quick View

Endpoints mark the termination or completion of any session or connection. The Endpoints list on the left determines which view is displayed on the right. Select **All Endpoints** (default) to display the Summary - All Endpoints View, which contains endpoint names, end-to-end times, client times, network delays, back-end times, and hit volumes. Select a endpoint to display the *SelectedEndpoint View*, which displays graphical views showing how an endpoint page is performing, providing resources such as page performance, hit performance, errors, and warnings metrics for that selection.

Table 3. Summary - All Endpoints View

Embedded Views	Purpose	Metrics
Status	Lists all the endpoints and their status. For each endpoint, displays total time and volume metrics.	• End to End Time • Back End Time • Network Delay • Client Time • Volume
Alarms View	Displays alarms for all endpoints.	
Where to go next	Click an endpoint. Opens the Endpoints: *SelectedEndpoint dashboard*.	

Table 4. *SelectedEndpoint* View

Embedded Views	Purpose	Metrics
Page Performance	Monitors time spent on the server-side and the client-side for pages retrieved from this endpoint.	• Back End Time • Network Delay • Client Time
Hit Performance	Monitors time spent on the server-side and the client-side for hits retrieved from this endpoint.	• Back End Time • Network Delay • Client Time
Errors	Monitors errors occurring for pages and for hits.	See Status metrics
Warnings	Monitors warnings occurring for pages and for hits.	See Status metrics
Alarms View	Displays alarms for the selected endpoint.	
Where to go next	Click **Explore**. Opens the Endpoints: *SelectedEndpoint dashboard*.	

Web Site: *SelectedWebSite* dashboard

To investigate a web site, start with the Summary Tab, identify issues using the charts and metrics, and drill down to more detail by selecting the tab that matches the name of the view.

- Summary Tab—similar to the *SelectedWebSite View*, this view adds End to End Time, Capture Rate, and Volume metrics.
- Pages Tab—displays detailed page performance and page volume metrics.
- Hits Tab—displays detailed hit performance and hit volume metrics plus request and response code metrics.

Table 5. Summary Tab

Embedded Views	Purpose	Metrics
Page Performance	Monitors time spent on the server-side and the client-side for pages retrieved from this web site.	• Back End Time • Network Delay • Client Time • End to End Time • Capture Rate • Volume
Hit Performance	Monitors time spent on the server-side and the client-side for hits retrieved from this web site.	• Back End Time • Network Delay • Client Time • End to End Time • Capture Rate • Volume
Alarms View	Displays alarms for the selected web site.	
Where to go next	Click another tab to view more detail.	

Table 6. Pages Tab

Embedded Views	Purpose	Metrics
Page Performance	Investigate additional metrics that contribute to overall page performance, including redirects and incomplete downloads.	• Initial Response Time • Think Time • Hits Redirects • Hits Redirect Percent • Incomplete Download • Incomplete Download Percent
Page Volume	Investigate additional metrics that contribute to the overall page volume, including the size of headers and content for both requests and responses.	• Hits Captured • Encrypted • Encrypted Percent • Volume of Client Request Content • Volume of Client Request Headers • Volume of Server Response Content • Volume of Server Response Headers • Volume of Server Responses
Warnings and Errors	Investigate warnings and errors for the pages associated with this web site.	See Status metrics and Status expanded metrics.
Alarms View	Displays alarms for the selected web site.	
Where to go next	Investigate metrics in other tabs, create a report, or return to the Web Sites and Endpoints dashboard.	

Table 7. Hits Tab

Embedded Views	Purpose	Metrics
Hit Performance	Investigate metrics that contribute to overall hit performance.	• Back End Time • Network Delay • Client Time • Initial Response Time
Hit Volume	Investigate metrics that contribute to the overall hit volume, including the size of headers and content for both requests and responses.	• Capture Rate • Volume • Encrypted • Encrypted Percent • Volume of Client Request Content • Volume of Client Request Headers • Volume of Client Requests • Volume of Server Response Content • Volume of Server Response Headers • Volume of Server Responses
HTTP	Investigate the request methods and response codes issued for hits on this web site.	• Request Method *<command>* • Response Code *<code#>*
Warnings and Errors	Investigate warnings and errors for the hits associated with this web site.	See Status metrics and Status expanded metrics.

Table 7. Hits Tab

Embedded Views	Purpose	Metrics
Alarms View	Displays alarms for the selected web site.	
Where to go next	Investigate metrics in other tabs, create a report, or return to the Web Sites and Endpoints dashboard.	

Endpoints: *SelectedEndpoint* dashboard

To investigate an endpoint, start with the Summary Tab, identify issues using the charts and metrics, and drill down to more detail by selecting the tab that matches the name of the view.

- Summary Tab—very similar to the *SelectedEndpoint View*, this view includes additional metrics.
- Pages Tab—displays detailed page performance and page volume metrics.
- Hits Tab—displays detailed hit performance and hit volume metrics plus request and response code metrics.

Table 8. Summary Tab

Embedded Views	Purpose	Metrics
Hit Performance	Monitors time spent on the server-side and the client-side for pages retrieved from this endpoint.	• Back End Time • Network Delay • Client Time • Initial Response Time
Hit Volume	Monitors time spent on the server-side and the client-side for hits retrieved from this endpoint.	• Capture Rate • Encrypted • Encrypted Percent • Volume of Client Request Content • Volume of Client Request Headers • Volume of Client Requests • Volume of Server Response Content • Volume of Server Responses
Alarms View	Displays alarms for the selected endpoint.	
Where to go next	Click another tab to view more detail.	

Table 9. Pages Tab

Embedded Views	Purpose	Metrics
Page Performance	Investigate additional metrics that contribute to overall page performance, including redirects and incomplete downloads.	• Initial Response Time • Think Time • Hits Redirects • Hits Redirect Percent • Incomplete Download • Incomplete Download Percent
Page Volume	Investigate additional metrics that contribute to the overall page volume, including the size of headers and content for both requests and responses.	• Hits Captured • Encrypted • Encrypted Percent • Volume of Client Request Content • Volume of Client Request Headers • Volume of Client Requests • Volume of Server Response Content • Volume of Server Responses
Warnings and Errors	Investigate warnings and errors for the pages associated with this endpoint.	See Status metrics and Status expanded metrics.
Alarms View	Displays alarms for the selected endpoint.	
Where to go next	Investigate metrics in other tabs, create a report, or return to the Web Sites and Endpoints dashboard.	

Table 10. Hits Tab

Embedded Views	Purpose	Metrics
Hit Performance	Investigate metrics that contribute to overall hit performance.	• Back End Time • Network Delay • Client Time • Initial Response Time
Hit Volume	Investigate metrics that contribute to the overall hit volume, including the size of headers and content for both requests and responses.	• Capture Rate • Volume • Encrypted • Encrypted Percent • Volume of Client Request Content • Volume of Client Request Headers • Volume of Client Requests • Volume of Server Response Content • Volume of Server Response Headers • Volume of Server Responses
HTTP	Investigate the request methods and response codes issued for hits on this endpoint.	• Request Method <*command*> • Response Code <*code#*>
Warnings and Errors	Investigate warnings and errors for the hits associated with this endpoint.	See Status metrics and Status expanded metrics.

Table 10. Hits Tab

Embedded Views	Purpose	Metrics
Alarms View	Displays alarms for the selected endpoint.	
Where to go next	Investigate metrics in other tabs, create a report, or return to the Web Sites and Endpoints dashboard.	

Appliance Health dashboard

You can use the Appliance Health dashboard to review the performance of all Foglight APM Appliances in your installation as well as the software components hosted on the appliances. To access this dashboard, open the navigation panel, under Dashboards, click **APM > Support > Appliance Health**.

NOTE: For an example of how to use this dashboard, see the *Foglight APM Administration and Configuration Guide*.

The tiles across the top of the dashboard summarize the health of the appliances, Sniffers, Archivers, and Relayers. If a tile shows an appliance or component in a non-normal state, click the tile to display more information in the quick view below. The possible quick views are:

- Appliances Quick View
- Sniffers Quick View
- Archivers Quick View
- Relayers Quick View

You can use the FAQts View and related FAQts tab to select from a list of common questions about your appliances and software components.

Appliances Quick View

The Appliances list on the left determines which view is displayed on the right. Select **All Appliances** (default) to display the Summary - All Appliances View, which contains CPU usage metrics and process counts for each of your appliances in a table format. Select an appliance to display the *SelectedAppliance View*, which displays graphical views showing how the appliance uses resources, such as CPU and memory.

Table 11. Summary - All Appliances View

Embedded Views	Purpose	Metrics
Status	Lists all the appliances and their status. For each appliance, aggregates usage and process counts for the CPUs on the appliance.	• CPU - I/O Wait Percent • CPU - Idle Percent • CPU - System Percent • CPU - User Percent • Processes Running • Processes Blocked
Alarms View	Displays alarms for all appliances. **NOTE:** This does not include cleared alarms and is not affected by the chosen time range.	
Where to go next	Click an appliance. Opens the Appliance: *SelectedAppliance dashboard*.	

Table 12. *SelectedAppliance* View

Embedded Views	Purpose	Metrics
CPU Utilization	Monitors how the appliance is using its CPUs over time as a percentage of available CPU resources.	• CPU - I/O Wait Percent • CPU - Idle Percent • CPU - System Percent • CPU - User Percent
Memory Utilization	Monitors how the appliance is using memory resources over time.	• Memory - Free Memory • Memory - Buffer Memory Used • Memory - Cache Memory Used • Memory - Swap Memory Used
Filesystem Usage	Monitors how the appliance is using the various file systems over time.	• Filesystem - Archiver Used Percent • Filesystem - Foglight Used Percent • Filesystem - Quest Used Percent • Filesystem - Root Used Percent • Filesystem - Tmp Used Percent • Filesystem - Var Used Percent
Appliance Details	Displays details about the appliance, such as the software components it hosts and the number of CPUs available.	
Alarms View	Displays alarms for the selected appliance.	
Where to go next	Click **Explore**. Opens the Appliance: *SelectedAppliance dashboard*.	

Sniffers Quick View

The Sniffers list on the left determines which view is displayed on the right. Select **All Sniffers** to display the Summary - All Sniffers View. Select a Sniffer to display the *SelectedSniffer View*.

Table 13. Summary - All Sniffers View

Embedded Views	Purpose	Metrics
Status	Lists all the Sniffers and their status. Summarizes packet capture metrics and error percentages for each Sniffer.	• Packets - Capture Rate • Hits - Relayed • Packets - Captured • Packets - Dropped • TCP Segments - Client Missing Percent • TCP Segments - Server Missing Percent • SSL - Error Percent
Alarms View	Displays alarms for all Sniffers. **NOTE:** This does not include cleared alarms and is not affected by the chosen time range.	
Where to go next	Click a Sniffer. Opens the Sniffer: *SelectedSniffer dashboard*.	

Table 14. *SelectedSniffer* View

Embedded Views	Purpose	Metrics
Capture Rate	Monitors the rate (packets per second) at which the Sniffer captures packets.	• Packets - Capture Rate
Capture Volume	Monitors how many packets are captured and whether they are processed or dropped.	• Packets - Captured • Packets - Dropped • Packets - Processed
Capture Quality	Monitors problems in data capture.	• Packets - Drop Percent • TCP Segments - Client Missing Percent • TCP Segments - Server Missing Percent • SSL - Error Percent
SSL Volume	Monitors how many SSL connections are started and whether they are dropped or errors are detected.	• SSL - Connections Started • SSL - Connections Dropped • SSL - Connection Error
Alarms View	Displays alarms for the selected Sniffer.	
Where to go next	Click **Explore**. Opens the Sniffer: *SelectedSniffer dashboard*.	

Archivers Quick View

The Archivers list on the left determines which view is displayed on the right. Select **All Archivers** to display the Summary - All Archivers View. Select an Archiver to display the *SelectedArchiver View*.

Table 15. Summary - All Archivers View

Embedded Views	Purpose	Metrics
Status	Lists all the Archivers and their status. Summarizes hit capture metrics and memory use for each Archiver.	• Hits - Capture Rate • Capture Volume • Hits - Discard Rate • Hits - Drop Rate • Shared Memory Used • JVM - Free Memory
Alarms View	Displays alarms for all Archivers. **NOTE:** This does not include cleared alarms and is not affected by the chosen time range.	
Where to go next	Click an Archiver. Opens the Archiver: *SelectedArchiver dashboard*.	

Table 16. *SelectedArchiver* View

Embedded Views	Purpose	Metrics
Hits - Capture Rate	Monitors the rate (hits per second) at which the Archiver receives hits from the Sniffer.	• Hits - Capture Rate
Hits - Discarded and Dropped	Monitors the number of hits that have been discarded before or after processing.	• Hits - Discarded • Hits - Dropped
Shared Memory Used	Monitors shared memory.	• Shared Memory Size • Shared Memory Used

Table 16. *SelectedArchiver* View

Embedded Views	Purpose	Metrics
JVM Memory Usage	Monitors how the Archiver uses its Java virtual machine.	• JVM - Total Memory • JVM - Free Memory • JVM - Used Memory
Alarms View	Displays alarms for the selected Archiver.	
Where to go next	Click **Explore**. Opens the Archiver: *SelectedArchiver dashboard*.	

Relayers Quick View

The Relayers list on the left determines which view is displayed on the right. Select **All Relayers** to display the Summary - All Relayers View. Select a Relayer to display the *SelectedRelayer View*.

Table 17. Summary - All Relayers View

Embedded Views	Purpose	Metrics
Status	Lists all the Relayers and their status. For each Relayer, states how much hit content was not delivered to Archivers and summarizes message statistics and queue status.	• Relayer - Hit Content Dropped Due to Exceeding Max Size • Relayer - Discarded Messages • Relayer - Open Messages • Relayer - Messages Transferred
Alarms View	Displays alarms for all Relayers. **NOTE:** This does not include cleared alarms and is not affected by the chosen time range.	
Where to go next	Click a Relayer. Opens the Relayer: *SelectedRelayer dashboard*.	

Table 18. *SelectedRelayer* View

Embedded Views	Purpose	Metrics
Hit Content Dropped	Monitors how much hit content the Relayer drops when sending hits to Archivers.	• Relayer - Hit Content Dropped Due to Exceeding Max Size
Messages Discarded	Monitors how many messages the Relayer does not deliver to Archivers.	• Relayer - Discarded Messages
Queues	Monitors the Relayer's queue for each Archiver.	• Relayer - Queue Memory Used Percent • Relayer - Captured Hits • Relayer - Hits Capture Rate • Relayer - Discarded Messages • Relayer - Queued Messages • Relayer - Socket Error Uploading Capture Data
Alarms View	Displays alarms for the selected Relayer.	
Where to go next	Click **Explore**. Opens the Relayer: *SelectedRelayer dashboard*.	

Appliance: *SelectedAppliance* dashboard

To investigate an appliance, start with the Summary Tab, identify issues using the charts and metrics, and drill down to more detail by selecting the tab that matches the name of the view.

- Summary Tab—very similar to the *SelectedAppliance View*, this dashboard breaks out a few more metrics into their own charts, such as CPU utilization by core and process activity.
- CPU Tab—highlights heavily loaded cores and offers CPU-based metrics by core.
- Memory Tab—displays memory use over time in customizable charts.
- Storage Tab—displays file system use over time in customizable charts.
- Processes Tab—displays active and blocked process in customizable charts.
- NICs Tab—displays information about the appliance's control NIC (eth0), which is used to communicate with the Management Server, and the appliance's auxiliary NIC (eth1), which is used for Sniffer to Archiver communication.

Table 19. Summary Tab

Embedded Views	Purpose	Metrics
CPU	Monitors aggregate metrics for all CPUs as well as individual CPU behavior.	• CPU - I/O Wait Percent • CPU - Idle Percent • CPU - System Percent • CPU - User Percent
Memory	Monitors how the selected appliance uses buffer, caches, and swap memory.	• Memory - Free Memory • Memory - Buffer Memory Used • Memory - Cache Memory Used • Memory - Swap Memory Used
Storage	Monitors how the appliance is using the various file systems over time.	• Filesystem - Archiver Used Percent • Filesystem - Foglight Used Percent • Filesystem - Quest Used Percent • Filesystem - Root Used Percent • Filesystem - Tmp Used Percent • Filesystem - Var Used Percent
Processes	Monitors processes running on the appliance.	• Processes Running • Processes Blocked
Appliance Information	Displays details about the appliance, such as the software components it hosts and the number of CPUs available.	
Where to go next	Click another tab to view more detail.	

Table 20. CPU Tab

Embedded Views	Purpose	Metrics
Aggregate CPU	Investigate aggregate metrics for all CPUs.	• CPU - I/O Wait Percent • CPU - Idle Percent • CPU - System Percent • CPU - User Percent
CPU Core Utilization	Uses color and speed to highlight heavily loaded cores.	*Calculated*
CPU Core Details	Select a heavily loaded core from the list to view metrics for that CPU.	Same metrics as Aggregate CPU, but for the selected CPU.
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Table 21. Memory Tab

Embedded Views	Purpose	Metrics
Memory Utilization	Investigate how the appliance uses all memory over time, including the peak usage and current usage.	• Memory Utilization
Buffer Memory	Investigate how the appliance uses buffer memory over time, including the peak usage and current usage.	• Memory - Buffer Memory Used
Cache Memory	Investigate how the appliance uses cache memory over time, including the peak usage and current usage.	• Memory - Cache Memory Used
Free Memory	Investigate how the appliance frees memory over time, including the peak amount of free memory and current amount.	• Memory - Free Memory
Swap Memory	Investigate how the appliance uses swap memory over time, including the peak usage and current usage.	• Memory - Swap Memory Used
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Table 22. Storage Tab

Embedded Views	Purpose	Metrics
Root	Investigate how the appliance uses Root storage. Root contains operating system files.	• Filesystem - Root Used Percent • Filesystem - Root Used
Foglight	Investigate how the appliance uses Foglight storage. Foglight contains all Foglight management server software components and associated files.	• Filesystem - Foglight Used Percent • Filesystem - Foglight Used
Quest	Investigate how the appliance uses Quest storage. Quest contains all software components other than Foglight management server.	• Filesystem - Quest Used Percent • Filesystem - Quest Used

Table 22. Storage Tab

Embedded Views	Purpose	Metrics
Tmp	Investigate how the appliance uses Tmp storage. Tmp is used for temporary files.	• Filesystem - Tmp Used Percent • Filesystem - Tmp Used
Archiver	Investigate how the appliance uses Archiver storage. Archiver contains the Archiver database.	• Filesystem - Archiver Used Percent • Filesystem - Archiver Used
Var	Investigate how the appliance uses Var storage. Var contains logging and system files.	• Filesystem - Var Used Percent • Filesystem - Var Used
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Table 23. Processes Tab

Embedded Views	Purpose	Metrics
Processes Running	Investigate the number of active processes running at any given time. Includes peak and current numbers.	• Processes Running
Processes Blocked	Investigate the number of blocked processes at any given time. Includes peak and current numbers.	• Processes Blocked
Alarms View	Displays alarms.	
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Table 24. NICs Tab

Embedded Views	Purpose	Metrics
Control NIC	Investigate the performance of the NIC used to communicate with the Management Server.	• Receive Metrics • Transmit Metrics
Auxiliary NIC	Investigate the performance of the NIC used for Sniffer to Archiver communication. Also called the capture subnet.	• Receive Metrics • Transmit Metrics
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Sniffer: *SelectedSniffer* dashboard

To investigate a Sniffer, start with the Summary Tab, identify issues using the charts and metrics, and drill down to more detail by selecting the tab that matches the name of the view.

- Summary Tab—more detailed than the *SelectedSniffer View*, this dashboard summarizes the key performance metrics for the selected Sniffer.
- Capture Tab—displays how the Sniffer handles packets and TCP traffic.
- Hit Processing Tab—displays how the Sniffer handles hits.
- Traffic Tab—displays how the Sniffer handles TCP and SSL connections.
- SSL Decryption Tab—displays SSL connection and traffic data in a customizable table.

- Sniffer Health Tab—displays how the Sniffer appliance handles memory and communicates with siteminder servers.
- Monitoring NICs Tab—displays warnings, errors, and appliance restarts.

Table 25. Summary Tab

Embedded Views	Purpose	Metrics
Capture Metrics	Monitors how the Sniffer handles packets.	• Packets - Capture Rate • Packets - Captured • Packets - Processed • Packets - Dropped • Packets - Drop Percent • TCP Segments - Client • TCP Segments - Client Missing • TCP Segments - Server • TCP Segments - Server Missing
Hit Processing	Monitors how the Sniffer handles hits.	• Hits - Relayed • Hits - Discarded • Hits - Excluded • Hits - Resets on Client • Hits - Resets on Server • Hits - Timeout on Client • Hits - Timeout on Server • Hits - Chunked Transfer Error • Hits - Discarded For Malformed HTTP • Hits - Discarded for Packet Drops
SSL Volume	Monitors how the Sniffer handles SSL connections.	• SSL - Connections Started • SSL - Connections Dropped • SSL - Connection Error
Sniffer Health	Monitor warnings, errors, and Sniffer Appliance restarts.	• Memory Consumption Warning • Restarts • Siteminder - Lookup Failures • Siteminder - Lookup Failures Percent • Packet Errors
TCP Volume	Monitors how the Sniffer handles TCP connections.	• TCP - Connections Started • TCP - Connections Dropped • TCP - Protocol Error
Alarms View	Displays alarms.	
Where to go next	Click another tab to view more detail.	

Embedded Views	Purpose	Metrics
Capture Rate	View the Sniffer's capture rate in a customizable chart.	• Packets - Capture Rate
Packet Capture	Investigate how the Sniffer performs while capturing live web traffic.	• Packets - Captured • Packets - Processed • Packets - Dropped

Embedded Views	Purpose	Metrics
Traffic Quality	Investigate how the Sniffer performs when handling TCP traffic.	• TCP Segments - Client • TCP Segments - Client Missing • TCP Segments - Server • TCP Segments - Server Missing • TCP Segments - Client Missing Percent • TCP Segments - Server Missing Percent
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Table 26. Hit Processing Tab

Embedded Views	Purpose	Metrics
Hits	Investigate what the Sniffer does with hits.	• Hits - Relayed • Hits - Discarded • Hits - Excluded
Resets	Investigate the number of hits for which the Sniffer did not capture full content due to a reset request from the client or server.	• Hits - Resets on Client • Hits - Resets on Server
Bytes Relayed	Investigate the number of bytes relayed from the Sniffer to the Archiver.	• Hits - Relayed Bytes • Hits - Relayed Request Bytes • Hits - Relayed Response Bytes
Timeouts	Investigate the number hits for which the Sniffer did not capture full content due to timeouts.	• Hits - Timeout on Client • Hits - Timeout on Server
Errors	Investigate why the Sniffer discarded hits.	• Hits - Chunked Transfer Error • Hits - Discarded For Malformed HTTP • Hits - Discarded for Packet Drops
Truncated Hits	Investigate the number hits the Sniffer truncated.	• Hits - Truncated Requests • Hits - Truncated Response
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Table 27. Traffic Tab

Embedded Views	Purpose	Metrics
SSL	Investigate how the selected Sniffer performs when handling SSL connections.	• SSL - Connections Started • SSL - Connections Dropped • SSL - Connection Error • SSL - Error Percent
TCP	Investigate how the selected Sniffer performs when handling TCP connections.	• TCP - Connections Started • TCP - Connections Dropped • TCP - Protocol Error

Table 27. Traffic Tab

Embedded Views	Purpose	Metrics
SOAP	Investigate how the selected Sniffer performs when handling SOAP messages.	• SOAP - Messages • SOAP - Errors • SOAP - Parsing Errors • SOAP - Errors Percent
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Table 28. SSL Decryption Tab

Embedded Views	Purpose	Metrics
Table	Investigate how the selected Sniffer performs when handling SSL connections and decrypting SSL traffic.	• SSL - TCP Client Segments • SSL - Missing TCP Client Segments • SSL - Missing TCP Client Segment Percent • SSL - TCP Server Segments • SSL - Missing TCP Server Segments • SSL - Missing TCP Server Segment Percent • SSL - Connections Started • SSL - Connections Dropped • SSL - Connection Error • SSL - Error Percent
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Table 29. Sniffer Health Tab

Embedded Views	Purpose	Metrics
Sniffer Metrics	Investigate memory problems that may cause the Sniffer Appliance to restart.	• Memory Consumption Warning • Restarts
SiteMinder	Investigate the selected Sniffer's communication with SiteMinder servers.	• Siteminder - Lookups • Siteminder - Lookup Failures
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Table 30. Monitoring NICs Tab

Embedded Views	Purpose	Metrics
Packet Drop Percentage	Uses color and speed to highlight heavily loaded NICs. An overload monitoring NIC drops a significant number of packets as its buffer fills.	• Packets - Captured • Packets - Processed • Packets - Dropped
Details	Select a heavily loaded NIC from the list to view Sniffer metrics for the NIC.	• Packets - Captured • Packets - Capture Rate • Packets - Processed • Packets - Dropped • Packets - Drop Percent • Packets - Error
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Archiver: *SelectedArchiver* dashboard

To investigate a Sniffer, start with the Summary Tab, identify issues using the charts and metrics, and drill down to more detail by selecting the tab that matches the name of the view.

- Summary Tab—more detailed than the *SelectedArchiver View*, this dashboard summarizes the key performance metrics for the selected Archiver.
- Capture Tab—visualizes how capture data moves from the Relayer to the Archiver.
- Hit Processing Tab—displays hit and page processing metrics and error metrics.
- Database Tab—displays database usage metrics.
- Traffic Analysis Tab—displays metrics related to sessions, sequences, and custom fields.
- Search Analysis Tab—displays search request by details.
- Script Metrics Tab—displays script execution and error count.
- Stomp Metrics Tab—displays stomp connections count and metrics.

Table 31. Summary Tab

Embedded Views	Purpose	Metrics
Capture Metrics	Monitors how much and how quickly the Archiver captures data, and how it uses shared memory.	• Hits - Capture Rate • Capture Volume • Capture Volume - Instrumentation • Shared Memory Size • Shared Memory Used
Hit Processing	Monitors how the Archiver processes incoming hits, including how many hits are discard, dropped, or caused errors or warnings.	• Hits - Captured • Hits - Discarded • Hits - Dropped • Hits - Client Connection Reset • Hits - Client Timeout • Hits - Server Connection Reset • Hits - Server Timeout • Hits - Captured • Pages - Captured • Hits - Chunked Transfer Error • Hits - Excessive Clock Skew • Hits - HTML Parse Error • Traffic Analysis - Rules Error • Batch Error • Hits - Analyzer Error • Hits - Analyzer Warning
Database	Monitors activity for the integrated Archiver database, including how space is allocated and shard status.	• Database - Size • Database - Hits • Database - Sessions • Database - Shards • Database - Aspects • Shards - Closing • Shards - Waiting to Copy • Shards - Copying
Traffic Analysis	Monitors how the Archiver processes the incoming hits.	• Traffic Analysis - Sessions Active • Traffic Analysis - Sessions Completed • Traffic Analysis - Session Timeout • Traffic Analysis - Sequence Hit Analyzers Active • Traffic Analysis - Sequences Completed • Clock Skew • Configuration Available • RAM Filesystem Size • Refresh Metric Time
Alarms View	Displays alarms.	
Where to go next	Click another tab to view more detail.	

Table 32. Capture Tab

Embedded Views	Purpose	Metrics
Hits Capture Rate	Monitors the volume of hits passed to the Archiver. The spinner uses color and speed to highlight low or high volumes.	• Capture Rate • Capture Volume • Capture Volume - Instrumentation
Hits (arrow)	Monitors the number of hits passed to the Archiver. The arrow uses color and speed to highlight low or high volumes.	• Hits - Captured
Relayer Queue Metrics	Monitors how the Relayer uses its queue memory.	• Hits - Captured • Relayer - Queue Memory Used Percent • Relayer - Discarded Pages Due to Queue Full
Shared Memory	Monitors how the Archiver uses shared memory.	• Shared Memory Size • Shared Memory Used
Write Buffer Failed	Monitors how often write buffers are lost due to an internal error.	• Write Buffer Failed
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard. For more details on Relayers, return to the Appliance Dashboard and click the **Relayers** tile.	

Table 33. Hit Processing Tab

Embedded Views	Purpose	Metrics
Hit Processing	Investigate what the Archiver does with hits.	• Hits - Captured • Hits - Discarded • Hits - Dropped
Page Processing	Investigate what the Archiver does with pages.	• Pages - Captured • Pages - Discarded
Hit Breakdown	Investigate the volume of HTML hits and instrumentation hits. Also review hit analyzer error and warning counts.	• Hits - HTML • Hits - Instrumentation • Hits - Analyzer Error • Hits - Analyzer Warning

Table 33. Hit Processing Tab

Embedded Views	Purpose	Metrics
Metrics and Errors	Investigate warnings, errors, corrupted data, and content issues experienced by the Archiver.	• Hits - Client Connection Reset • Hits - Client Timeout • Hits - Server Connection Reset • Hits - Server Timeout • Hits - Truncated Request • Hits - Truncated Response • Hits - Chunked Transfer Error • Hits - Excessive Clock Skew • Hits - HTML Parse Error • Traffic Analysis - Rules Error • Hits - Unknown Exception • Hits - Missing Property • Hits - Missing Response • Hits - Missing Session Identifier
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Table 34. Database Tab

Embedded Views	Purpose	Metrics
Database Size	Investigate how the database size changes over time.	• Database - Size
Capture Database	Monitors how space is allocated in the Archiver database.	• Database - Size • Database - Hits • Database - Sessions • Database - Shards • Database - Aspects
Database Shards	Investigate how the number of shards in the database changes over time.	• Database - Shards
Database Metrics	Investigate how database space allocations change over time.	• Database - Hits • Database - Sessions • Database - Aspects
MySQL Index Database	Investigate how the embedded MySQL database itself is performing.	• MySQL - Key Read Requests • MySQL - Key Write Requests • MySQL - Key Reads • MySQL - Key Writes • MySQL - Key Blocks Max Used • MySQL - Key Blocks Not Flushed • MySQL - Key Blocks Unused
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Table 35. Traffic Analysis Tab

Embedded Views	Purpose	Metrics
Sessions Active	Investigate how many sessions are active over time.	• Traffic Analysis - Sessions Active
Sessions Completed	Investigate how many sessions are completed over time.	• Traffic Analysis - Sessions Completed
Sequences Active	Investigate how many sequences are active over time.	• Traffic Analysis - Sequences Active
Sequences Completed	Investigate how many sequences are completed over time.	• Traffic Analysis - Sequences Completed
Session Timeout	Investigate how many sessions timeout.	• Traffic Analysis - Session Timeout
Traffic Analysis Metrics	Investigate analyzer activity, including how many custom fields are actively being updated by analyzers.	• Traffic Analysis - Custom Fields Active • Traffic Analysis - Session Hit Analyzers Active • Traffic Analysis - Sequence Hit Analyzers Active
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Table 36. Search Analysis Tab

Embedded Views	Purpose	Metrics
Properties	Investigate the number of search requests by various details.	For an alphabetical list of all search metrics, see Archiver Search metrics.
Hits	Investigate the number of search requests by hit detail.	
Pages	Investigate the number of search requests by page detail.	
Sessions	Investigate the number of search requests by session detail.	
Other	Investigate the number of other kinds of search requests.	
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Table 37. Script Metrics Tab

Embedded Views	Purpose	Metrics
Script Metrics	Investigate how often scripts execute.	
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Table 38. Stomp Metrics Tab

Embedded Views	Purpose	Metrics
Stomp Connections Active	Investigate how many connections are active.	• stompConnectionsActive
Stomp Connection	Investigate the details for each selected Stomp Connection.	• stompMessagesCaptured • stompConnectionErrors • stompBytesSent • stompMessagesDiscardedExceedsMaxSize • stompMessagesDiscardedQueueFull • stompMessagesDiscardedUploadError
Where to go next	Investigate metrics in other tabs, create a report, or return to the Appliance Health Dashboard.	

Relayer: *SelectedRelayer* dashboard

To investigate a Relayer, start with the Summary Tab, identify issues using the charts and metrics, and drill down to more detail by selecting the tab that matches the name of the view.

- Summary Tab—more detailed than the *SelectedRelayer View*, this dashboard summarizes the key performance metrics for the selected Archiver.
- Queues Tab—displays hit and page processing metrics and error metrics.

Table 39. Summary Tab

Embedded Views	Purpose	Metrics
Message Processing	Monitors how the Relayer performs when handling messages.	• Relayer - Discarded Messages • Relayer - Open Messages • Relayer - Messages Transferred • Relayer - Discarded Messages Due to Exceeding Max Size • Relayer - Discarded Messages Due to Memory Full
Where to go next	Click another tab to view more detail.	

Table 40. Queues Tab

Embedded Views	Purpose	Metrics
Queue - Memory Used	Uses color and speed to highlight when the Relayer queue memory is very full.	• Relayer - Queue Memory Used Percent
Queue Details	Investigate how the Relayer manages its queue and the hits and messages within it.	• Relayer - Captured Hits • Relayer - Discarded Pages Due to Queue Full • Relayer - Queue Memory Used Percent • Relayer - Hits Capture Rate • Relayer - Queued Messages
Where to go next	Click another tab to view more detail.	

Alarms View

The Alarms view displays a contextual list of alarms, which may help you identify and diagnose issues with your environment.

FAQts View

The FAQts view displays a frequently asked question related to the data you are currently viewing in the quick view. In the FAQts view, you can:

- Display the answer to the current question by clicking **Show Me**. The data that answers your question appears in a customizable table or chart.
- Scroll through the questions using the **Previous** and **Next** arrows.
- Display a list of questions by clicking the **List** icon. Click a question to display the answer.

Alternatively, you can click the **FAQts** tab at the top of the dashboard. In the FAQts tab, you can:

- View all questions or questions by category.
- Search for questions.
- Mark questions as favorites by clicking the **Favorites** icon.
- Create a report by clicking the **Create Report** icon.

2

Health metrics

Health metrics report on the performance of your appliances, Sniffers, Archivers, and Relayers.

This section describes the following monitoring dashboards:

- Appliance health metrics
- Sniffer health metrics
- Archiver health metrics
- Relayer health metrics

Appliance health metrics

You can find a selection of these metrics displayed in the following dashboards:

- **APM > Support > Appliance Health >** Appliances Quick View
- **APM > Support > Appliance Health >** Appliances Quick View > Appliance: *SelectedAppliance dashboard*

Table 41. Appliance health metrics

Appliance Health	Metric Description	Topology Object Name
CPU - I/O Wait Percent	Percentage of CPU time spent waiting for disk operations, averaged over a five-second period when the metric was last updated. This value can spike up to 70%+ for very short periods of time (one observation), but should usually be 10% or less. A consistent high value indicates a drive failure or serious software problem.	`cpuIOWaitPercent`
CPU - Idle Percent	Percentage of CPU time spent idle, averaged over a five-second period when the metric was last updated. A consistent low value indicates the appliance is heavily loaded.	`cpuIdlePercent`
CPU - System Percent	Percentage of CPU time spent running system processes, averaged over a five-second period when the metric was last updated. This value should normally be 10% or less.	`cpuSystemPercent`
CPU - User Percent	Percentage of CPU time spent running user processes, averaged over a five-second period when the metric was last updated. These user processes include the Archiver and Server JVM processes. A consistent high value indicates the appliance is heavily loaded.	`cpuUserPercent`
Filesystem - Archiver Used	Amount of space used on the Archiver filesystem.	`filesystemArchiver Used`
Filesystem - Archiver Used Percent	Percentage of space used on the Archiver filesystem.	`filesystemArchiver UsedPercent`
Filesystem - Foglight Used	Amount of space used on the */foglight* filesystem.	`filesystemFoglight Used`

Table 41. Appliance health metrics

Appliance Health	Metric Description	Topology Object Name
Filesystem - Foglight Used Percent	Percentage of space used on the */foglight* filesystem.	`filesystemFoglightUsedPercent`
Filesystem - Quest Used	Amount of space used on the */quest* filesystem.	`filesystemQuestUsed`
Filesystem - Quest Used Percent	Percentage of space used on the */quest* filesystem.	`filesystemQuestUsedPercent`
Filesystem - Root Used	Amount of space used on the “/” filesystem.	`filesystemRootUsed`
Filesystem - Root Used Percent	Percentage of space used on the “/” filesystem.	`filesystemRootUsedPercent`
Filesystem - Tmp Used	Amount of space used on the */tmp* filesystem.	`filesystemTmpUsed`
Filesystem - Tmp Used Percent	Percentage of space used on the */tmp* filesystem.	`filesystemTmpUsedPercent`
Filesystem - Var Used	Amount of space used on the */var* filesystem.	`filesystemVarUsed`
Filesystem - Var Used Percent	Percentage of space used on the */var* filesystem.	`filesystemVarUsedPercent`
Memory - Buffer Memory Used	Amount of memory used by the appliance operating system to buffer disk writes. This value is typically low even in high-volume installations. Archivers rely upon write buffering at the RAID controller, so there should be almost no write buffering in the operating system.	`memoryUsedBuffer`
Memory - Cache Memory Used	Amount of memory used by the appliance operating system to buffer disk-read caching. The operating system uses free memory as a disk cache to improve I/O performance. A significant amount of memory (50% or more) may be allocated to disk caching, or this value may be as low as one-eighth of total memory.	`memoryUsedCache`
Memory - Free Memory	Amount of appliance operating system memory that is currently not used for any purpose. It is normal to have almost no free memory, since the operating system tends to use any available memory either for disk read caching or write buffering.	`memoryFree`
Memory - Swap Memory Used	Amount of temporary swap memory used by the appliance operating system. This value should be zero for Archiver Appliances, which are configured to never swap. If usage on any appliance exceeds 90%, a problem may be developing with the appliance. Contact Quest Technical Support.	`memoryUsedSwap`
Processes Blocked	Number of processes that are waiting to run, averaged over a five-second period when the metric was last updated. This value may be consistently zero for lightly loaded systems. Under heavy load, this value oscillates normally as I/O operations are synchronized. A consistent high value may indicate a threading problem.	`processesBlocked`
Processes Running	Number of processes currently running, averaged over a five-second period when the metric was last updated. This value may be small for lightly loaded systems. Under heavy load, this value oscillates normally as I/O operations are synchronized.	`processesRunning`

Sniffer health metrics

Sniffer health metrics are grouped into Sniffer General Health Metrics, Sniffer TCP Metrics, and Sniffer SSL Metrics.

You can find a selection of these metrics displayed in the following dashboards:

- **APM > Support > Appliance Health >** Sniffers Quick View
- **APM > Support > Appliance Health >** Sniffers Quick View > Sniffer: *SelectedSniffer dashboard*

Sniffer General Health Metrics

A hit is anything returned as a result of HTTP GET or POST request.

Table 42. Sniffer General heath metrics

Sniffer General	Metric Description	Topology Object Name
Hits - Chunked Transfer Error	Number of hits that did not receive the entire request or response content because of a dechunking error. This metric is incremented if the hit response contains chunked content and one of the chunks cannot be parsed due to a formatting or syntax error in the chunking header. This metric is also incremented if the chunked content is not properly terminated.	`hitsChunkedTransfer Error`
Hits - Discarded	Number of hits discarded within the time range. Hits may be discarded for the following reasons: • The hit contained packets that were dropped. • A hit analyzer excluded the hit. • The TCP connection timed out before the hit was complete. • The URL was malformed.	`hitsDiscarded`
Hits - Discarded for API Errors	Number of times the Sniffer calls any of the Relayer's API (except for `newMessage`) and the call fails. A call fails when the shared memory segment is full.	`hitsDiscardedFor ApiErrors`
Hits - Discarded For Malformed HTTP	Number of hits that were not sent to the Archiver because the Sniffer did not recognize the application data as being a valid HTTP request start. For example, if an initial block of bytes is sent from the client to the server on a HTTP connection, but the Sniffer does not detect a HTTP request line in those first bytes, then the hit is discarded and this metric is incremented.	`hitsDiscardedFor MalformedHTTP`
Hits - Discarded for New Message Errors	Number of times the Sniffer calls the Relayer's `newMessage` api and the call fails. The Sniffer calls the API at the start of each hit or page. A call fails when the shared memory segment is full.	`hitsDiscardedFor NewMessageErrors`
Hits - Discarded for Packet Drops	Number of hits that were not sent to the Archiver because one or more packets were not seen during the hit capture, resulting in incomplete content.	`hitsDiscardedFor PacketDrops`
Hits - Excluded	Not implemented.	`hitsExcluded`
Hits - Relayed	Number of hits transmitted to Archivers in the time range.	`hitsRelayed`
Hits - Relayed Bytes	Number of bytes transmitted to the Archiver in the time range.	`hitsBytesRelayed`
Hits - Relayed Request Bytes	Number of bytes for requests transmitted to the Archiver in the time range.	`hitsBytesRelayed Request`

Table 42. Sniffer General heath metrics

Sniffer General	Metric Description	Topology Object Name
Hits - Relayed Response Bytes	Number of bytes for responses transmitted to the Archiver in the time range.	`hitsBytesRelayed Response`
Hits - Resets on Client	Number of hits that did not receive the entire content of the client's request because a TCP reset was sent by the client.	`hitsResetsClient`
Hits - Resets on Server	Number of hits that did not receive the entire content of the server's response because a TCP reset was sent by the server.	`hitsResetsServer`
Hits - Timeout on Client	Number of hits that did not receive the entire content of the client's request before a timeout occurred.	`hitsTimeoutClient`
Hits - Timeout on Server	Number of hits that did not receive the entire content of the server's response before a timeout occurred.	`hitsTimeoutServer`
Hits - Truncated Requests	Number of hits where the request content was truncated to the maximum request length configuration.	`hitsTruncatedRequests`
Hits - Truncated Response	Number of relayed hits where the response content was truncated to the maximum response length configuration.	`hitsTruncated Responses`
Memory Consumption Warning	Number of times a Sniffer has exceeded the designated limit of physical memory or swap space consumption. Generally, this threshold is set at 85% for physical memory and 20% for swap space. These limits can be exceeded when a Sniffer has been configured to monitor an amount of network traffic that exceeds its capacity.	`memoryConsumption Warning`
Packets - Captured	Number of packets captured in the time range.The Sniffer processes only the packets that qualify based on the configuration settings for the Sniffer, monitored IP addresses, and monitored port.	`packetsCaptured`
Packets - Capture Rate	= Packets - Captured / second Rate at which the Sniffer captures packets.	`packetsCaptureRate`
Packets - Drop Percent	= Packets - Dropped / Packets - Captured Percentage of packets dropped by the Sniffer. A value of zero means that no packets are being dropped. Typically, this metric is either zero or a very small percentage. A significant value indicates that the Sniffer may be overloaded.	`packetsDropPercent`
Packets - Dropped	Number of packets dropped in the time range. A large number of dropped packets may indicate that the Sniffer is overloaded.	`packetsDropped`
Packets - Error	Number of packets all monitoring ports failed to receive due to low-level link issues. Normally, this number should be at or near zero. A significant number of errors could indicate a problem with the network infrastructure, such as network taps that are feeding the port.	`packetsError`
Packets - Processed	Number of packets processed in the time range. The Sniffer processes only the packets that qualify based on the configuration settings for the Sniffer, monitored IP addresses, and monitored port.	`packetsProcessed`

Table 42. Sniffer General heath metrics

Sniffer General	Metric Description	Topology Object Name
Restarts	Number of times the Sniffer stopped and restarted. A Sniffer may restart for the following reasons: • Exceeded its allowed level of memory use. • Exceeded its memory threshold. • Encountered an unrecoverable error. • Encountered an incorrect thread count. A restart typically results in a five-minute gap in captured data.	`restarts`
Siteminder - Lookups	Number of times the Sniffer queries SiteMinder to identify user sessions in the monitored traffic.	`siteminderLookups`
Siteminder - Lookup Failures	Number of communication errors that occurred between the Sniffer and one or more SiteMinder Policy Servers.	`siteminderLookup Failures`
Siteminder - Lookup Failures Percent	Percentage of times that requests to SiteMinder to identify user sessions failed.	`siteminderLookup FailuresPercent`
SOAP - Errors	Number of SOAP messages that caused errors during processing. Errors can occur for the following reasons: • The SOAP message is corrupted. • An XML parsing error occurred. This metric includes all of the errors also counted by the *SOAP - Parsing Errors* metric.	`soapErrors`
SOAP - Errors Percent	= SOAP - Errors / SOAP - Messages Percentage of SOAP messages with errors.	`soapErrorPercent`
SOAP - Messages	Number of SOAP messages processed in the time range.	`soapMessages`
SOAP - Parsing Errors	Number of SOAP messages that cannot be parsed by the Sniffer due to badly formed XML.	`soapXmlParsingErrors`

Sniffer TCP Metrics

Table 43. Sniffer TCP metrics

Sniffer TCP	Metric Description	Topology Object Name
TCP - Connections Dropped	Number of TCP connections that the Sniffer has discarded. Whenever a Sniffer's appliance is heavily overloaded, the Sniffer begins discarding TCP connections to avoid running out of memory. Under normal conditions, this metric should always be zero. If this metric is greater than zero, it indicates that some data loss is likely occurring.	`tcpConnectionsDropped`
TCP - Connections Started	Number of TCP connections that have been initiated in the time range.	`tcpConnectionsStarted`

Table 43. Sniffer TCP metrics

Sniffer TCP	Metric Description	Topology Object Name
TCP - Protocol Error	Number of TCP connection errors discovered in the monitored traffic. Generally, this value should be zero or close to zero. A high number of errors indicates that problems are occurring at the TCP layer of the communication stack. Protocol errors that contribute to this metric include: • WSAEPROTOTYPE—Protocol wrong type for socket. • WSAENOPROTOOPT—Bad protocol option. • WSAEPROTONOSUPPORT—Protocol not supported. • WSAESOCKTNOSUPPORT—Socket type not supported. • WSAEOPNOTSUPP—Operation not supported. • WSAEPFNOSUPPORT—Protocol family not supported.	tcpConnections ProtocolError
TCP Segments - Client	Number of TCP segments originating from client machines. If this metric is zero, that indicates there is likely a problem with the network taps that are collecting traffic for the appliance. If at the same time, the *TCP Segments - Server* metric is greater than zero, that indicates that the appliance is seeing only one-way server-to-client traffic, and is not seeing the client-to-server traffic.	tcpSegmentsClient
TCP Segments - Client Missing	Number of segments in the client-to-server traffic that the agent has determined are missing. The agent reassembles TCP segments into complete messages before it begins its analysis. In this process, missing segments that are needed to build a message can be detected.	tcpSegmentsClient Missing
TCP Segments - Client Missing Percent	= TCP Segments - Client Missing / TCP Segments - Client Percentage of client segments that are missing. In ideal circumstances, this metric would always be zero. In practice, a small percentage (1% or lower) of missing segments is typical.	tcpSegmentsClient MissingPercent
TCP Segments - Server	Number of TCP segments originating from server machines. If this metric is zero, that indicates there is likely a problem with the network taps that are collecting traffic for the appliance. If, at the same time, the *TCP Segments - Client* metric is greater than zero, that indicates that the appliance is seeing only one-way client-to-server traffic, and is not seeing the server-to-client traffic.	tcpSegmentsServer
TCP Segments - Server Missing	Number of segments in the server-to-client traffic that the agent has determined are missing. The agent reassembles TCP segments into complete messages before it begins its analysis. In this process, missing segments that are needed to build a message can be detected.	tcpSegmentsServer Missing
TCP Segments - Server Missing Percent	= TCP Segments - Server Missing / TCP Segments - Server Percentage of server segments that are missing out of the total number of segments have been captured. In ideal circumstances, this metric would always be zero. In practice, a small percentage of missing segments is typical.	tcpSegmentsServer MissingPercent

Sniffer SSL Metrics

For a list of supported versions of SSL, key exchange algorithms, and cipher suites, look up “Supported Encryption Technologies” in the online help or the *APM Administration and Configuration Guide*.

Table 44. Sniffer SSL metrics

Sniffer SSL	Metric Description	Topology Object Name
SSL - Connection Error	Number of SSL communications in which an error is detected. For a list of the possible causes of SSL errors, see Summary of causes for SSL connection errors.	`sslConnectionsError`
SSL - Connections Dropped	Number of SSL connections dropped. Under normal conditions, this metric should be close to the value of the *SSL - Connections Started* metric.	`sslConnectionsDropped`
SSL - Connections Started	Number of new SSL communications initiated in the time range.	`sslConnectionsStarted`
SSL - Error Percent	= SSL-Connection Error / SSL-Connections Started Percentage of SSL connections where an error was detected.	`sslConnectionsError Percent`
SSL - Missing TCP Client Segment Percent	Same as TCP Segments - Client Missing Percent, but for SSL traffic only.	`sslMissingTCPClient SegmentPercent`
SSL - Missing TCP Client Segments	Same as TCP Segments - Client Missing, but for SSL traffic only.	`sslMissingTCPClient Segments`
SSL - Missing TCP Server Segment Percent	Same as TCP Segments - Client Missing Percent, but for SSL traffic only.	`sslMissingTCPServer SegmentPercent`
SSL - Missing TCP Server Segments	Same as TCP Segments - Server Missing, but for SSL traffic only.	`sslMissingTCPServer Segments`
SSL - TCP Client Segments	Same as TCP Segments - Client, but for SSL traffic only.	`sslTCPClientSegments`
SSL - TCP Server Segments	Same as TCP Segments - Server, but for SSL traffic only.	`sslTCPServerSegments`
SSL Error - Incomplete Key	Number of loaded key files that did not contain a valid private key. You can try re-exporting the private key from the web server and reloading it in Foglight.	`sslErrorIncompleteKey`
SSL Error - Invalid Key	Number of times an SSL key is invalid. Keys may be invalid for the following reasons: • An SSL key has expired. • An SSL key is incorrect. • For SafeNet HSM servers, there was an error decrypting the SSL handshake.	`sslErrorInvalidKey`
SSL Error - Invalid Record Format	Number of times an SSL record did not contain enough data to extract all of the fields needed to decrypt the session data. Packet drops may be triggering this error.	`sslErrorInvalidRecord Format`
SSL Error - Invalid State	Number of times the decryption process is aborted when packets are missing in the monitored traffic and the SSL cipher suite in use is a block cipher. Any value greater than 0 indicates that some traffic is being missed. You should check for missing segments using the SSL - Missing TCP Client Segment Percent and SSL - Missing TCP Server Segment Percent metrics.	`sslErrorInvalidState`
SSL Error - Memory	Number of times one or more SSL-related objects failed to be allocated. This situation may be caused when an appliance experiences a low memory condition.	`sslErrorMemory`

Table 44. Sniffer SSL metrics

Sniffer SSL	Metric Description	Topology Object Name
SSL Error - Missing Key	Number of missing SSL keys. Keys may be missing for the following reasons: • A file loaded for this server/port pair does not contain a valid key. • The file contains certificates that prevent the Sniffer from locating the key. In this case, an APM Administrator needs to add the keys in Foglight. • *SafeNet HSM*–The Sniffer cannot access the partition or private key on the server. • *SafeNet HSM*–The keys are no longer on the server.	`sslErrorMissingKey`
SSL Error - Unsupported Cipher	Number of times an unsupported cipher suite has been selected for encrypting the data between the client and the web server. This situation prevents Sniffers from decrypting the client's session. To resolve errors, try configuring SSL on the web server to use only supported cipher suites.	`sslErrorUnsupported Cipher`
SSL Error - Missing Reusable Secret	Number of times a new SSL connection wants to use a previously negotiated shared secret, but the Sniffer is unable to participate because it does not know the secret. In this case, the Sniffer cannot decrypt traffic and therefore ignores the connection. This situation occurs most frequently after an appliance is started or restarted. As clients negotiate new shared secrets with the restarted Sniffer, this error count tends to go down. **NOTE:** The web server's SSL configuration determines how long a previously negotiated shared secret can be re-used in new connections. Usually, these timers are set at less than 15 minutes. The longer the timer, the longer it takes the Sniffer to start decrypting sessions after a restart.	`sslErrorMissing ReusableSecret`
SSL Error - Unexpected Client Hello	Number of times the server received a duplicate Client Hello message. This situation should never happen and indicates that the session has entered an invalid state and should be ignored. If this metric is incremented frequently, a non-standard version of SSL may be in use.	`sslErrorUnexpected ClientHello`
SSL Error - Unexpected Server Hello	Number of times the client received a duplicate Server Hello message. This situation should never happen and indicates that the session has entered an invalid state and should be ignored. If this metric is incremented frequently, a non-standard version of SSL may be in use.	`sslErrorUnexpected ServerHello`
SSL Error - Unknown Handshake	Number of times an unknown handshake message was received during the startup phase of communication. This could indicate that a non-standard version of SSL is in use.	`sslErrorUnknown Handshake`
SSL Error - Unsupported Key Exchange	Number of times that an unsupported SSL key exchange algorithm has been selected for use between the client and server. This situation is most likely caused by use of the unsupported Diffie-Hellman Key Exchange algorithm. Foglight APM support only the RSA key exchange. To resolve errors, try configuring SSL on the web server to exclude all Diffie-Hellman algorithms.	`sslErrorUnsupported KeyExchange`
SSL Error - Unsupported Version	Number of times the Sniffer encountered an unsupported version of SSL.	`sslErrorUnsupported Version`

Archiver health metrics

Archiver health metrics are grouped into General Archiver health metrics, Archiver Hit and Page Handling metrics, Archiver Database metrics, and Archiver Search metrics.

You can find a selection of these metrics displayed in the following dashboards:

- **APM > Support > Appliance Health >** Archivers Quick View
- **APM > Support > Appliance Health >** Archiver: *SelectedArchiver dashboard*

General Archiver health metrics

Table 45. General Archiver health metrics

Archiver General	Metric Description	Topology Object Name
Batch Error	Number of hits that could not be processed due to an Archiver runtime failure. The logs on the appliance may contain details about these errors.	`batchError`
Batch Queue Count	Number of raw hits waiting to be captured. Each Archiver has a limited number of hits that it can buffer before capturing. The Archiver queue may often be at the limit. The Relayer buffers incoming hits in shared memory and feeds these raw hits to the Archiver.	`batchQueueCount`
Batch Queue Limit	Number of raw hits that can be stored before processing. Each Archiver has a limited number of hits that it can buffer. This metric value does not change while the Archiver is running.	`batchQueueLimit`
Cached Items	Number of unique data items currently held in the Archiver memory cache. A very large value may indicate a data quality problem or extreme load.	`cachedItems`
Capture Volume	Amount of raw traffic (in KB) received from all Sniffers. This measures the total network bandwidth consumed between the Sniffers and Archiver.	`captureVolume`
Capture Volume - Instrumentation	Amount of raw instrumentation traffic (in KB) received from all Sniffers. This measures the network bandwidth consumed by instrumentation traffic between the Sniffers and Archiver.	`captureVolume Instrumentation`
Clock Skew	Estimated difference (in milliseconds) between the clock on the appliance running this Sniffer and the clock on the appliance running the Management Server.	`clockSkew`
Configuration Available	Reports whether configuration polling to the Management Server was successful on the last attempt. When an Archiver is added to Foglight, it begins to contact the Management Server periodically for configuration settings. When an Archiver is restarted, it must be able to contact the Management Server successfully once before it is able to capture incoming hits.	`configAvailable`
Interaction Metric Count	Tracks the total number of metrics generated for all standard metrics, including websites and endpoints. The counter is limited to a maximum value of 5M. If this counter reaches the 5M limit, this is an indication that some metric data was lost.	`interactionMetricCount`

Table 45. General Archiver health metrics

Archiver General	Metric Description	Topology Object Name
JVM - Free Memory	Amount of unused memory within the Java Virtual Machine (JVM) running the Archiver. This value should oscillate as normal garbage collection occurs. A consistently low amount of free memory may indicate a problem. The amount of free memory can change very rapidly as the JVM automatically resizes the heap.	`jvmFreeMemory`
JVM - GC Time	Amount of time spent in garbage collection.	`jvmGCTime`
JVM - Total Memory	Amount of heap memory allocated to the Java Virtual Machine (JVM) running the Archiver. The JVM automatically resizes the heap to keep garbage collection times to a minimum. This value can increase suddenly when a spike in capture traffic occurs. It decreases gradually over periods of relative inactivity.	`jvmTotalMemory`
JVM - Used Memory	= JVM - Total Memory — JVM - Free Memory Amount of memory currently in use within the Java Virtual Machine (JVM) running the Archiver. This value should oscillate as normal garbage collection occurs. A consistently high amount of used memory may indicate a problem. The amount of used memory can change very rapidly as the JVM automatically resizes the heap.	`jvmUsedMemory`
RAM Filesystem Size	Size of the in-memory filesystem backing the open shard.	`ramFilesystemSize`
Refresh Metric Time	Amount of time required to recalculate metrics for the JVM, MySQL, caches, database counts (shards, hits, sessions, and aspects) and database size.	`refreshMetricTime`
Shared Memory Size	Amount of shared memory, which is used to store raw hits before they are sent to the Archiver for processing. Each Relayer has a limited number of hits that it can buffer. This metric value does not change while the Archiver is running.	`sharedMemorySize`
Shared Memory Used	Amount of shared memory consumed by raw hits waiting to be captured. Each Relayer has a limited amount of memory to buffer hits before being sent to the Archiver. When this buffer fills up, the Relayer stops reading batches from the network, which may cause hits to be discarded. Relayers should normally use very little shared memory, but short periodic bursts are normal. A sustained value above 25% of the shared memory size indicates the appliance is having difficulty keeping up with incoming traffic.	`sharedMemoryUsed`
Traffic Analysis - Custom Fields Active	Number of custom field facts defined by all active sessions and sequences. One fact is created for each custom field value. Larger values indicate increased memory usage.	`trafficAnalysis CustomFieldsActive`
Traffic Analysis - Rules Error	Number of times an internal failure has occurred in the rules engine. Any failures are most likely triggered by a recent change in your traffic analysis configuration. If failures have been reported, export your current configuration and contact Quest Technical Support.	`trafficAnalysis RulesError`

Table 45. General Archiver health metrics

Archiver General	Metric Description	Topology Object Name
Traffic Analysis - Rules Fired	Number of times a hit analyzer match condition or sequence analyzer event condition was true. Larger values indicate increased CPU time spent on traffic analysis. Creating hit analyzer or sequence analyzers tends to increase this value.	`trafficAnalysis RulesFired`
Traffic Analysis - Sequence Hit Analyzers Active	Number of hit analyzer facts defined by all active sequences. One fact is created for every sequence, for every dependent hit analyzer type matched. Larger values indicate increased memory usage.	`trafficAnalysis SequenceHit AnalyzersActive`
Traffic Analysis - Sequences Active	Number of sequence facts currently active in the Archiver rules engine used for traffic analysis. This value increases as start events are met for defined sequence analyzers, and decreases as sequences stop. Larger values indicate increased memory usage and CPU time spent on traffic analysis. Creating sequence analyzers tends to increase this value.	`trafficAnalysis SequencesActive`
Traffic Analysis - Sequences Completed	Number of sequences (across all sequence analyzers) that stopped and were written to the capture database.	`trafficAnalysis SequencesCompleted`
Traffic Analysis - Session Hit Analyzers Active	Number of hit analyzer facts defined by all active sessions. One fact is created for every session for every hit analyzer defined. Larger values indicate greater memory usage. Creating hit analyzers tends to increase this value.	`trafficAnalysis SessionHitAnalyzers Active`
Traffic Analysis - Session Timeout	Number of completed sessions that stopped due to a session timeout rule. For more information, see “Defining when a session has ended” in the online help or *Foglight APM Administration and Configuration Guide*.	`trafficAnalysis SessionTimeout`
Traffic Analysis - Sessions Active	Number of session facts currently active in the Archiver rules engine used for traffic analysis. This value increases as new session IDs are captured, and decreases as sessions stop. This indirectly measures the number of concurrent client browsers. Larger values indicate increased memory usage and higher concurrency. A large number of sessions may also indicate timeouts that are set too short.	`trafficAnalysis SessionsActive`
Traffic Analysis - Sessions Completed	Number of sessions that stopped and were written to the capture database.	`trafficAnalysis SessionsCompleted`
Write Buffer Failed	Number of times a write buffer was lost due to an internal error. The support bundle includes all the error details.	`writeBufferFailed`

Archiver Hit and Page Handling metrics

Table 46. Archiver Hit and Page Handling metrics

Archiver Hit and Page Handling	Metric Description	Topology Object Name
Hits - Analyzer Error	Number of hits marked as errors by a hit analyzer. All hits are considered OK until an error is detected.	`hitsAnalyzerError`
Hits - Analyzer Warning	Number of hits marked as warnings by a hit analyzer. All hits are considered OK until an error or warning is detected.	`hitsAnalyzerWarning`

Table 46. Archiver Hit and Page Handling metrics

Archiver Hit and Page Handling	Metric Description	Topology Object Name
Hits - Capture Error	Number of hits that were not captured properly. This indicates hits that were incomplete or corrupted, but no other error condition was present, such as a connection reset or timeout. This metric is updated automatically before traffic analysis begins.	`hitsCaptureError`
Hits - Captured	Number of hits the Sniffer has sent to the Archiver.	`hitsCaptured`
Hits - Capture Rate	= Hits - Captured / second Rate at which hits are captured per second.	`hitsCaptureRate`
Hits - Chunked Transfer Error	Number of hits using chunked encoding that could not be de-chunked due to invalid formatting. This metric is updated automatically before traffic analysis begins. These hits were captured normally, however the content may not display properly in session playback.	`hitsChunkedTransfer Error`
Hits - Client Connection Reset	Number of hits that were terminated by a connection reset by the client. Due to the abnormal termination of the hit, not all hit details, including content, may have been captured. This metric is updated automatically before traffic analysis processing begins.	`hitsClientConnection Reset`
Hits - Client Timeout	Number of hits that timed out. A timeout is determined by the Sniffer when the client does not respond within the TCP timeout period. This metric is updated automatically before traffic analysis processing begins.	`hitsClientTimeout`
Hits - Corrupt Cookie Name	Number of hits with corrupt cookie names. These hits were captured normally, however the invalid details were not written to the capture database. This metric is updated automatically before traffic analysis processing begins.	`hitsCorruptCookieName`
Hits - Corrupt Request Field Name	Number of hits with corrupt request field names. These hits were captured normally, however the invalid details were not written to the capture database. This metric is updated automatically before traffic analysis processing begins.	`hitsCorruptRequest FieldName`
Hits - Corrupt Request Header Name	Number of hits with corrupt request header names. These hits were captured normally, however the invalid details were not written to the capture database. This metric is updated automatically before traffic analysis processing begins.	`hitsCorruptRequest HeaderName`
Hits - Corrupt Response Content	Number of hits with a corrupt response. A corrupt response is usually due to either response headers that could not be parsed or compressed content that could not be inflated. This metric is updated automatically before traffic analysis processing begins.	`hitsCorruptResponse Content`
Hits - Corrupt Response Header Name	Number of hits with corrupt response header names. These hits were captured normally, however the invalid details were not written to the capture database. This metric is updated automatically before traffic analysis processing begins.	`hitsCorruptResponse HeaderName`
Hits - Discard Rate	= Hits - Discarded / second Rate at which an Archiver discards hits.	*Calculated on demand*

Table 46. Archiver Hit and Page Handling metrics

Archiver Hit and Page Handling	Metric Description	Topology Object Name
Hits - Discarded	Number of hits discarded by an Archiver before processing could complete. Each Archiver has a limited amount of memory to buffer hits before processing, and when this buffer fills up incoming hits are thrown away. Discarded hits are not counted as Hits Captured. Discards can also occur at the Sniffer before hits are sent.	`hitsDiscarded`
Hits - Drop Rate	= Hits - Dropped / second Rate at which an Archiver drops hits.	*Calculated on demand*
Hits - Dropped	Number of hits discarded by the Sniffer because a packet was dropped within the hit or SSL connection. This is typically because the Sniffer is overloaded.	`hitsDropped`
Hits - Excessive Clock Skew	Number of hits discarded because the hit time is so skewed that processing the hit would cause a segment error. This metric is updated automatically before traffic analysis processing begins.	`hitsExcessiveClockSkew`
Hits - HTML	Number of HTML hits the Sniffer has sent to the Archiver.	`hitsHTML`
Hits - HTML Parse Error	Number of HTML hits in which an error was encountered while parsing. This metric is updated automatically before traffic analysis processing begins.	`hitsHTMLParseError`
Hits - Instrumentation	Number of instrumentation hits that were captured and processed.	`hitsInstrumentation`
Hits - Missing Property	Number of hits that were missing one or more required properties. The required properties are: URL, Method, and ClientIP. This metric is updated automatically before traffic analysis processing begins.	`hitsMissingProperty`
Hits - Missing Response	Number of hits for which no response was captured. This is often caused by a connection reset or timeout. Due to the abnormal termination of the hit, not all hit details, including content, may have been captured. This metric is updated automatically before traffic analysis processing begins.	`hitsMissingResponse`
Hits - Missing Session Identifier	Number of hits that were not sessionized. This metric is updated automatically before traffic analysis processing begins.	`hitsMissingSessionID`
Hits - Response Content Dropped by Analyzer	Number of hits whose content was dropped by a hit analyzer. This metric is automatically updated upon completion of traffic analysis.	`hitsResponseContentDroppedByAnalyzer`
Hits - Response Content Dropped by Archiver	Number of *non-text* hits whose response content was dropped due to the fact that the captured content did not match the *Content-Length* header. This metric is updated automatically before traffic analysis processing begins.	`hitsResponseContentDroppedByArchiver`
Hits - Response Content Dropped by Sniffer	Number of *non-text* hits whose response content was dropped due to the fact that the captured content exceeded the maximum response size defined for the Sniffer. This metric is updated automatically before traffic analysis processing begins.	`hitsResponseContentDroppedBySniffer`
Hits - Response Content Incomplete	Number of text hits whose captured content did not match the Content-Length header. This metric is updated automatically before traffic analysis processing begins.	`hitsResponseContentIncomplete`
Hits - Response Content Keywords Indexed	Number of hits that were written to the content keyword index, as directed by a hit analyzer.	`hitsResponseContentKeywordIndexed`

Table 46. Archiver Hit and Page Handling metrics

Archiver Hit and Page Handling	Metric Description	Topology Object Name
Hits - Server Connection Reset	Number of hits that were terminated by a connection reset by either the client or the server. Due to the abnormal termination of the hit, not all hit details, including content, may have been captured. This metric is updated automatically before traffic analysis processing begins.	`hitsServerConnectionReset`
Hits - Server Timeout	Number of hits that timed out. A timeout is determined by the Sniffer when the server does not respond within the TCP timeout period. This metric is updated automatically before traffic analysis processing begins.	`hitsServerTimeout`
Hits - Truncated Request	Number of text hits whose request content was truncated due to the fact that the captured content exceeded the maximum request size defined. This metric is updated automatically before traffic analysis processing begins.	`hitsTruncatedRequest`
Hits - Truncated Response	Number of text hits whose response content was truncated due to the fact that the captured content exceeded the maximum response size. This metric is updated automatically before traffic analysis processing begins.	`hitsTruncatedResponse`
Hits - Unknown Exception	Number of hits that had an unknown exception. This metric is updated automatically before traffic analysis processing begins.	`hitsUnknownException`
Pages - Client Connection Reset	Number of pages that were terminated by a connection reset by the client. Due to the abnormal termination of the page, not all hit details, including content, may have been captured. This metric is updated automatically before traffic analysis processing begins.	`pagesClientConnectionReset`
Pages - Client Timeout	Number of pages that timed out. A timeout is determined by the Sniffer when the client does not respond within the TCP timeout period. This metric is updated automatically before traffic analysis processing begins.	`pagesClientTimeout`
Pages - Discarded	Number of pages discarded by an Archiver before processing could complete. Each Archiver has a limited amount of memory to buffer pages before processing, and when this buffer fills up incoming pages are thrown away. Discarded pages are not counted as Pages Captured. Discards can also occur at the Sniffer before pages are sent.	`pagesDiscarded`
Pages - Dropped	Number of pages discarded by the Sniffer because a packet was dropped within the page or SSL connection. This is typically because the Sniffer is overloaded.	`pagesDropped`
Pages - Captured	Number of pages the Sniffer has sent to the Archiver.	`pagesCaptured`
Pages - Server Connection Reset	Number of pages that were terminated by a connection reset by the server. Due to the abnormal termination of the page, not all page details, including content, may have been captured. This metric is updated automatically before traffic analysis processing begins.	`pagesServerConnectionReset`

Table 46. Archiver Hit and Page Handling metrics

Archiver Hit and Page Handling	Metric Description	Topology Object Name
Pages - Server Timeout	Number of pages that timed out. A timeout is determined by the Sniffer when the server does not respond within the TCP timeout period. This metric is updated automatically before traffic analysis processing begins.	`pagesServerTimeout`
Pages - Unknown Exception	Number of pages that had an unknown exception. This metric is updated automatically before traffic analysis processing begins.	`pagesUnknownException`

Archiver Database metrics

Table 47. Archiver Database metrics

Archiver Database	Metric Description	Topology Object Name
Database - Aspects	Number of aspects across all database shards. Aspects are created dynamically as needed to track fields, cookies, headers, and other searchable database objects. A large number of aspects can degrade search performance and may indicate a data-quality problem.	`databaseAspects`
Database - Hits	Number of hits in all shards of the capture database. This is the total count of hits available for search and replay. Hits are removed from the database as shards expire.	`databaseHits`
Database - Sessions	Number of sessions in all shards of the capture database. This is the total count of sessions available for search and replay. Sessions are removed from the database as shards expire.	`databaseSessions`
Database - Shards	Number of database shards currently used in the capture database. Each shards contains a time span of captured data. More shards are required for more replay history, but a large number of shards can degrade search performance. Shards are deleted as they expire.	`databaseShards`
Database - Size	Total size of all database shards currently stored in the Archiver databases.	`databaseSize`
MySQL - Key Blocks Max Used	Reports the value of the MySQL status variable `Key_blocks_used`, which is defined as the number of used blocks in the `MyISAM` key cache. This value is a high-water mark that indicates the maximum number of blocks that have ever been in use at one time.	`mysqlKeyBlocksMaxUsed`
MySQL - Key Blocks Not Flushed	Reports the value of the MySQL status variable `Key_blocks_not_flushed`, which is defined as the number of key blocks in the `MyISAM` key cache that have changed but have not yet been flushed to disk.	`mysqlKeyBlocksNotFlushed`
MySQL - Key Blocks Unused	Reports the value of the MySQL status variable `Key_blocks_unused`, which is defined as the number of unused blocks in the `MyISAM` key cache. You can use this value to determine how much of the key cache is in use.	`mysqlKeyBlocksUnused`

Table 47. Archiver Database metrics

Archiver Database	Metric Description	Topology Object Name
MySQL - Key Read Requests	Reports the value of the MySQL status variable `Key_read_requests`, which is defined as the number of requests to read a key block from the `MyISAM` key cache.	`mysqlKeyReadRequests`
MySQL - Key Reads	Reports the value of the MySQL status variable `Key_reads`, which is defined as the number of physical reads of a key block from disk into the `MyISAM` key cache. If `Key_reads` is large, then your `key_buffer_size` value is probably too small. The cache miss rate can be calculated as `Key_reads/Key_read_requests`.	`mysqlKeyReads`
MySQL - Key Write Requests	Reports the value of the MySQL status variable `Key_write_requests`, which is defined as the number of requests to write a key block to the `MyISAM` key cache.	`mysqlKeyWriteRequests`
MySQL - Key Writes	Reports the value of the MySQL status variable `Key_writes`, which is defined as the number of physical writes of a key block from the `MyISAM` key cache to disk.	`mysqlKeyWrites`
Shards - Closing	Number of shards currently being closed by the Archiver. This value should be between zero and one when the Archiver is healthy.	`shardsClosing`
Shards - Copying	Not implemented.	`shardsCopying`
Shards - Waiting to Copy	Not implemented.	`shardsWaitingCopy`

Archiver Search metrics

Table 48. Archiver Search metrics

Archiver Search	Metric Description	Topology Object Name
Searches - Browsers	Number of searches for all browsers.	`searchBrowsers`
Searches - Browser Categories	Number of searches for browser categories.	`searchBrowser Categories`
Searches - Cities	Number of searches for all cities.	`searchCities`
Searches - Content Categories	Number of searches for content categories.	`searchContent Categories`
Searches - Cookies	Number of searches for all cookie names.	`searchCookies`
Searches - Countries	Number of searches for all countries.	`searchCountries`
Searches - Endpoints	Number of searches for all endpoints (server plus port combinations).	`searchEndpoints`
Searches - Fields	Number of searches for all field names.	`searchFields`
Searches - Hits	Number of searches for captured hits.	`searchHits`
Searches - Hits Empty	Number of hit searches that returned zero results.	`searchHitsEmpty`
Searches - Hits Slow	Number of hit searches that took longer than ten seconds to complete.	`searchHitsSlow`
Searches - Hits Timeout	Number of hit searches that took longer than five minutes to complete.	`searchHitsTimeout`
Searches - Hit Details	Number of searches for hit details based on specific hit or session IDs, used for replay or export.	`searchHitDetails`

Table 48. Archiver Search metrics

Archiver Search	Metric Description	Topology Object Name
Searches - Hit Details Empty	Number of searches for hit details that returned zero results.	searchHitDetailsEmpty
Searches - Hit Details Slow	Number of searches for hit details that took longer than 15 seconds to complete.	searchHitDetailsSlow
Searches - Hit Details Timeout	Number of searches for hit details that took longer than five minutes to complete.	searchHitDetails Timeout
Searches - Hit Metric IDs	Number of searches for metric identifiers.	searchHitMetricIDs
Searches - ISPs	Number of searches for all ISPs.	searchISPs
Searches - Local Config	Number of searches for local Archiver configuration (hardware-specific settings).	searchLocalConfig
Searches - Metrics	Number of searches for raw metric values.	searchMetrics
Searches - Metric Deltas	Number of requests for five minute delta metrics.	searchMetricDeltas
Searches - Oldest Hits	Number of searches for the oldest hit.	searchOldestHits
Searches - Operating Systems	Number of searches for the operating system detail.	searchOperating Systems
Searches - Regions	Number of searches for regions.	searchRegions
Searches - Request Content	Number of searches for request content categories.	searchRequestContent
Searches - Request Headers	Number of searches for request headers.	searchRequestHeaders
Searches - Request Paths	Number of searches for request paths.	searchRequestPaths
Searches - Response Content	Number of searches for response content categories.	searchResponseContent
Searches - Response Headers	Number of searches for response headers.	searchResponseHeaders
Searches - Response HTML Titles	Number of searches for all HTML titles available. This metadata is used in the FxV browser interface for searching and building filters.	searchResponseHTML Titles
Searches - Sessions	Number of searches for captured sessions.	searchSessions
Searches - Sessions Empty	Number of session searches that returned zero results.	searchSessionsEmpty
Searches - Sessions Slow	Number of session searches that took longer than ten seconds to complete.	searchSessionsSlow
Searches - Sessions Timeout	Number of session searches that took longer than five minutes to complete.	searchSessionsTimeout
Searches - Session Metric IDs	Number of searches for session metric identifiers.	searchSessions MetricIDs
Searches - Sequences	Number of searches for captured sequences.	searchSequences
Searches - Sequences Empty	Number of sequences searches that returned zero results.	searchSequencesEmpty
Searches - Sequences Slow	Number of sequences searches that took longer than ten seconds to complete.	searchSequencesSlow
Searches - Sequences Timeout	Number of sequences searches that took longer than five minutes to complete.	searchSequences Timeout
Searches - Sequences Metric IDs	Number of searches for sequence metric identifiers.	searchSequences MetricIDs
Searches - Sniffer IDs	Number of searches for the Sniffer ID detail.	searchSnifferIDs

Table 48. Archiver Search metrics

Archiver Search	Metric Description	Topology Object Name
Searches - Subnets	Number of searches for the subnet detail.	`searchSubnets`
Searches - SOAP Operation Names	Number of searches for the SOAP Operation detail.	`searchSOAPOperation Names`
Searches - SOAP Web Service Names	Number of searches for the SOAP Web Service detail.	`searchSOAPWebService Names`
Searches - URL Scheme and Authorities	Number of searches for the URL Scheme and Authority detail.	`searchURLSchemeAnd Authorities`

Table 49. Archiver Stomp metrics

Archiver Stomp	Metric Description	Topology Object Name
stompConnectionsActive	The total number of connections currently active. A connection is created for each broker/login combination.	`stompConnectionsActive`
stompBytesSent	Count of message bytes sent on this connection.	`stompBytesSent`
stompConnectionError	Count of errors when attempting to establish a connection for this connection.	`stompConnectionErrors`
stompMessagesCaptured	Count of messages successfully sent on this connection.	`stompMessagesCaptured`
stompMessagesDiscardedExceedsMaxSize	Count of messages discarded due to max message size for this connection.	`stompMessagesDiscarded ExceedsMaxSize`
stompMessagesDiscardedQueueFull	Count of messages discarded due to upload queue full for this connection.	`stompMessagesDiscarded QueueFull`
stompMessagesDiscardedUploadError	Count of errors encountered while uploading messages on this connection.	`stompMessagesDiscarded UploadError`

Relayer health metrics

You can find a selection of these metrics displayed in the following dashboards:

- **APM > Support > Appliance Health >** Relayers Quick View
- **APM > Support > Appliance Health >** Relayers Quick View > Relayer: *SelectedRelayer dashboard*

Table 50. Relayer health metrics

Relayer Health Metric	Description	Topology Object Name
Relayer - Discarded Hits Due to Queue Full	Number of hits discarded by the Sniffer because the upload queue was full. Each Sniffer has a queue to buffer hits before sending them. When this queue fills up, hits are discarded.	`hitsDiscardedQueue Full` See Note.
Relayer - Discarded Messages	Number of messages (hits plus pages) that were discarded by the Sniffer before being sent to an Archiver. There are a number of reasons why a hits and pages may be discarded.	`messagesDiscarded`
Relayer - Discarded Messages Due to Exceeding Max Size	Number of messages (hits plus pages) that exceeded the max message size. The maximum message size is currently set at 1.2MB.	`messagesDiscarded MaxSizeExceeded`

Table 50. Relayer health metrics

Relayer Health Metric	Description	Topology Object Name
Relayer - Discarded Messages Due to Memory Full	Number of hits discarded by the Sniffer because the shared memory segment was full. Each Sniffer has a shared memory buffer used to transfer data. When shared memory fills up, incoming hits are discarded.	`messagesDiscarded MemoryFull`
Relayer - Discarded Pages Due to Queue Full	Number of pages discarded by the Sniffer because the upload queue was full. Each Sniffer has a queue to buffer pages before sending them. When this queue fills up, pages are discarded.	`pagesDiscardedQueue Full` See Note.
Relayer - Error Downloading Configuration	The Relayer periodically polls the Management Server for its configuration. This metric reports the number of times that the Relayer was unable to connect to the Management Server or download its configuration.	`configDownloadError`
Relayer - Hit Content Dropped Due to Exceeding Max Size	Number of hits whose content was dropped because the hit would have exceeded the maximum size limit. See also Relayer - Discarded Messages Due to Exceeding Max Size.	`hitContentDropped`
Relayer - Hits Capture Rate	Rate at which hits are captured per second. Each Sniffer has a queue to send hits to an Archiver. This rate is calculated for each queue.	`hitsCaptureRate` See Note.
Relayer - Late Arriving Data Timeouts	The Relayer merges data from multiple sources. For example, when a Sniffer sends both hit and page data, the Archiver's Relayer merges that data. The Relayer also merges some of the instrumentation data with the hit. The Relayer waits for a defined period of time, after which it sends the hit along to the Archiver without the additional data. This metric reports the number of times the Relayer timed out while waiting for the additional data to be merged with the hit.	`lateArrivingData Timeouts`
Relayer - Messages Transferred	Number of messages (hits plus pages) received by the Relayer and forwarded to the Archiver.	`messagesTransferred`
Relayer - Open Messages	Number of messages (hits plus pages) that are currently being captured by the Relayer.	`messagesOpen`
Relayer - Captured Hits	Number of hits captured and forwarded for this Archiver.	`hitsCaptured` See Note.
Relayer - Captured Pages	Number of pages captured and forwarded for this Archiver.	`pagesCaptured` See Note.
Relayer - Pages Capture Rate	Rate at which pages are captured per second. Each Sniffer has a queue to send hits to an Archiver. This rate is calculated for each queue.	`pagesCaptureRate` See Note.
Relayer - Queue Memory Unused Percent	Percentage of the upload queue (between Sniffers and Archivers) that is not currently in use. Each Sniffer has a queue for each Archiver to buffer messages before sending them. When this queue fills up, messages are discarded.	`queueMemoryUnused Percent` See Note.
Relayer - Queue Memory Used Percent	Percentage of the upload queue (between Sniffers and Archivers) that is currently in use. Each Sniffer has a queue for each Archiver to buffer messages before sending them. When this queue fills up, messages are discarded.	`queueMemoryUsed Percent` See Note.

Table 50. Relayer health metrics

Relayer Health Metric	Description	Topology Object Name
Relayer - Queued Messages	Number of messages (hits plus pages) currently in the upload queue between Sniffers and Archivers. The Hits Captured or Page Captured metric is not incremented until the hit or page is removed from the upload queue.	`queueMessagesCurrent` See Note.
Relayer - Socket Error Uploading Capture Data	Number of errors encountered while uploading capture data to the Archiver.	`captureUploadError` See Note.

Note

NOTE: Each Sniffer has a separate queue for each Archiver in its capture group. The following Relayer metrics are calculated per Sniffer, per Archiver:

- `captureUploadError`
- `hitsDiscardedQueueFull`
- `hitsCaptured`
- `hitsCaptureRate`
- `pagesDiscardedQueueFull`
- `pagesCaptured`
- `pagesCaptureRate`
- `queueMemoryUnusedPercent`
- `queueMemoryUsedPercent`
- `queueMessagesCurrent`

3

Standard metrics

Standard metrics feed all the top-level APM dashboards, such as the Geographical Perspective and the Web Sites and Endpoints dashboards. To generate standard metrics, APM Administrators define hit, sequence, and session analyzers. When an analyzer's conditions are satisfied, standard metrics are created and stored as topology objects in the Management Server database. Every five minutes, the current value of the metrics are captured in a snapshot and passed to the Management Server for display in the APM dashboards. For more information, see "Working with metrics" in the *Foglight APM Administration and Configuration Guide* online help.

The values for standard metrics are extracted from the captured web traffic data, which is stored as hit, session, and sequence details in the Archiver database. Standard metric names and detail names are closely aligned to highlight the relationships between them.

To find the top-level APM dashboards, in the navigation panel, under Dashboards, expand **APM**. The following dashboards (and their associated views) display a selection of standard metrics:

- **APM > Geographical Perspective**
- **APM > Sequence Explorer**
- **APM > Web Sites and Endpoints**

The following topics are covered in this section:

Understanding how standard metrics are organized

Standard metrics are organized into metric groups, which are described in the following sections. Depending on what types of activity an analyzer tracks, different groups of metrics are collected, as summarized in the following table.

Table 51. Standard metric groups

Type	Definition	Topology Type	Metric Groups
Hits	A hit is an HTTP request and its associated response. Typically, HTTP requests use the GET and POST methods. Responses are returned from the web site in various forms such as HTML, text, images, JavaScript, other scripts, CSS files, frames, applets, or any other web-viewable object.	`EUHitMetrics` `EUHitExpanded Metrics`	• Capture metrics • Volume metrics • Performance metrics • Hit metrics • Hit request/response metrics • Navigation timing metrics • Status metrics • Status expanded metrics
Pages	A page is a web document that usually contains HTML and is presented in a browser as a complete window to the user of a web site. Included in a page are all the hits that are triggered for images, style sheets, and other components when the user navigates to a URL for the page.	`EUPageMetrics`	• Capture metrics • Volume metrics • Performance metrics • Hit metrics • Page metrics • Status metrics • Status expanded metrics
Sequences	A sequence tracks all pages and hits experienced by the end user while navigating the steps in a process (as defined by the events in a sequence analyzer).	`EUSequence Interaction Metrics`	• Capture metrics • Sequence metrics • Status metrics • Status rollup metrics
Sequence Events	A sequence event is a step within a sequence.	`EUASequence Interaction EventMetrics`	• Capture metrics • Sequence event metrics
Sessions	A session contains all the sequences, pages, and hits experienced by the end user during a user session.	`EUSession Interaction Metrics`	• Capture metrics • Status metrics • Status rollup metrics
Navigation Timings	Timing details reflect the entire performance of a page, including third-party content called by the page. Timing details are collected through browser instrumentation that relies on the W3C Navigation Timing standard.	`EUNavigation TimingMetrics`	• Capture metrics • Navigation timing metrics
Script Execution	Information about scripts executed.	`EUScript Execution Metrics`	• Script execution metrics

Capture metrics

Capture metrics describe the number and rate of download attempts.

Table 52. Capture metrics

Capture Metric	Metric Description	Topology Object Name
Captured	Number of hits or pages captured by the system. For a sequence, the number of times the sequence occurred. For a sequence event, the number of times the event occurred.	`captured`
Capture Rate	= Captured / second Rate of download attempts.	`captureRate`

Volume metrics

Volume metrics describe the volume of download attempts.

Table 53. Volume metrics

Volume Metric	Metric Description	Topology Object Name
Volume	= Volume of Client Requests + Volume of Server Responses Total size of HTTP messages sent by both clients and servers. This includes content and headers for both requests and responses.	`volume`
Volume of Client Request Content	Size of the HTTP content (or payload) in requests sent from clients. For SOAP applications, this will be the size of the SOAP message. Web applications typically do not have much request content. This metric does not include the size of HTTP, IP and TCP headers.	`volumeRequestContent`
Volume of Client Request Headers	Size of the HTTP headers in requests sent from clients.	`volumeRequestHeaders`
Volume of Client Requests	Number of unique application bytes sent from the client to the web server. The application bytes include the HTTP header and content. The byte count *excludes* bytes from the link level header, IP header, TCP header, SSL header, and invalid application bytes. For more information, see Calculating volume.	`volumeRequestTotal`
Volume of Server Response Content	Size of the HTTP content (or payload) in responses sent from servers. For web sites, this will be the size of the HTML content. For SOAP applications, this will the size of the SOAP message. This metric does not include the size of HTTP, IP and TCP headers.	`volumeResponseContent`
Volume of Server Response Headers	Size of the HTTP headers in responses sent from servers.	`volumeResponseHeaders`
Volume of Server Responses	Number of unique application bytes sent from the web server to the client. The application bytes include the HTTP header and content. The byte count *excludes* bytes from the link level header, IP header, TCP header, SSL header, and invalid application bytes. For more information, see Calculating volume.	`volumeResponseTotal`

Performance metrics

The performance metrics calculate time spent by web servers responding to client requests for either a page or a hit.

Table 54. Performance metrics

Performance	Metric Description	Topology Object Name
Back End Time	Amount of time spent by web servers processing HTTP requests sent from client browsers. For more information, see Understanding what is included in Back End Time, Tracking time spent on a hit, and Tracking time spent on a page.	`backEndTime`
Back End Time Exceeds Threshold	Number of times that the value of Back End Time exceeded the defined threshold.	`backEndTimeExceeds Threshold`
Back End Time Exceeds Threshold Percent	Percentage of hits or pages where the Back End Time exceeded the defined threshold.	`backEndTimeExceeds ThresholdPercent`
Back End Time Service Level	Percentage of requests processed by web servers within the time specified as the service level agreement (SLA) threshold. This metric enables you to create service level contracts based on the request processing times for a server. **NOTE:** Consider creating rules based on this metric that trigger alarms when performance begins to degrade and when it falls below your SLA. For example, if your SLA specifies that 80% of requests must be processed in 250 milliseconds or faster, you can design a rule that issues a warning when this metric falls between 85% and 80%, and sends a critical alarm when performance falls below 80%.	`backEndTimeService Level`
End to End Time	= Back End Time + Client Time + Network Delay Amount of time required to download a hit or page. For more information, see Tracking time spent on a page and Tracking time spent on a page. The metric contains the time taken for both successful and unsuccessful downloads, however it does not include time for hits that failed to download due to a server timeout. The contents of the URL must be downloaded successfully at least once by at least one user before the metric is recorded. This check prevents invalid entries from being created when hacker attacks generate requests for non-existent URLs.	`endToEndTime`
End to End Time Exceeds Threshold	Number of times that the value of End To End Time exceeded the defined threshold.	`endToEndTimeExceeds Threshold`

Table 54. Performance metrics

Performance	Metric Description	Topology Object Name
End to End Time Exceeds Threshold Percent	Percentage of hits or pages where the End to End Time exceeded the defined threshold.	`endToEndTimeExceeds ThresholdPercent`
End to End Time Service Level	= # of page downloads satisfying threshold / # of page downloads Percentage of page downloads that satisfy the service level agreement (SLA) threshold for the End-to-End Time metric. This metrics enables you to create service level contracts based on the page download times for an application or across the entire enterprise. **NOTE:** Consider creating rules based on this metric that trigger alarms when performance begins to degrade and when it falls below your SLA. For example, if your SLA specifies that 80% of page downloads must be occur within four seconds, you can design a rule that issues a warning when this metric falls between 85% and 80%, and sends a critical alarm when performance falls below 80%.	`endToEndTimeService Level`

Hit metrics

Table 55. Hit metrics

Hit Metric	Metric Description	Topology Object Name
Client Time	Amount of time spent on the client side of the network during a hit or page download. This metric includes: • Network delays that occur when transmitting the response to the client. • Time the client spends processing the responses it receives. • Network delay for the transmission of the final acknowledgement from the client. For more information, see Understanding what is included in Client Time, Tracking time spent on a hit, and Tracking time spent on a page.	`clientTime`
Encrypted	Number of hits or pages that are encrypted.	`encrypted`
Encrypted Percent	Percentage of hits or pages that are encrypted.	`encryptedPercent`

Table 55. Hit metrics

Hit Metric	Metric Description	Topology Object Name
Initial Response Time	Amount of time required to process the initial request for a page, build a response, and begin transmitting the response. This metric does not include any network delays or client-side processing delays. For more information, see Tracking time spent on a hit and Tracking time spent on a page. **NOTE:** Some commands include a request, but not a response, due to the client halting the command (with a TCP Reset) or due to the server being overloaded. For these commands, the value of the Initial Response Time metric is undefined.	`initialResponseTime`
Network Delay	= Current Smoothed Round Trip Time / 2 Estimated amount of time for the client request to reach the web server, and for the final response packet to reach the client (assuming there is no client acknowledgement required). The Smoothed Round Trip Time (SRRT) is an averaged estimate of the round-trip times experienced during the life of a TCP connection. For more information, see Estimating network delay.	`networkDelay`

Page metrics

Table 56. Page metrics

Page Metric	Metric Description	Topology Object Name
Hits Captured	Number of hits captured.	`hitsCaptured`
Hits Redirects	Number of hits that are redirected with an HTTP redirection response code (3xx). When a request results in a redirect, the resource is not downloaded. Instead the server usually transmits a new URL with the redirected location of the resource or informs the browser that the resource has not changed since it was last requested. In this latter case, the browser knows it can use a local copy of the requested resource from its cache.	`hitsRedirects`
Hits Redirect Percent	Percentage of hits that are redirected.	`hitsRedirectsRate`
Incomplete Download	Number of hits where a TCP reset occurs before the hit is completed.	`incompleteDownload`
Incomplete Download Percent	Percentage of hits where the download did not complete due to a TCP Reset.	`incompleteDownload Rate`
Think Time	Amount of time a user spends on a dowloaded page (reading and thinking about the content) before going to another page. The timer that tracks think time *starts* when a page is fully downloaded and *stops* when the next page is requested.	`thinkTime`

Session metrics

Session metrics describe the user's session, including sequence-related counts.

Table 57. Session metrics

Session Metric	Metric Description	Topology Object Name
Access Speed	Effective speed at which an end user accesses a web application or web site. For more information, see Calculating access speed.	`accessSpeed`
Duration	Length of the session.	`duration`
Sequences Captured	Total number of sequences captured during user sessions.	`sequencesCaptured`
Sequences Completed	Total numbers of sequences completed during user sessions.	`sequencesCompleted`
Sequences With Error Status	Total numbers of sequences with an error status during user sessions.	`sequencesWithError`
Sequences With Exit	Total numbers of sequences that exited during user sessions.	`sequencesWithExit`
Sequences With Warning Status	Total numbers of sequences with a warning status during user sessions.	`sequencesWithWarning`

Sequence metrics

Sequence metrics are created for each sequence analyzer. All end user activity that initiated the sequence contribute to this metric.

Table 58. Sequence metrics

Sequence	Metric Description	Topology Object Name
Duration	Length of the sequence.	`duration`
Events Captured	Number of sequence events captured for the sequence. An event can occur more than once during a sequence, and the total number of events are counted in this metric.	`eventsCaptured`
Distinct Events Captured	Number of distinct sequence events for a sequence. Events that occur more than once are counted only once in this metric.	`eventsCaptured Distinct`
Exited After Hit Back End Time Exceeded Threshold	Number of times the sequence exited due to Back-End time exceeding a threshold.	`exitedAfterHitBackEnd TimeExceedsThreshold`
Exited After Hit End to End Time Exceeds Threshold	Number of times the sequence exited due to End-to-End time exceeding a threshold.	`exitedAfterHitEndTo EndTimeExceeds Threshold`
Exited After Hit Server Connection Reset Error	Number of times the sequence exited due to a server connection reset error.	`exitedAfterHitError ConnectionResetServer`
Exited After Hit Content Error	Number of sequences that exited due to a hit content error.	`exitedAfterHitError Content`
Exited After Hit Custom Error	Number of sequences that exited due to a hit custom error.	`exitedAfterHitError Custom`
Exited After Hit Http Error	Number of sequences that exited due to a hit HTTP error.	`exitedAfterHitError Http`
Exited After Hit Http Client Error	Number of sequences that exited due to a hit HTTP client error.	`exitedAfterHitError HttpClient`
Exited After Hit Http Server Error	Number of sequences that exited due to a hit HTTP server error.	`exitedAfterHitError HttpServer`

Table 58. Sequence metrics

Sequence	Metric Description	Topology Object Name
Exited After Hit Timeout	Number of sequences that exited due to a hit timeout.	`exitedAfterHitErrorTimeout`
Exited After Page Back End Time Exceeded Threshold	Number of sequences that exited due to page back end time exceeding threshold.	`exitedAfterPageBackEndTimeExceedsThreshold`
Exited After Page End to End Time Exceeds Threshold	Number of sequences that exited due to page end to end time exceeding threshold.	`exitedAfterPageEndToEndTimeExceedsThreshold`
Exited After Page Server Connection Reset Error	Number of sequences that exited due to a page server connection reset error.	`exitedAfterPageErrorConnectionResetServer`
Exited After Page Content Error	Number of sequences that exited due to a page content error.	`exitedAfterPageErrorContent`
Exited After Page Custom Error	Number of sequences that exited due to a page custom error.	`exitedAfterPageErrorCustom`
Exited After Page Http Error	Number of sequences that exited due to a page HTTP error.	`exitedAfterPageErrorHttp`
Exited After Page Http Client Error	Number of sequences that exited due to a page HTTP client error.	`exitedAfterPageErrorHttpClient`
Exited After Page Http Server Error	Number of sequences that exited due to a page HTTP server error.	`exitedAfterPageErrorHttpServer`
Exited After Page Timeout Error	Number of sequences that exited due to a page timeout.	`exitedAfterPageErrorTimeout`
Exited After Navigation Timing Content Delivery Time Exceeded Threshold	Number of sequences that exited due to the navigation timing content delivery time exceeding threshold.	`exitedAfterNavigationTimingContentDeliveryTimeExceedsThreshold`
Exited After Navigation Timing Page Completion Time Exceeds Threshold	Number of sequences that exited due to the navigation timing page completion time exceeding threshold.	`exitedAfterNavigationTimingPageCompletionTimeExceedsThreshold`
Exited After Repeating the Last Event	Number of sequences that exited due to repeating the last event.	`exitedAfterRepeat`
Exited After Repeating the Last Event Multiple Times	Number of sequences that exited due to repeating the last event multiple times.	`exitedAfterMultipleRepeats`
Completed (sequence)	Number of sequences that did not timeout and that did not have a final status of Error.	`completed`
Completed Percent	Percentage of sequences that did not timeout and that did not have a final status of Error.	`completedPercent`
Completed Rate	= Completed (sequence) / sec	`completedRate`
Duration	Time spent by users traversing the steps of the sequence.	`duration`
Exited (sequence)	Number of sequences that timed out.	`exited`
Exited Percent	Percentage of sequences that timed out.	`exitedPercent`
Exited Rate	= Exited (sequence) / sec	`exitedRate`
Failed (sequence)	Number of sequences that did not timeout, but had a final status of Error.	`failed`

Table 58. Sequence metrics

Sequence	Metric Description	Topology Object Name
Failed Percent	Percentage of sequences that did not timeout, but had a final status of Error.	`failedPercent`
Failed Rate	= Failed (sequence) / sec	`failedRate`

Sequence event metrics

A set of sequence event metrics are created for each event in a sequence.

Table 59. Sequence event metrics

Sequence	Metric Description	Topology Object Name
Completed (event)	Number of times an event was the last event in a Completed (sequence).	`completed`
Completed Percent	Percentage of times an event was the last event in a Completed (sequence).	`completedPercent`
Completed Rate	= Completed / sec	`completedRate`
Exited (event)	Number of times an event was the last event in an Exited (sequence).	`exited`
Exited Percent	Percentage of times an event was the last event in an Exited (sequence).	`exitedPercent`
Exited Rate	= Exited (event) / sec	`exitedRate`
Failed (event)	Number of times an event was the last event in a Failed (sequence).	`failed`
Failed Percent	Percentage of times an event was the last event in a Failed (sequence).	`failedPercent`
Failed Rate	= Failed (event) / sec	`failedRate`
Repeated	Number of times an event repeated in a sequence. Repeated means that the event occurred more than once in succession in the same sequence.	`repeated`
Repeated Percent	Percentage of times an event repeated in a sequence.	`repeatedPercent`
Repeated Rate	= Repeated / sec	`repeatedRate`

Navigation timing metrics

Navigation timing metrics are based on the details captured by the Foglight APM instrumentation library using the WC3 Navigation Timing API.

TIP: Most current browsers support the Navigation Timing API. To find out which earlier browser versions support the API, search the Internet for “navigation timing api browser support.”

The following table maps the metric name to its topology object. For metric definitions and an explanation of navigation timing, see Navigation timing metrics.

Table 60. Metric names - topology object mapping

Navigation Timing	Topology Object Name
App Cache Time	`appCacheTime`
Content Delivery Time	`contentDeliveryTime`

Table 60. Metric names - topology object mapping

Navigation Timing	Topology Object Name
Content Delivery Time Exceeds Threshold	`contentDeliveryTimeExceedsThreshold`
Content Delivery Time Exceeds Threshold Percent	`contentDeliveryTimeExceedsThresholdPercent`
Content Delivery Time Service Level	`contentDeliveryTimeServiceLevel`
DNS Time	`dnsTime`
Load Time	`loadTime`
Page Completion Time	`pageCompletionTime`
Page Completion Time Exceeds Threshold	`pageCompletionTimeExceedsThreshold`
Page Completion Time Exceeds Threshold Percent	`pageCompletionTimeExceedsThresholdPercent`
Page Completion Time Service Level	`pageCompletionTimeServiceLevel`
Processing Time	`processingTime`
Redirect Time	`redirectTime`
Request Time	`requestTime`
Response Time	`responseTime`
TCP Time	`tcpTime`
Unload Time	`unloadTime`

Hit request/response metrics

Table 61. Hit request/response metrics

Hit Request	Metric Description	Topology Object Name

Request Method <*command*>	Each version of this metric contains the number of times that the specified command was sent to the server. The <*command*> can be any of the following methods:
	• `CONNECT`
	• `DELETE`
	• `GET`
	• `HEAD`
	• `OPTIONS`
	• `POST`
	• `PUT`
	• `TRACE`
		`requestMethod`<*commmand*>
		
		where <*command*> can be one of the following methods:
		`CONNECT`
		`DELETE`
		`GET`
		`HEAD`
		`OPTIONS`
		`POST`
		`PUT`
		`TRACE`

Request Method <*command*> Percent	Each version of this metric contains the percentage of times that the specified command was sent to the server. For a list of commands, see Request Method <*command*>.	`requestMethod`<*commmand*> `Percent`

Table 61. Hit request/response metrics

Hit Request	Metric Description	Topology Object Name

Response Code <*code#*>	Each version of this metric contains the number of times the specified response code appeared in monitored traffic. For every HTTP command, the web server returns a HTTP response code to indicate the success or failure of the command. The categories and most common response codes are:
	• 1xx: Informational - (100: Continue)
	• 2xx: Success - (200: OK)
	• 3xx: Redirect - (301: Moved Permanently; 303: Not Modified)
	• 4xx: Client Error - (404: Object Not Found)
	• 5xx: Server Error - (503: Service Unavailable)
		`responseCode<code#>`
		
		where <*code#*> can be one of the following three-digit codes:
		`100,101,200,201,202, 203,204,205,206,`
		`300,301,302,303,304, 305,306,307,`
		`400,401,402,403,404, 405,406,407,408,409, 410,411,412,413,414, 415,416,417,428,429, 431,`
		`500,501,502,503,504, 505,511`

Response Code <*code#*> Percent	Each version of this metric contains the percentage of times the specified response code appeared in monitored traffic. For a list of codes, see Response Code <*code#*>.	`responseCode<code#> Percent`

Status metrics

Table 62. Status metrics

Status Metric	Metric Description	Topology Object Name
Error Status	Number of hits where an error has occurred.	`error`
Error Status Percent	Percentage of hits where an error has occurred.	`errorPercent`
Error Status Rate	Number of hits per second where an error has occurred.	`errorRate`
Ok Status	Number of hits where no error or warning has occurred.	`ok`
Ok Status Percent	Percentage of hits where no error or warning occurred.	`okPercent`
Warning Status	Number of hits where a warning has occurred.	`warning`
Warning Status Percent	Percentage of hits where a warning has occurred.	`warningPercent`

Status expanded metrics

Table 63. Status expanded metrics

Status Metric	Metric Description	Topology Object Name
Capture Exception	Number of hits where an exception occurred during capture.	`exception`
Capture Exception Percent	Percentage of hits where an exception occurred during capture.	`exceptionPercent`

Table 63. Status expanded metrics

Status Metric	Metric Description	Topology Object Name
Content Error	Number of hits where an error in the request or response content occurred. An example includes an HTTP GET request for a JPEG file resulting in an HTML response. For more information, see Understanding content categories and content error counts.	`errorContent`
Content Error Percent	Percentage of hits where a content error occurred.	`errorContentPercent`
Custom Error	Number of hits where a Hit Analyzer detected conditions that indicate an error had occurred and changed the status of the hit to “Error.”	`errorCustom`
Http Error	Number of hits where an HTTP response code indicated a client error (400-499) or a server error (500-599).	`errorHttp`
Http Client Error	Number of HTTP client errors that have occurred. These client errors are identified by the HTTP response code that is sent back by the web server. Response codes in the range from 400-499 represent client errors. Some common HTTP client errors are: 400 - Bad Request 401 - Unauthorized 402 - Payment Required 403 - Forbidden 404 - Object Not Found 405 - Method Not Allowed 406 - Not Acceptable 407 - Proxy Authentication Required 408 - Request Timeout 409 - Conflict 410 - Gone 411 - Length Required 412 - Precondition Failed 413 - Request Entity Too Large 414 - Request URI Too Long 415 - Unsupported Mail Type	`errorHttpClient`
Http Client Error Percent	Percentage of hits where an HTTP response code in the 400-499 range was returned.	`errorHttpClient Percent`
Http Error Percent	Percentage of hits where an HTTP response code in the 400-599 range was returned.	`errorHttpPercent`
Http Error Rate	Number of hits per second where an HTTP error has occurred.	`errorHttpRate`
Http Server Error	Number of HTTP server errors that have occurred. These server errors are identified by the HTTP response code that is sent back by the web server. Response codes in the range from 500-599 represent server errors. Some common HTTP server errors are: 500 - Internal Server Error 501 - Not Implemented 502 - Bad Gateway 503 - Service Unavailable 504 - Gateway Timeout	`errorHttpServer`

Table 63. Status expanded metrics

Status Metric	Metric Description	Topology Object Name
Http Server Error Percent	Percentage of hits where an HTTP response code in the 500-599 range was returned.	errorHttpServer Percent
Server Connection Reset Error	Number of hits where the server closed the TCP connection using a TCP reset command before the hit was completed. The TCP reset command is a TCP segment with the RESET flag set. After the reset is sent, both client and server can no longer send or receive any data on the connection. TCP resets are typically sent when either the TCP connection is in an invalid state or either side needs to quickly shutdown the connection to reclaim resources in use by the connection. A TCP reset can be sent at any time during a connection: during connection setup, data exchange or during through the close sequence.	errorConnectionReset Server
Server Connection Reset Error Percent	Percentage of hits where a server reset the TCP connection before the hit was complete.	errorConnectionReset ServerPercent
Timeout	Number of hits where a timeout occurred. A timeout can occur when a command request is sent from a client and the server begins to transmit a response, but the response is never completed.	errorTimeout
Timeout Percent	Percentage of hits where a timeout occurred.	errorTimeoutPercent
Timeout Rate	Number of hits per second where a timeout has occurred.	errorTimeoutRate
Client Timeout	Number of hits where a client timeout occurred. A client timeout can occur when a command request is sent from a client and the server begins to transmit a response, but the client stops responding before the response is complete.	errorTimeoutClient
Client Timeout Percent	Percentage of hits where a client timeout occurred.	errorTimeoutClient Percent
Server Timeout	Number of hits where a server timeout occurred. A server timeout can occur when a command request is sent from a client and the server begins to transmit a response, but the server stops responding before the response is complete.	errorTimeoutServer
Server Timeout Percent	Percentage of hits where a server timeout occurred.	errorTimeoutServer Percent
Client Connection Reset Warning	Number of hits where the client closed the TCP connection using a TCP reset command before the hit was completed. The TCP reset command is a TCP segment with the RESET flag set. After the reset is sent, both client and server can no longer send or receive any data on the connection. TCP resets are typically sent when either the TCP connection is in an invalid state or either side needs to quickly shutdown the connection to reclaim resources in use by the connection. A TCP reset can be sent at any time during a connection: during connection setup, data exchange or during the close sequence.	warningConnection ResetClient
Client Connection Reset Warning Percent	Percentage of hits where the client reset the connection.	warningConnection ResetClient Percent

Table 63. Status expanded metrics

Status Metric	Metric Description	Topology Object Name
Custom Warning	Number of hits where a Hit Analyzer detected conditions that indicate an error had occurred and changed the status of the hit to “Warning.”	`warningCustom`
Custom Warning Percent	Percentage of hits where a custom warning occurred.	`warningCustomPercent`

Status rollup metrics

Table 64. Status rollup metrics

Status Metric	Metric Description	Topology Object Name
Hits Captured	Total number of hits captured.	`hitsCaptured`
Download Hits Captured	Number of hits that are file downloads such as PDF and DOC files.	`hitsCapturedDownload`
Favicon Hits Captured	Number of hits for the “favicon.ico” file.	`hitsCapturedFavicon`
HTML Hits Captured	Number of hits containing HTML content.	`hitsCapturedHTML`
HTML Fragment Hits Captured	Number of hits where the response contains a fragment of HTML.	`hitsCapturedHTML Fragment`
Image Hits Captured	Number of hits where the request file is an image file such as a JPEG or PNG.	`hitsCapturedImage`
Instrumentation Hits Captured	Number of hits that are generated by client-side instrumentation.	`hitsCaptured Instrumentation`
JavaScript Hits Captured	Number of hits for JavaScript files.	`hitsCaptured JavaScript`
JSON Hits Captured	Number of hits where a JSON response was sent from the server.	`hitsCapturedJSON`
Media Hits Captured	Number of hits where the request file is a media file such as MPEG or WAV.	`hitsCapturedMedia`
Other Hits Captured	Number of hits that do not fall into the other content categories.	`hitsCapturedOther`
SOAP Hits Captured	Number of hits that contain SOAP messages.	`hitsCapturedSOAP`
StyleSheet Hits Captured	Number of hits for stylesheet files.	`hitsCapturedStyle Sheet`
XML Hits Captured	Number of hits that contain XML content.	`hitsCapturedXML`
Hits With Back End Time Exceeded Threshold	Number of hits where the Back-End time exceeded a defined threshold.	`hitsWithBackEndTime ExceedsThreshold`
Hits With End to End Time Exceeds Threshold	Number of hits where the End-to-End time exceeded a defined threshold.	`hitsWithEndToEndTime ExceedsThreshold`
Hits With Server Connection Reset Error	Number of hits where a server reset the TCP connection before the hit was completed.	`hitsWithError ConnectionReset Server`
Hits With Content Error	Number of hits with a content error. See Content Error.	`hitsWithErrorContent`
Hits With Custom Error	Number of hits with a custom error. See Custom Error.	`hitsWithErrorCustom`
Hits With Http Error	Hits where an HTTP error occurred. See Http Error.	`hitsWithErrorHttp`
Hits With Http Client Error	Hits where an HTTP client error occurred. See Http Client Error.	`hitsWithErrorHttp Client`

Table 64. Status rollup metrics

Status Metric	Metric Description	Topology Object Name
Hits With Http Server Error	Number of hits where an HTTP server error occurred. See Http Server Error.	hitsWithErrorHttp Server
Hits With Timeout	Number of hits where a timeout occurred.	hitsWithErrorTimeout
Hits With Warning	Number of hits that had a Hit Analyzer change the status of the hit to “Warning.”	hitsWithWarning
Navigation Timings Captured	Number of hits where navigation timing metrics were collected.	navigationTimings Captured
Navigation Timings With Content Delivery Time Exceeded Threshold	Number of hits where the Content Delivery Time exceeded a defined threshold.	navigationTimingsWith ContentDeliveryTime ExceedsThreshold
Navigation Timings With Page Completion Time Exceeds Threshold	Number of hits with a Page Completion Time that exceeded a defined threshold.	navigationTimingsWith PageCompletionTime ExceedsThreshold
Pages Captured	Number of pages captured.	pagesCaptured
Pages With Back End Time Exceeded Threshold	Number of pages where the Back-End Time exceeded a defined threshold.	pagesWithBackEndTime ExceedsThreshold
Pages With End to End Time Exceeds Threshold	Number of hits where the End-to-End Time exceeded a defined threshold.	pagesWithEndToEndTime ExceedsThreshold
Pages With Server Connection Reset Error	Number of pages where a server reset the TCP connection before the hit was completed.	pagesWithError ConnectionReset Server
Pages With Content Error	Number of pages with a content error. See Content Error.	pagesWithErrorContent
Pages With Custom Error	Number of pages with a custom error. See Custom Error.	pagesWithErrorCustom
Pages With Custom Error	Number of pages with a custom error. See Custom Error.	pagesWithErrorHttp
Pages With Http Error	Pages where an HTTP error occurred. See Http Error.	pagesWithErrorHttp
Pages With Http Client Error	Pages where an HTTP client error occurred. See Http Client Error.	pagesWithErrorHttp Client
Pages With Http Server Error	Number of pages where an HTTP server error occurred. See Http Server Error.	pagesWithErrorHttp Server
Pages With Timeout	Number of pages where a timeout occurred.	pagesWithErrorTimeout

Script execution metrics

Table 65. Script execution metrics

Status Metric	Metric Description	Topology Object Name
Script Execution Count	Number of times the script executed.	executionCount
Script Error Count	Number of script executions that resulted in a script error. Script errors can be caused by uncaught exceptions, violations in the security policy, or explicitly by the script invoking the fail method. **NOTE:** A script execution ends when an error occurs, therefore each execution with an error increments this metric by one.	errorCount

4

Archiver database details

The values for standard metrics are extracted or calculated from the captured web traffic data, which is stored as hit, session, and sequence details in the Archiver database. You can use these Archiver details as match conditions for analyzers or searches.

Some of these details are calculated. For more information, see Understanding calculated data.

The details are summarized in the following sections:

- Hit, session, and sequence details
- Page details (available when a hit is also a page)
- Navigation timing metrics (available when using instrumentation)

Hit, session, and sequence details

The Archiver saves the following details for hits, sessions, and/or sequences (as indicated).

Table 66. Hit, session, and sequence details

Hit Detail	Description	Hit	Session	Sequence
Access Speed	Estimates the speed of the end user's network connection. For more information, see Calculating access speed.	—	✓	—
Back-End Time	The total time spent by back-end servers processing HTTP requests sent from the client. For more information, see Tracking time spent on a hit and Understanding what is included in Back End Time.	✓	—	—
Back-End Time Exceeds Threshold	Boolean that indicates if the value of Back-End Time exceeded the defined threshold.	✓	—	—
Base HREF	Value contained in the HTML `<base href>` tag.	✓	—	—
Browser	Browser name and version, such as Microsoft Internet Explorer 9.0.	✓	✓	–
Browser Category	Browser family, such as Mozilla Firefox.	✓	✓	–
City	Originating city associated with the user's session.	✓	✓	–
Client IP	Client IP address associated with the user's session.	✓	✓	–
Client Time	The amount of time spent on the client side of the network during a hit download. For more information, see Tracking time spent on a hit and Understanding what is included in Client Time.	✓	–	–
Content Category	Content category for the hit content, such as HTML, Image, or Media. For a complete list of categories, see Understanding content categories and content error counts.	✓	–	–
Cookie	Cookie found in a hit's HTTP requests and responses.	✓	–	–

Table 66. Hit, session, and sequence details

Hit Detail	Description	Hit	Session	Sequence
Country	Originating country associated with the user's session.	✓	✓	–
Custom Fields	Custom fields are defined in the configuration and contain customized values.	✓	✓	✓
End-to-End Time	Equal to the sum of Back-End Time, Client Time, and Network Delay. For more information, see Tracking time spent on a hit.	✓	–	–
End-to-End Time Exceeds Threshold	Boolean that indicates if the value of End To End Time exceeded the defined threshold.	✓	–	–
Exception	Reports on Archiver exception conditions that occurred for a hit.	✓	–	–
Has Content Error	Boolean that indicates if this hit has content errors. A content error occurs when one type of content (such as an image) is requested, but a different content type is returned.	✓	–	–
Has Custom Error	Boolean that indicates if this hit has custom errors. A custom error occurs when a user-defined hit analyzer marks some content as an error.	✓	–	–
Has Navigation Timings	Boolean that indicates if this hit has Navigation API timings. Requires browser instrumentation. When true, the Archiver stores the navigation details listed in Navigation timing metrics.	✓	–	–
Has Page Data	Boolean that indicates if this hit is also a page. When true, the Archiver stores the details listed in Page details.	✓	–	–
Hit Analyzer Match	Number of times a hit analyzer matched the hit.	✓	✓	✓
HTTP Error	Indicates if an HTTP client or server error has occurred. Possible values are `client`, `server`, or an empty String (if no error occurred).	✓	–	–
Instrumentation Browser Error Count	Number of browser errors reported in this instrumentation hit.	✓	–	–
Instrumentation Custom Logging Count	Number of custom log messages reported in this instrumentation hit.	✓	–	–
Is Instrumentation Hit	Boolean that indicates if this hit was sent by client-side instrumentation.	✓	–	–
ISP	Originating ISP associated with the user's session.	✓	✓	
Metric Updates	List of custom metrics that were updated by this hit and their values at the time they were updated.	✓	✓	
Network Delay	Estimate of the amount of time it takes for a packet to traverse the network from client to server.	✓	–	–
Operating System	Operating system used by the client machine during the user's session.	✓	✓	–
Region	Originating region for the user's session.	✓	✓	–
Request Content	Indexed keywords of the HTTP request.	✓	–	–
Request Content Size	Size of the HTTP request content.	✓	–	–
Request Field	HTTP request parameters that appear in the requesting URL.	✓	–	–
Request Header	HTTP header in the request.	✓	–	–

Table 66. Hit, session, and sequence details

Hit Detail	Description	Hit	Session	Sequence
Request Method	Request method used, such as `GET` or `POST`.	✓	–	–
Request Path	Full path of the HTTP request.	✓	–	–
Request Referer	Value of the referer tag in the HTTP request.	✓	–	–
Request Total Bytes	Total amount of bytes in an HTTP request.	✓	–	–
Request URL	URL transmitted in an HTTP request.	✓	–	–
Response Code	HTTP response code generated by the server in response to an HTTP request.	✓	–	–
Response Contains Frameset Tag	Indicates if the HTTP response contains the HTML `<frameset>` tag.	✓	–	–
Response Contains HTML Tag	Indicates if the HTTP response contains the `<html>` tag.	✓	–	–
Response Content	Indexed keywords in the HTTP response.	✓	–	–
Response Content Size	Size of the HTTP response.	✓	–	–
Response Header	HTTP response header.	✓	–	–
Response HTML Title	Value of the HTML `<title>` tag found in the HTTP response.	✓	–	–
Response Total Bytes	Total amount of bytes in an HTTP response.	✓	–	–
Script Output	Value of a script calculation, which can be used to update metrics.	✓	✓	✓
Sequence Custom Field	Sequence-scoped custom fields updated by an analyzer when processing the hit.	–	–	✓
Sequence Duration	Length of the current sequence.	–	–	✓
Server IP	IP address of the server that responds to a client request.	✓	–	–
Server Port	TCP port used by the monitored application.	✓	–	–
Session Custom Field	Session-scoped custom fields updated by an analyzer when processing the hit.	✓	✓	✓
Session Duration	Length of the user's session.	–	✓	–
Session ID	Value of the session identifier (usually a cookie) that is used to uniquely identify a user's session.	✓	✓	✓
Sniffer ID)	Name of the sniffer that captured a hit.	✓	–	–
SOAP Adapter Path	URI that receives SOAP requests.	✓	–	–
SOAP Fault Actor	Value of the `<faultactor>` tag in the SOAP Fault element.	✓	–	–
SOAP Fault Code	Value of the `<faultcode>` tag in the SOAP Fault element.	✓	–	–
SOAP Fault Details	Value of the `<faultstring>` tag in the SOAP Fault element.	✓	–	–
SOAP Fault Subcode	Value of the `<faultsubcode>` tag in the SOAP Fault element.	✓	–	–

Table 66. Hit, session, and sequence details

Hit Detail	Description	Hit	Session	Sequence
SOAP Operation Name	Name of the SOAP operation (or method) found in an HTTP request.	✓	–	–
SOAP Web Service Name	Name of the SOAP web service associated to the requested SOAP operation.	✓	–	–
Stop Reason	Reason the user's sequence or session stopped.	–	✓	✓
Subnet	Originating subnet for the user's session.	✓	✓	–
Status	Status of the hit or session. The status observed in the web traffic can be changed by analyzers.	✓	✓	–
URL Scheme and Authority	The part of the URL that includes the scheme and authority (domain and port).	✓	–	–
Username	The login or account name that a user entered to gain access to a monitored application.	✓	✓	–
Virtual Server IP	IP address of the server extracted from the IP layer (in many cases, this is a VIP address).	✓	–	–
Web Site	Web site of the monitored application.	✓	–	–

Page details

When a hit is also a page, the following details are created and stored.

Table 67. Page details

Page Detail	Description
HTTP Version	Version number of HTTP.
Page Back-End Time	Sum of the Back-End Time values for all hits associated with the page.
Page Back-End Time Exceeds Threshold	Number of times that the value of Back End Time exceeded the defined threshold.
Page Client Time	Sum of the Client Time values for all hits associated with the page.
Page End-to-End-Time	Total time taken to download a complete page. For more information, see Tracking time spent on a page.
Page End-to-End Time Exceeds Threshold	Number of times that the value of End-to-End Time exceeded the defined threshold.
Page Hit Count	Total number of hits that occurred for a page.
Page Hit Redirect Count	Number of hits for the page that were redirected.
Page Hit Timeout Count	Number of hits for the page that timed out.
Page Incomplete Download	Boolean that indicates if a download was interrupted by the client before completion.
Page Network Delay	Sum of Network Delay values for all hits associated with the page.
Page Request Content Size	Sum of Request Content Size values for all hits associated with page.
Page Request Total Bytes	Sum of Request Total Bytes values for all hits associated with page.
Page Response Content Size	Sum of Response Content Size values for all hits associated with the page.
Page Response Total Bytes	Sum of Response Total Bytes values for all hits associated with the page.

Table 67. Page details

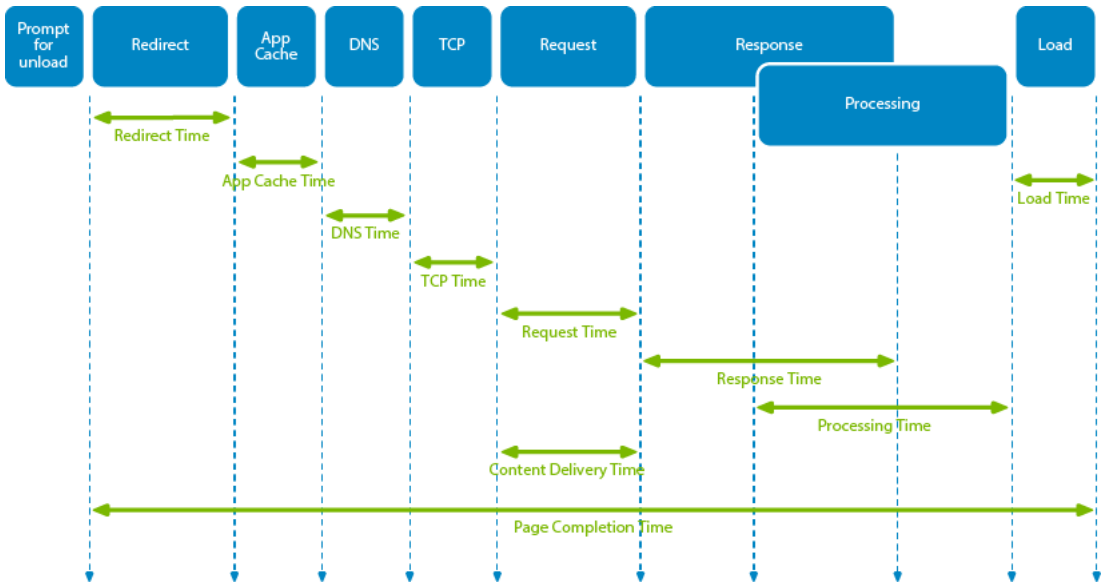
Page Detail	Description
Page TCP Connections Used Count	Number of TCP connections opened by the client to the server to retrieve page content.
Page Think Time	Amount of time a user spends on a downloaded page (reading and thinking about the content) before going to another page. The timer that tracks think time *starts* when a page is fully downloaded and *stops* when the next page is requested.

5

Navigation timing metrics

When you add instrumentation to your monitoring solution, Foglight APM uses the WC3 Navigation Timing API to collect client-side metrics from the browser. The following diagram shows the activities that make up a request for content and then shows how the Foglight APM metrics report on these activities. The metrics in bold have associated metrics for setting service level thresholds and reporting on how often those thresholds are exceeded.

Figure 1. Navigation timing metrics diagram



The following table defines the Foglight APM metrics.

Table 68. Navigation timing metrics details

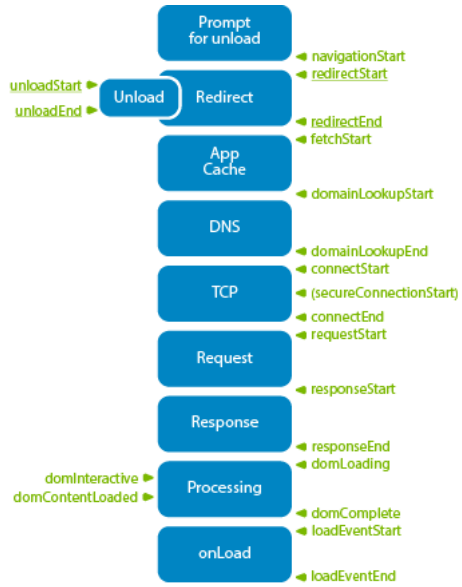
Navigation Detail	Description	API Attributes Calculation
App Cache Time	Amount of time taken by the user agent to check local application caches to see if the requested resource already exists on the client device.	`= domainLookupStart – fetchStart`
Content Delivery Time	Amount of time taken from the beginning of the request to the beginning of the response. For an example, see Tracking time spent on a hit using instrumentation.	`= responseStart – requestStart`
Content Delivery Time Exceeds Threshold	Number of times content delivery exceeded the specified threshold.	Foglight-derived metric
Content Delivery Time Exceeds Threshold Percent	Percentage of times content delivery exceeded the specified threshold.	Foglight-derived metric
Content Delivery Time Service Level	Maximum amount of time that should be required to deliver content.	User-specified
DNS Time	Amount of time taken by the user agent to perform a domain name lookup for the request.	`= domainLookupEnd – domainLookupStart`

Table 68. Navigation timing metrics details

Navigation Detail	Description	API Attributes Calculation
Load Time	Amount of time taken by the user agent from the beginning to the end of the load event. Generally, this would be the amount of time required for the browser to render the page.	`= loadEventEnd – loadEventStart`
Page Completion Time	Total time taken from the beginning of a navigation action (e.g., a click on a hyperlink) to the end of the load event when the page has been completely displayed in the browser. For an example, see Tracking time spent on a page using instrumentation.	`= loadEventEnd – navigationStart`
Page Completion Time Exceeds Threshold	Number of times the page completion times exceeded the specified threshold.	Foglight-derived metric
Page Completion Time Exceeds Threshold Percent	Percentage of times page completion times exceeded the specified threshold.	Foglight-derived metric
Page Completion Time Service Level	Maximum amount of time that should be required to complete a page load.	User-specified
Processing Time	Amount of time from the beginning of the response until the user agent sets the current document readiness state to “complete.” **NOTE:** Processing Time and Response Time can overlap, since the “domLoading” event can actually start before the “responseEnd” event, as the browser can start loading the DOM before the response is fully received. **NOTE:** Processing Time can be seen as an approximate measure of browser rendering time, but can include some time when the response is still being transmitted.	`= domComplete – domLoading`
Redirect Time	Amount of time taken from the start of the fetch initiated by a redirect until the full response of the last redirect has been received by the user agent.	`= redirectEnd – redirectStart`
Request Time	Amount of time taken from when the user agent initiates a request to the server until the first byte of the response is received.	`= responseStart – requestStart`
Response Time	Amount of time from the time the user agent receives the first byte of the response until the last byte of the response is received.	`= responseEnd – responseStart`
TCP Time	Amount of time taken by the user agent to establish a TCP connection to the server to retrieve the current document. This includes the time taken for an SSL handshake (if any).	`= connectEnd – connectStart`
Unload Time	Amount of time taken to unload the page in the user agent.	`= unloadEventEnd – unloadEventStart`

The calculations in the API Attributes Calculation column refer to the attributes shown in the following diagram. The attributes and diagram come from the *Navigation Timing W3C Recommendation 17 December 2012* located here http://www.w3.org/TR/2012/REC-navigation-timing-20121217/#sec-navigation-timing-interface.

Figure 2. API attributes calculation



When a hit analyzer captures metrics for an instrumented page, Foglight calculates the derived metrics and stores all the Foglight APM metrics as Foglight metrics (topology objects) so that the metrics can be used in top-level dashboards.

6

Understanding calculated data

Some details (and by extension, the standard metrics that require these details) are calculated based on other captured details. For example, the detail Hit End-to-End Time is calculated using three other details. This section explains how details are used to track time during a hit or after a page download, how calculated details are derived, and how details are updated based on client and server activities.

This section contains the following explanations:

- Tracking time spent on a hit
- Tracking time spent on a page
- Understanding what is included in Back End Time
- Understanding what is included in Client Time
- Estimating network delay
- Calculating access speed
- Tracking exceptions
- Calculating volume
- Summary of causes for SSL connection errors
- Understanding content categories and content error counts
- Tracking time spent outside the network

Tracking time spent on a hit

The following hit details track time spent on a hit:

- Hit Initial Response Time
- Hit Back End Time
- Hit Client Time
- Hit End-to-End Time
- Hit Network Delay

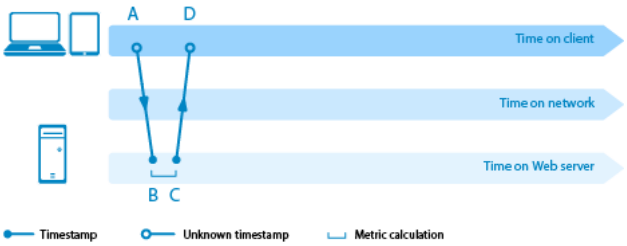
The following sections describe two scenarios. In the first scenario, the response content fits into a single network packet. The second scenario describes a more complex example where the size of the response content requires that it be transmitted to the client using multiple packets.

- Single-packet response to a hit
- Multiple-packet response to a hit

Single-packet response to a hit

In the following diagram, the horizontal arrows represent time passing on a client, network, and web server. The diagonal arrows show a request and a response crossing the network.

Figure 3. Single-packet response to a hit



The following table describes which hit details are updated at each timestamp.

Table 69. Hit details updated for single-packet response to a hit

Time	Activity	Hit Details
A	Client requests content from the server, for example GET `index.html`.	—
B	Server receives the request.	*Network Delay estimate*
C	Server sends a packet with the entire response content.	Initial Response Time = C – B
D	Client receives the response content.	*Network Delay estimate*

In this example, the Back End Time (total time spent on the server) is equal to the Initial Response Time. Because there is no client-side processing time, the value of Client Time is zero. Therefore, this hit's End-to-End Time can be expressed as follows:

Hit End-to-End Time = Hit Back End Time + (Hit Network Delay * 2)

To understand Network Delay, see Estimating network delay.

Multiple-packet response to a hit

When the content requested cannot fit in a single packet, the server sends the content using server bursts. Each burst of data consists of at least two data packets. Then the server needs to wait for an acknowledgement from the client between bursts. Once it receives an acknowledgement for the first burst, it can send the next burst of data.

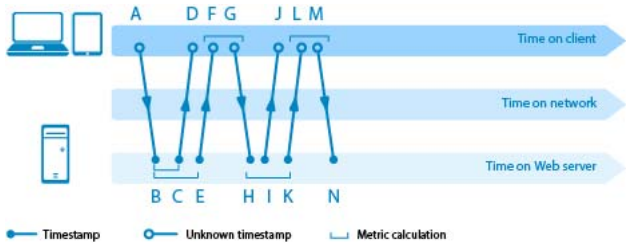
The TCP protocol limits how much data the server can send in a burst without receiving an acknowledgement from the client. This maximum amount of data the server can send is called the TCP window size. This window size is negotiated between the client and server when the TCP connection is established, and is used for congestion control on the network. (In the example used in the previous section, the TCP window size was 3000 bytes.)

As the server sends data, it reduces the TCP window size by the amount of data that was sent. As the client ACKs data, the server can increase its current window size by the amount of data ACKed. So long as the server continually receives ACKs from the client, its TCP window will remain "open," allowing it to continue to send data. However, if there is congestion or delays on the network, the server continues to send data until its window size is zero, which means the window is "closed." At that point, the server must stop sending data and wait for an ACK from the client.

Obviously, larger windows allow the server to burst more data at once, but also increase the chance for congestion which can lead to dropped packets and the server needing to resend data. Smaller windows decrease the chance for causing congestion but result in less efficient communication between client and server.

The following diagram shows a server response with two data bursts. The horizontal arrows represent time passing on a client, network, and web server. The diagonal arrows show requests and responses crossing the network from the client to a web server and back again.

Figure 4. Multi-packet response to a hit



The following table describes which hit details are updated.

Table 70. Hit details updated for multi-packet response to a hit

Time	Activity	Hit Details
A	Client requests content from the server.	—
B	Server receives the request.	*Network Delay estimate*
C	Server prepares a burst response and sends the first packet containing the hit status.	Initial Response Time = C – B
D	Client receives status.	—
E	Server sends the second packet in the burst, which contains the first part of the response content.	Back End Time = E – B
F	Client receives the response content.	
G	Client sends an acknowledgement packet.	
H	Server receives the acknowledgement.	Client Time = H – E
I	Server sends the next packet containing more response content.	
J	Client receives the response content.	
K	Server sends the final packet containing the last of the response content.	Back End Time = Back End Time + (K – H)
L	Client receives the response content.	
M	Client sends an acknowledgement packet.	
N	Server receives the acknowledgement. The hit is complete.	Client Time = Client Time + (N – K)

In this example, hit details are updated more than once. The new values are added to the existing value for the hit detail. Note that the Initial Response Time (C – B) is already included in Back End Time (D – B).

The Hit End-to-End Time can be expressed as follows:

Hit End-to-End Time = Final Hit Back End Time + Final Hit Client Time + Hit Network Delay

To understand Hit Network Delay, see Estimating network delay.

Tracking time spent on a page

A page consists of one or more hits, each of which is identified by a URL. A page download begins when the end user opens a new web page. The client sends a request for content. A web server starts sending back the response content to the client. The server may also contact another server to provide additional content. As the client receives content, it may open parallel connections and begin downloading some of the other hits embedded in the page. By downloading hits in parallel, the client maximizes its available access pipe and the end user gets the page content faster.

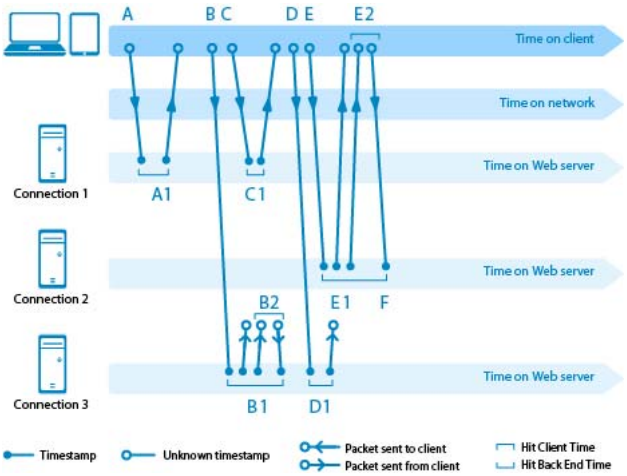
The following page details track time spent on a hit:

- Page Back End Time

- Page Client Time
- Page End-To-End Time (equals Page Back End Time plus Page Client Time)
- Page Network Delay

For example, consider a web page that contains four content elements. The client opens three connections to the server to retrieve content required for the page download. The client can be receive multiple responses from the server at once. The following diagram shows each of the client requests (labeled A through E). The Back End Time for each hit is shown in the web server arrows (labeled A1 through E1).

Figure 5. Tracking time spent on a page



The following table describes the workflow, where each letter represents the start of a new hit. A letter followed by the number 1 (for example, B1) represents Hit Back End Time for the hit. A letter with the number 2 (for example, B2) represents Hit Client Time for the hit. Page details are updated at the end of the page download.

Table 71. Tracking time spent on a page - workflow details

Time	Activity
A	Client requests `main.html` page from the server.
A1	Server receives the request and sends a response.
B	Client requests `frame1.html` from the server.
B1	Server receives the request and sends a burst response.
B2	Server sends the second packet in the burst, the client receives the packet, prepares and sends an acknowledgement, and the server receives the acknowledgement.
C	Client requests `style.css` content from the server.
C1	Server receives the request and sends a response. Note that the time span for C1 occurs entirely within the time span of B1.
D	Client requests `image1.gif` content from the server.
D1	Server receives the request and sends a response.
E	Client requests `news.asp` content from the server.
E1	Server receives the request and sends a burst response. Note that the time span for E1 partially overlaps the time span for D1.
E2	Server sends the second packet in the burst, the client receives the packet, prepares and sends an acknowledgement, and the server receives the acknowledgement.
F	Server receives the final acknowledgement. The page is complete.

In this example, Hit Back End Time C1 is entirely overlapped by B1, while E1 partially overlaps D2. There are no Hit Client Time overlaps. Therefore, the page metrics can be expressed as follows:

Page Back End Time = A1 + B1 + C1 + D1 + E1 – (C1 + E1 overlap)

Page Client Time = B2 + E2
Page End-to-End Time = Page Back End Time + Page Client Time + Page Network Delay

To understand Page Network Delay, see Estimating network delay.

Understanding what is included in Back End Time

The Back End Time metric includes the initial response time plus all the time spent by web servers processing HTTP requests sent from client browsers. Each hit has only one occurrence of initial request processing. The remaining server-side processing occurs as repeated data bursts and acknowledgement processing.

To process hits, web servers use a variety of resources such as databases, network bandwidth, system memory, and CPU. The web server must hold many of these resources until the required response is prepared and then transmitted back to the client. Therefore, time spent in the following processing activities are included in the Hit Back End Time metric:

- Parsing of the HTTP request by the web server to determine which page is being requested and with what parameters
- Directing the request to the code that supplies the page
- Accessing back-end resources such as database servers or storage area networks
- Preparing and formatting output parameters
- Generating HTML and formatting an HTTP response message

If your network includes a reverse proxy (either standalone or as part of a load balancer) and the Sniffer is installed in the recommended location in front of this device, the metric also includes the time spent on a request by the reverse proxy or load balancer. For more information, see “Understanding how load balancers and reverse proxies affect metrics” in the *Foglight APM Installation and Setup Guide*.

NOTE: Web servers may also query external network resources, such as a content delivery network. The time spent on these servers results in a partial page response.

The Page Back End Time calculation can be expressed as follows:
Page Back End Time = Sum of Hit Back End Times – Sum of overlapping Hit Back End Times

Some hits include a request, but not a response. For example, the end user may cause a TCP reset by clicking the browser’s Stop button. Or the server may be overloaded and unable to respond. For these commands, the Back End Time metric is undefined.

Understanding what is included in Client Time

Whenever a server responds with a data burst, Foglight calculates the Hit Client Time hit detail. The hit detail captures the time that a server spends waiting for a client acknowledgement packet. In particular, the Hit Client Time includes the time required for the following activities:

- travel time from the server to the client for the last packet of a data burst
- reassembly of packets on the client and any other client-side processing
- preparation of an acknowledgement packet
- travel time from the client to the server for the acknowledgement packet

The Page Client Time can be expressed as follows:

Page Client Time = Sum of Hit Client Times – Sum of overlapping Hit Client Times

Another way to think of Page Client Time is as a measure of all time not spent at the back end:
Page Client Time = Page End-to-End Time – Page Back End Time – Page Network Delay

Estimating network delay

Network delay is the time taken by the network to transfer the initial request for a page from the client to the server, or transfer the final acknowledgement packet from the client back to the server. Network delay cannot be measured directly without browser instrumentation. An estimate of Network Delay is valuable in the End-to-End Time calculations because it helps provide a more realistic value for the hit detail than is possible with only server-side processing values.

Foglight calculates an estimate for the Network Delay metric by adding equal halves of the following: the amount of time for packets to be sent, and the amount of time it takes for the TCP acknowledgement of the packet to be received. The time it takes to send data and receive an acknowledgment is often called the *round-trip time*. This estimate of network delay assumes the upstream and downstream speeds between the client and server are the same. If they are different, then this estimate is less accurate. Differing upstream and downstream speeds are often the case with DSL or cable modem access.

NOTE: The estimate of Network Delay may be affected by the presence of a reverse proxy (either standalone or as part of a load balancer) in your network. If the Sniffer is installed between the proxy and the web server farm, the round-trip time is calculated from the proxy to the Sniffer, rather than from the end user to the Sniffer. This results in Network Delay estimates that are a fraction of the actual round-trip time. For more information, see “Understanding how load balancers and reverse proxies affect metrics” in the *Foglight APM Installation and Setup Guide*.

Foglight uses a characteristic of the TCP protocol stack to establish the round-trip time. In a TCP connection, each packet that contains data must be acknowledged by means of another packet (ACK packet) from the receiving side. Another useful characteristic of the TCP protocol is that the receiving side is required to immediately send an acknowledgement for every two full data packets it receives. This sending of acknowledgements is how TCP controls data congestion.

By measuring the delay from when a data packet is sent until the acknowledgement is received, a passive monitor can calculate a round-trip time for data packet (i) as follows:

```
RTT(i) = Tend(i) – Tstart(i)
where
Tstart(i) = timestamp when the server sends data packet(i)
Tend(i) = timestamp when the server receives the acknowledgement packet from the
client
```

As round-trip times can vary throughout the life of a TCP connection, the samples for a round-tip time are averaged into a smoothed round-trip time estimate (SRTT). When a hit is completed, the current smoothed round-trip time is used to set the value of the Network Delay metric:
Hit Network Delay = Smoothed Round-trip Time / 2

A page may consist of multiple hits sent across multiple TCP connections. In this case, the Page Network Delay equals the average of the network delay of its component hits.
Page Network Delay = Sum of Hit Network Delays / Number of hits for the page

Calculating access speed

Access speed refers to the effective speed at which an end user accesses a web application or web site. Several factors influence the user’s access speed:

- Type of connection to the ISP (such as analog modem, DSL, cable modem, ISDN)
- TCP transmit and receive window sizes

- Network congestion

Many other characteristics of an end user's session are influenced by the user's access speed. In particular, a slower access speed may mean that the affected user experiences slower page loading times (End-to-End Times or Load Times) than that experienced by other users.

The Access Speed calculation differs depending on whether it is calculated from the client side (using instrumentation) or from the server side.

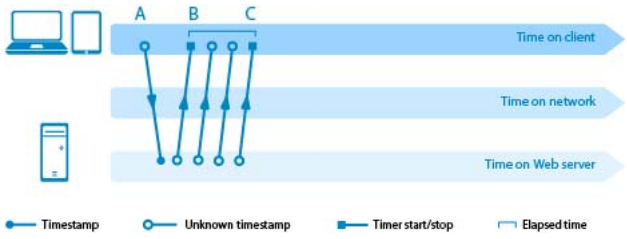
For more details, see the following topics:

- Calculating access speed from the client side
- Calculating access speed from the server side

Calculating access speed from the client side

The following diagram shows how the Access Detail is calculated when instrumentation is used to capture client-side timestamps.

Figure 6. Calculating access speed from the client side



The following table summarizes the workflow.

Table 72. Calculating access speed from the client side - workflow details

Time	Activity
A	Client requests the main page.
B	Client receives the page and runs the instrumentation script. **The script starts a timer.**
C	Client receives the last of the packets. **The script stops the timer.**

Foglight calculates the Access Speed by measuring the time required to transmit a block of data of a known size, and then dividing the data block size by the elapsed time.

Access Speed = Size of data block in bits / Elapsed time

Given a data block of 6,000 bytes (expressed as 48,000 bits) and an elapsed time of 0.5 seconds, the access speed for this user is:

Access Speed = 48,000 / 0.5 = 96,000 bits per second

Calculating access speed from the server side

When calculating access speed from the server side, Foglight takes steps to reduce the affects of network delay and network congestion on the values it uses to determine access speed. It does so by identifying the best candidate data blocks as follows:

- **Finding maximum-sized data blocks**. Timing the largest data blocks reduces the impact of network congestion and routing delays on the speed calculation. If small blocks were used, then the time to transmit the data block could be the same as the routing or congestion delays, which prevents the extraction of the actual transmit time from the total measured time. The system can determine the size of a packet during the client-server connection setup (three-way handshake).

- **Finding two sequential data blocks with no errors**. The data blocks need to be sequential to trigger a client acknowledgement packet. The packets must pass all checksum tests.
- **Verifying the client has successfully received the data blocks**. The server's sending of one or more data blocks does not necessarily mean the client will receive them. If the block is lost and must be resent, then the measured time cannot be used for access speed calculation.
- **Finding blocks where the client is not receiving data from other sources**. If other data is being transmitted to the client at the same time, that activity may interfere with the elapsed time. The system therefore tracks the state of all connections opened between the client and server, and ensures that a single connection is active when a sample is taken.

Foglight calculates the transmit time for the candidate data blocks using the same technique used for multiple packets. It starts with the time when the server received the client's acknowledgement packet and subtracts the time when the first packet was sent. This method always includes some round-trip time delay. Foglight therefore calculates an estimated round-trip time (using the same formula used for the Network Delay detail) and subtracts it from the transmit time to get the sample time. For more about acknowledgement packets and round-trip time, see Multiple-packet response to a hit and Estimating network delay.

Therefore, the access speed for one sample data block is calculated as follows:
Sample access speed = Size of packets in bits / (Transmit time – Round-trip time estimate)

Access speed samples are typically taken from different connections throughout the end user's session, allowing for a large number of sample points. The above calculation is performed whenever two valid maximum-sized TCP segments are transmitted during the session. An internal Round-Trip Time metric is kept up-to-date with current round-trip times.

Therefore, the Access Speed session detail is calculated as follows at the end of the user session:
Access Speed = Sum of sample access speeds / Number of samples

Tracking exceptions

When a hit does not contain all expected content, Foglight updates the Exception detail and related standard metrics. Exceptions can occur for the following reasons:

- **Client Reset**—the client sent a TCP reset. TCP resets are most often triggered by a real user clicking the browser's Back or Stop buttons or closing the browser.
- **Client Timeout**—the client stopped acknowledging receipt of data from server. Two common reasons for client timeouts are that the client loses power or the client becomes disconnected from the network.
- **Server Reset**—the server sent a TCP reset. Some reasons for server resets include unexpected behavior by the client or a server becomes overloaded.
- **Server Timeout**—the server stopped sending content on TCP connection. This is usually caused by some kind of server overload situation.
- **Chunked Transfer Error**—*chunking* refers to the process of breaking up content into multiple, small blocks. This rare exception type indicates that some content was missing due to a parsing error of some kind. If you see this type of exception, it may mean that an application error occurred or that some kind of network data loss occurred.
- **Incomplete Content—**the reason for the incomplete content is unknown.

Foglight does not add exceptions to the performance metrics or hit metrics because they would skew all of the metrics downward if they were included.

Calculating volume

Foglight reconstructs hits from a sequence of TCP/IP packets that are sent from the server to the client. TCP/IP packets are ordered and error checked to reconstruct the unique application data stream. Duplicate or invalid data

is excluded from this data stream. From the application data stream, the start and stop of each hit can be identified, which in turn enables Foglight to determine the size of the hit (in bytes). Foglight adds the size of requests to the Volume of Client Requests metric and the size of responses to the Volume of Server Responses metrics. The calculation of volume can be expressed as follows:

Volume = Volume of Client Requests + Volume of Server Responses

The following invalid bytes are not included in volume calculations:

- If application bytes are included in a packet that is considered invalid, due to checksum errors or field validation errors, then these bytes are excluded from the byte volume.
- If a hit spans multiple packets, it is possible that one or more of the packets is present due to packet drops or checksum errors. In this case, the duplicate application bytes are not included in the total byte volume.

Summary of causes for SSL connection errors

Following is a list of common causes for SSL errors:

- An SSL session resume was used to reestablish a secure connection but the Sniffer did not see the original key exchange and is therefore unable to decrypt the SSL connection.
- There is no SSL key configured for a server that has had a secure connection opened to it.
- There is a mismatch in supported SSL versions.
- The client and server negotiated an unsupported cipher suite.
- The client and server attempted to use a Diffie-Hellman key exchange, which is not supported by Foglight.
- An SSL message was incomplete likely due to a dropped packet or missing segment.
- An SSL protocol error such as a bad record type or bad handshake occurred. This is usually caused by dropped packets or missing segments.

Understanding content categories and content error counts

Content categories are based on file extensions, `content-type`, or other criteria. Foglight increments the Hits Captured metrics to count hits by various content category. When discrepancies arise between the content category and the actual response content that was served to the client, Foglight increments the Content Error metric.

The following table summarizes the content categories and describes the conditions (if any) that trigger content errors.

Table 73. Content categories and error conditions

Content Category	Description	Content Error Conditions
Favicon	File ends with *favicon.ico*	Response `content-type` does not start with `image/`
Download	File extensions: bat, csv, doc, eml, pdf, ppt, pwd, rtf, txt, wri, xls, zip, pkg, msi, gz, z, bz2, iso, and jar	None
Image	File extensions: bmp, gif, ico, jpe, jpg, jpeg, png, tga, tif, and tiff	Response `content-type` does not start with `image/`

Table 73. Content categories and error conditions

Content Category	Description	Content Error Conditions
JavaScript	File extension js	Response `content-type` does not contain javascript
Media	File extensions: aif, asf, avi, flv, m3u, mp3, mp4, mpa, mpeg, mp1, mpg, mpv, pls, rmi, ra, rm, rmj, rp, wav, wma, wmv, and xpl	None
StyleSheet	File extension is css	Response `content-type` does not start with `text/css`
HTML	`hasHtmlTag=true`	None
Instrumentation	`isInstrumentationHit=true`	Size > 1K
SOAP	Hit contains a SOAP operation name	`soapFaultCode` is not empty
HTML Fragment	Response `content-type` contains `html`	None
JSON	Response `content-type` contains `json`	None
XML	Response `content-type` contains `xml`	None
Other	None of the above	None

Tracking time spent outside the network

Sniffers and Archivers are exceptionally good at monitoring and capturing web traffic occurring within a physical or virtual network. While they can capture data from the in-house server-side perspective and infer some client-side metrics based on server-side timestamps, they cannot directly track time spent by a client browser or external web servers. To track time spent outside the network, you need to use Foglight APM instrumentation.

NOTE: Foglight APM instrumentation uses the Navigation Timing API to capture performance data. Current versions of popular browsers support the Navigation Timing API. To find out which earlier versions of the browsers support the API, search the Internet for “navigation timing api browser support.”

The instrumentation library is a small JavaScript library, packaged as a single file, and included in a web page for the purposes of monitoring page performance and application health from inside the browser. The library generates a separate set of performance metrics based entirely on client-side timestamps. Foglight does not mix client-side and server-side timestamps for calculation purposes. The instrumentation metrics are stored with the hit, alongside the usual Archiver details.

To learn how to add instrumentation to your monitoring solution, see “Capturing traffic outside of the network” in the *Foglight APM Administration and Configuration Guide*. For a list of navigation timing details collected through instrumentation, see Navigation timing metrics.

The following sections elaborate on two key performance metrics: Content Delivery Time and Page Completion Time.

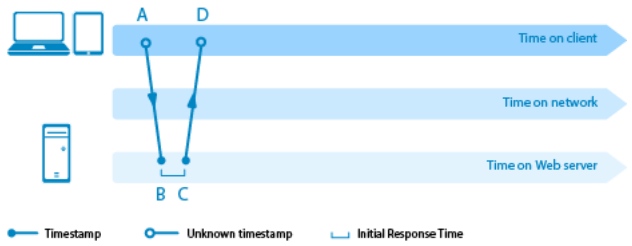
For more details, see the following topics:

- Tracking time spent on a hit using instrumentation
- Tracking time spent on a page using instrumentation

Tracking time spent on a hit using instrumentation

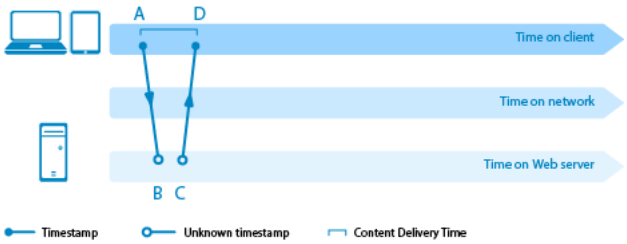
Recall that with Sniffer-only monitoring, the Initial Response Time reflects the time *from* when the server receives the request *to* when the server starts its response, as shown in the following diagram (**C–B**). The times from **A** to **B** and **C** to **D** are estimated. For more information, see Tracking time spent on a hit.

Figure 7. Tracking time spent on a hit using instrumentation (Initial Response Time)



After instrumenting a monitored web site or application, the instrumentation library tracks the client-side timestamps and reports overall Content Delivery Times, that is, the time it takes from the start of the client request to the moment when the client receives the first part of the response (**D–A**).

Figure 8. Tracking time spent on a hit using instrumentation (Content Delivery Time)



The following table summarizes the Content Delivery Time workflow.

Table 74. Content delivery Time - workflow details

Time	Activity
A	Client requests content. The script reads the timestamp associated with the Navigation Timing API attribute `requestStart`.
B	Server receives the request and prepares a response.
C	Server begins sending the response.
D	Client receives the beginning of the response. The script reads the timestamp associated with the Navigation Timing API attribute `responseStart`.

The instrumented hit time is calculated as follows:
Content Delivery Time = D – A

Or, in other words:
Content Delivery Time = `responseStart` **–** `requestStart`

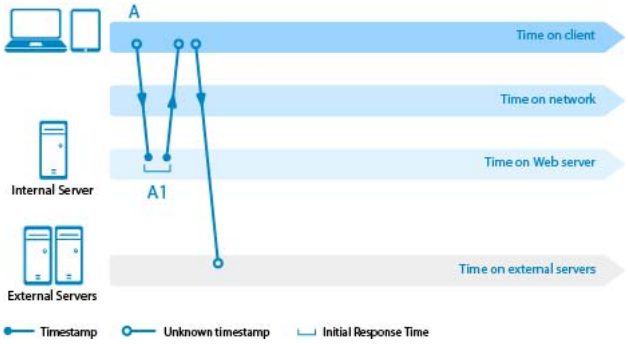
The actual network travel times from **A** to **B** and from **C** to **D** remain unknown. However, because Foglight stores both client-side Content Delivery Time and server-side Initial Response Time with the hit, it is possible to create a script to calculate the total network travel time by subtracting the Initial Response Time from the Content Delivery Time.

Tracking time spent on a page using instrumentation

While Sniffers capture data about calls out to external resources and back to internal web servers, they have little insight into the activities occurring on those external resources.

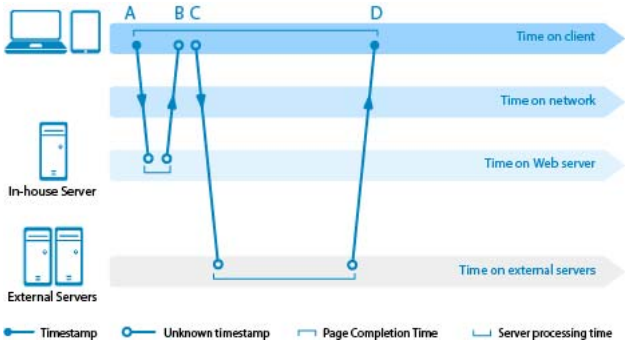
For example, a Sniffer cannot measure how long it takes for a content delivery network (CDN) server to respond to a request for content. The Sniffer can track only the time spent by an in-house web server responding to the initial page request (Initial Response Time), before the request is passed on to a CDN server to provide the page content—which does not reflect an end user's experience of how long it took for the page to load. In the following diagram, the Initial Response Time is marked as **A1**.

Figure 9. Tracking time spent on a page using instrumentation (Initial Response Time)



Instrumentation enables you to gather metrics about external activity through the use of client-side timestamps. In the following diagram, client-side timestamps at **A** and **D** provide the necessary information to calculate Page Completion Time.

Figure 10. Tracking time spent on a page using instrumentation (Content Delivery Time)



The following table summarizes the Page Completion Time workflow.

Table 75. Page Completion Time - workflow details

Time	Activity
A	Client requests `main.html` page from the server. The script reads the timestamp associated with the Navigation Timing API attribute `navigationStart`.
B	Client receives `main.html`.
C	Client requests content, such as `logo.gif`, from the external servers, and may request and receive other page content.
D	Client receives the final page response. The script reads the timestamp associated with the Navigation Timing API attribute `loadEventEnd`.

The instrumented page time is calculated as follows:

Page Completion Time = D – A

Or, in other words:

Page Completion Time = `loadEventEnd` **–** `navigationStart`

About Us

We are more than just a name

We are on a quest to make your information technology work harder for you. That is why we build community-driven software solutions that help you spend less time on IT administration and more time on business innovation. We help you modernize your data center, get you to the cloud quicker and provide the expertise, security and accessibility you need to grow your data-driven business. Combined with Quest's invitation to the global community to be a part of its innovation, and our firm commitment to ensuring customer satisfaction, we continue to deliver solutions that have a real impact on our customers today and leave a legacy we are proud of. We are challenging the status quo by transforming into a new software company. And as your partner, we work tirelessly to make sure your information technology is designed for you and by you. This is our mission, and we are in this together. Welcome to a new Quest. You are invited to Join the Innovation™.

Our brand, our vision. Together.

Our logo reflects our story: innovation, community and support. An important part of this story begins with the letter Q. It is a perfect circle, representing our commitment to technological precision and strength. The space in the Q itself symbolizes our need to add the missing piece—you—to the community, to the new Quest.

Contacting Quest

For sales or other inquiries, visit www.quest.com/contact.

Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.com.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request.
- View Knowledge Base articles.
- Sign up for product notifications.
- Download software and technical documentation.
- View how-to-videos.
- Engage in community discussions.
- Chat with support engineers online.
- View services to assist you with your product.