



Cloud Analytics Engine Data Learning Engine REST API Reference

Release

15.1X53



Modified: 2016-04-29

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Cloud Analytics Engine Data Learning Engine REST API Reference
15.1X53
Copyright © 2016, Juniper Networks, Inc.
All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <http://www.juniper.net/support/eula.html>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

	About the Documentation	vii
	Documentation and Release Notes	vii
	Supported Platforms	vii
	Using the Examples in This Manual	vii
	Merging a Full Example	viii
	Merging a Snippet	viii
	Documentation Conventions	ix
	Documentation Feedback	xi
	Requesting Technical Support	xi
	Self-Help Online Tools and Resources	xi
	Opening a Case with JTAC	xii
Part 1	Using the Data Learning Engine REST API	
Chapter 1	Introducing the Data Learning Engine REST API	3
	Data Learning Engine API Overview	3
	Data Learning Engine RESTful API Quick Reference	5
Chapter 2	Managing Compute Agent Information	7
	Subscribing to DLE Server-Sent Events	7
	GET /api/v1/events	7
	Server-Sent Events Data Learning Engine API Example	8
	Retrieving Compute Agents Information	9
	GET /api/v1/compute-agents	9
	Compute Agent Transfer Object	9
	Inquiry Compute Agent Data Learning Engine API Example	10
Chapter 3	Managing DLE Flow Path Traces	13
	Starting and Stopping a Flow Path Trace	13
	POST /api/v1/traces	13
	TraceConfig Transfer Object	14
	TunnelConfig Transfer Object	15
	Trace Creation Data Learning Engine API Example	16
	PUT /api/v1/traces/{traceId}	16
	Requesting Flow Path Trace Data	17
	GET /api/v1/traces	17
	GET /api/v1/traces/{traceId}	18
	Trace Data Transfer Object	18
	Trace Inquiries Data Learning Engine API Example	19
	Trace Data Inquiry Example For All Available Traces	19
	Trace Data Inquiry Example For a Specified Trace ID	21

	Requesting Summary Probe Data for a Trace	22
	GET /api/v1/traces/{traceId}/probes [?additionalInfo=hops]	22
	Probe Summary Transfer Object	23
	HopSummary Transfer Object	24
	Probe Inquiry Data Learning Engine API Example	24
	Probe Summary Inquiry Example	24
	Probe Summary Inquiry Example With Additional Hop Summary Information	27
	Requesting Detailed Probe Data for a Trace	30
	GET /api/v1/traces/{traceId}/probes/{probeId}	30
	Probe Detail Transfer Object	30
	Hop Transfer Object	31
	Hops Inquiries Data Learning Engine API Example	31
	Requesting Interface Statistics for a Trace	33
	GET /api/v1/traces/{traceId}/interface-stats	33
	Trace Interface Statistics Transfer Object	34
	Trace Interface Statistics Data Learning Engine API Example	36
Chapter 4	Managing DLE Streaming Data Subscription	41
	Subscribing to and Unsubscribing from DLE Flow Path Data Streaming	41
	POST /api/v1/subscription/subscribeAll	41
	DELETE /api/v1/subscription/unsubscribe	42
	Flow Path Response Data Object	42
Chapter 5	Retrieving Network Traffic Analysis Statistics	47
	Requesting NTA Top Application Statistics	47
	GET /api/v1/nta/top-applications	47
	NTA Top Stats Response Transfer Object	49
	NTA Top Applications Data Learning Engine API Example	49
	Requesting NTA Top Conversations Statistics	51
	GET /api/v1/nta/top-conversations	51
	NTA Top Stats Response Transfer Object	53
	NTA Top Conversations Data Learning Engine API Example	53
	Requesting NTA Application Flow Statistics	54
	GET /api/v1/nta/application-stats	54
	NTA Application Stats Samples Transfer Object	55
	NTA Application Stats Data Learning Engine API Example	56
	Requesting NTA Conversation Flow Statistics	57
	GET /api/v1/nta/conversation-stats	57
	NTA Conversation Stats Samples Transfer Object	58
	NTA Conversation Stats Data Learning Engine API Example	59

List of Tables

	About the Documentation	vii
	Table 1: Notice Icons	ix
	Table 2: Text and Syntax Conventions	ix
Part 1	Using the Data Learning Engine REST API	
Chapter 1	Introducing the Data Learning Engine REST API	3
	Table 3: Data Learning Engine RESTful API Quick-Reference	5
Chapter 2	Managing Compute Agent Information	7
	Table 4: Compute Agent Transfer Object	10
	Table 5: Compute Agent Network Interface Data Object	10
Chapter 3	Managing DLE Flow Path Traces	13
	Table 6: TraceConfig Transfer Object	14
	Table 7: TunnelConfig Transfer Object	15
	Table 8: Trace Data Transfer Object	18
	Table 9: Probe Summary Transfer Object	23
	Table 10: HopSummary Transfer Object	24
	Table 11: Probe Detail Transfer Object	30
	Table 12: Hop Transfer Object	31
	Table 13: Trace Interface Statistics Request Parameters	34
	Table 14: Trace Interface Statistics Transfer Object	34
	Table 15: Trace Interface Statistics Data Object	35
	Table 16: Trace Interface Traffic Statistics Samples Data Object	35
	Table 17: Trace Interface Packet Statistics Data Object	35
Chapter 4	Managing DLE Streaming Data Subscription	41
	Table 18: Flow Path Analytics Data JSON Response Object	43
	Table 19: Flow Path Analytics JSON Response Data: Path Object Elements	43
	Table 20: Flow Path Analytics JSON Response Data: Timestamp Object Elements	44
	Table 21: Flow Path Analytics JSON Response Data: Ingress Logical Interface Attribute Object Elements	44
	Table 22: Flow Path Analytics JSON Response Data: Egress Logical Interface Attribute Object Elements	44
	Table 23: Flow Path Analytics JSON Response Data: Layer 3 ECMP Object Elements	44
	Table 24: Flow Path Analytics JSON Response Data: Layer 2 ECMP List Object Elements	45
	Table 25: Flow Path Analytics JSON Response Data: Logical Interface Object Elements	45

	Table 26: Flow Path Analytics JSON Response Data: Interface Packet Statistics Object Elements	45
Chapter 5	Retrieving Network Traffic Analysis Statistics	47
	Table 27: NTA Top Applications Request Parameters	48
	Table 28: NTA Top Applications Response Data Object	48
	Table 29: NTA Top Stats Response Object Common Elements	49
	Table 30: NTA Top Conversations Request Parameters	52
	Table 31: NTA Top Conversations Response Data Object	52
	Table 32: NTA Top Stats Response Object Common Elements	53
	Table 33: NTA Application Stats Request Parameters	55
	Table 34: NTA Application Samples Transfer Object	56
	Table 35: NTA Conversation Stats Request Parameters	58
	Table 36: NTA Conversation Samples Transfer Object	59

About the Documentation

- Documentation and Release Notes on page vii
- Supported Platforms on page vii
- Using the Examples in This Manual on page vii
- Documentation Conventions on page ix
- Documentation Feedback on page xi
- Requesting Technical Support on page xi

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at <http://www.juniper.net/techpubs/>.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at <http://www.juniper.net/books>.

Supported Platforms

For the features described in this document, the following platforms are supported:

- [QFX Series](#)

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

```
system {
  scripts {
    commit {
      file ex-script.xml;
    }
  }
}
interfaces {
  fxp0 {
    disable;
    unit 0 {
      family inet {
        address 10.0.0.1/24;
      }
    }
  }
}
```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

```
[edit]
user@host# load merge /var/tmp/ex-script.conf
load complete
```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

```
commit {
  file ex-script-snippet.xml; }
```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:


```
[edit]
user@host# edit system scripts
[edit system scripts]
```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]
user@host# load merge relative /var/tmp/ex-script-snippet.conf
load complete
```

For more information about the **load** command, see the *CLI User Guide*.

Documentation Conventions

Table 1 on page ix defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page ix defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: user@host> configure

Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> show chassis alarms No alarms currently active
<i>Italic text like this</i>	<ul style="list-style-type: none">Introduces or emphasizes important new terms.Identifies guide names.Identifies RFC and Internet draft titles.	<ul style="list-style-type: none">A policy <i>term</i> is a named structure that defines match conditions and actions.<i>Junos OS CLI User Guide</i>RFC 1997, <i>BGP Communities Attribute</i>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# set system domain-name <i>domain-name</i>
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none">To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level.The console port is labeled CONSOLE.
< > (angle brackets)	Encloses optional keywords or variables.	stub <default-metric <i>metric</i>>;
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast multicast (<i>string1</i> <i>string2</i> <i>string3</i>)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[] (square brackets)	Encloses a variable for which you can substitute one or more values.	community name members [<i>community-ids</i>]
Indentation and braces ({ })	Identifies a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none">In the Logical Interfaces box, select All Interfaces.To cancel the configuration, click Cancel.

Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select Protocols>Ospf .

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at <http://www.juniper.net/techpubs/index.html>, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at <http://www.juniper.net/techpubs/feedback/>.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>

- Download the latest versions of software and review release notes:
<http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications:
<http://kb.juniper.net/InfoCenter/>
- Join and participate in the Juniper Networks Community Forum:
<http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <https://tools.juniper.net/SerialNumberEntitlementSearch/>

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at <http://www.juniper.net/cm/>.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see <http://www.juniper.net/support/requesting-support.html>.

PART 1

Using the Data Learning Engine REST API

- [Introducing the Data Learning Engine REST API on page 3](#)
- [Managing Compute Agent Information on page 7](#)
- [Managing DLE Flow Path Traces on page 13](#)
- [Managing DLE Streaming Data Subscription on page 41](#)
- [Retrieving Network Traffic Analysis Statistics on page 47](#)

CHAPTER 1

Introducing the Data Learning Engine REST API

- [Data Learning Engine API Overview on page 3](#)
- [Data Learning Engine RESTful API Quick Reference on page 5](#)

Data Learning Engine API Overview

The Data Learning Engine (DLE) API enables access to the network analytics data collection functions of the DLE component of Juniper Networks Cloud Analytics Engine. DLE acts as an independent network entity to gather available analytics data about devices in the network, and analyzes, normalizes, and stores it in a consistent way.

Using the DLE RESTful APIs, you can access the following DLE capabilities to request network analysis data collection and receive collected data:

- *Application Flow Path Tracing:*

Set up end-to-end and per-hop network tracing and analytics data collection for both underlay and overlay application flow paths.

DLE communicates with Cloud Analytics Engine Compute Agent (CA) components in the network to generate probes that emulate application traffic and collect flow data traces from the network devices in the flow path.

Using the DLE flow path APIs, you can:

- Get information about available CA components and choose which application flows to trace.
- Request to start or stop application flow data trace collection, including encapsulated flows for underlay-overlay correlation.
- Retrieve collected flow path trace data from DLE, including summary or detailed probe and hop data, and per-interface transmit and receive statistics.

- *Data Subscription Service:*

Subscribe to and automatically receive bulk flow path analytics data that DLE collects from network devices.

DLE supports a flow path analytics data subscription service that publishes network analytics data to subscribed clients as it is collected, enabling clients to consume that data in real-time and avoid the overhead of periodic polling. You can configure DLE to enable or disable the subscription service by setting property **net.juniper.analytics.dle.collection.subscription.status=true** (enabled) or **false** (disabled) in the DLE configuration file `/opt/cae/dle/conf/webserver.properties`. When subscribing, the client must be prepared to receive streamed UDP data that DLE publishes to the receiver port specified in the subscription request.

Using the DLE data subscription APIs, you can:

- Subscribe to and receive streamed CA flow path data collected by DLE for all currently active flows.
 - Unsubscribe and stop receiving streamed flow path data from DLE.
- *High-frequency Network Traffic Analysis (NTA) Statistics Sampling:*

Request NTA packet-based traffic sampling data collected by DLE.

For network devices managed by Juniper Networks Network Director, you can enable NTA data monitoring as described in [Configuring Network Traffic Analysis](#) in the Network Director documentation. Network devices stream the NTA data to DLE, and DLE processes and stores the samples as application or conversation flow statistics.

Using the DLE NTA APIs, you can:

- Retrieve cumulative and top NTA application and conversation sample results from DLE based on specified options.

See “[Data Learning Engine RESTful API Quick Reference](#)” on [page 5](#) for a complete list of DLE RESTful APIs with quick reference links to their usage details.

In the general context of the analytics data that can be retrieved using DLE APIs:

- A *flow* is represented by an N-tuple identifying the devices for which data is being collected.

Specifically, this is the following 5-tuple:

{ source IP address, destination IP address, transport protocol, application source port, application destination port }

- A *tunnel* represents an encapsulation type. The DLE API supports the VXLAN encapsulation type.

A tunnel or encapsulated flow is minimally identified by its assigned *VXLAN network identifier (vnid)*, *VTEP source IP address*, and *VTEP destination IP address*.

- An NTA *application* is identified by the pair of values *transport protocol* and *application port number*.
- An NTA *conversation* represents unidirectional information exchanged between two hosts. Bidirectional traffic is treated as two different conversations, one for each direction between the hosts. An NTA conversation is identified by the pair of values *source IP address* and *destination IP address*.
- DLE reports only TCP and UDP Layer 3 traffic.



NOTE: The default HTTP port for DLE is 8282, which can be changed using the DLE configuration file `/opt/cae/dle/conf/webserver.properties`.

Related Documentation

- *Cloud Analytics Engine Data Learning Engine REST API Reference*
- *Installing and Configuring Cloud Analytics Engine Data Learning Engine*
- *Cloud Analytics Engine Overview*

Data Learning Engine RESTful API Quick Reference

Table 3 on page 5 lists the DLE RESTful APIs available to access and manage DLE capabilities.

Table 3: Data Learning Engine RESTful API Quick-Reference

DLE Capability	RESTful APIs Available	Description
Managing Compute Agent and DLE Server Information	“GET /api/v1/events” on page 7	Request to subscribe to DLE Server-Sent Events (SSE) text-based streamed event data, to learn about Compute Agents DLE has discovered.
	“GET /api/v1/compute-agents” on page 9	Retrieve information about Compute Agents known to DLE, to learn about the application flows available to be traced.

Table 3: Data Learning Engine RESTful API Quick-Reference (*continued*)

DLE Capability	RESTful APIs Available	Description
Managing DLE Flow Path Traces	"POST /api/v1/traces" on page 13	Start an application flow path trace (non-overlay) or tunnel flow trace (overlay).
	"PUT /api/v1/traces/{traceld}" on page 16	Stop an active flow path trace.
	"GET /api/v1/traces" on page 17	Retrieve trace data for flow path traces managed by DLE, either all available traces or filter results based on trace status.
	"GET /api/v1/traces/{traceld}" on page 18	Retrieve detailed flow path trace data for a particular trace ID.
	"GET /api/v1/traces/{traceld}/probes [?additionalInfo=hops]" on page 22	Retrieve summary data from probes in a DLE flow path trace, including per-hop summary information if desired.
	"GET /api/v1/traces/{traceld}/probes/{probeId}" on page 30	Retrieve detailed probe and hops data for a particular probe in a DLE flow path trace.
	"GET /api/v1/traces/{traceld}/interface-stats" on page 33	Retrieve detailed device interface packet transfer statistics from a DLE flow path trace.
Managing DLE Data Subscriptions	"POST /api/v1/subscription/subscribeAll" on page 41	Subscribe to receive streaming bulk flow path analytics data for all flows known to DLE.
	"DELETE /api/v1/subscription/unsubscribe" on page 42	Unsubscribe from DLE bulk flow path analytics data subscription service.
Retrieving Network Traffic Analysis Statistics	"GET /api/v1/nta/top-applications" on page 47	Return NTA top conversation statistics collected by DLE for a network device.
	"GET /api/v1/nta/top-conversations" on page 51	Return NTA top conversation statistics collected by DLE for a network device.
	"GET /api/v1/nta/application-stats" on page 54	Return NTA application samples statistics collected by DLE for a network device.
	"GET /api/v1/nta/conversation-stats" on page 57	Return NTA conversation samples statistics collected by DLE for a network device.

Related Documentation

- [Data Learning Engine API Overview on page 3](#)
- [Cloud Analytics Engine Feature Guide for the QFX Series](#)

CHAPTER 2

Managing Compute Agent Information

- [Subscribing to DLE Server-Sent Events on page 7](#)
- [Retrieving Compute Agents Information on page 9](#)

Subscribing to DLE Server-Sent Events

Use the Cloud Analytics Engine DLE API as follows to subscribe to and receive Server-Sent Events notifications relevant to DLE clients.

- [GET /api/v1/events on page 7](#)
- [Server-Sent Events Data Learning Engine API Example on page 8](#)

GET /api/v1/events

Use this DLE API request to subscribe to DLE Server-Sent Events (SSE), which enables server-to-client streaming of text-based event data of interest to DLE component clients.

A DLE SSE event notification is delivered as a streaming HTTP response. The client initiates a regular HTTP request, the server responds with a custom **text/event-stream** content-type, and then the server streams UTF-8 encoded event data. When new events occur, data about the events is sent to subscribed clients as JSON objects.

- The event payload is formatted as defined by the particular event.
- The HTTP client or library used to run this API must be able to read responses incrementally as the data is available, rather than wait until the entire response has been received. Compatible clients or libraries include **curl/libcurl** and Apache HttpClient.
- The connection established in the subscription is kept alive by keep-alive messages (newlines with no other data) sent to the client periodically.

Event notifications supported by DLE:

- Compute Agent (CA) list updates: Notification that DLE has updated its list of known, active CA components.

Event data format:

- **caListUpdated**: Event name.
- **timeInMillis**: Timestamp of CA list update (epoch time in ms).

Sample event data in JSON format:

```
{"caListUpdated":{"timeInMillis":1418838003802}}
```

See DLE API [“GET /api/v1/compute-agents” on page 9](#) for details on how to retrieve information about available CA component information, which is required for activating flow traces using DLE API [“POST /api/v1/traces” on page 13](#)).

Request Parameters:

- None

Response Data:

- Streamed event data

For example, see [“Server-Sent Events Data Learning Engine API Example” on page 8](#) for CA list update event data (including keep-alive messages).

Server-Sent Events Data Learning Engine API Example

The following is a DLE API example of subscribing to and receiving Server-Sent Events (SSEs) data for Compute Agents list update events. See [“GET /api/v1/events” on page 7](#) for details on DLE Server-Sent Events subscription.

Request to subscribe to all available SSEs (**GET** method):

```
http://localhost:8282/api/vi/events
```

Streamed Event response data with keep-alive messages:

```
HTTP/1.1 200 OK
Date: Wed, 17 Dec 2014 17:38:37 GMT
Content-Type: text/event-stream
Transfer-Encoding: chunked
Server: Jetty(9.2.2.v20140723)
```

```
retry: 90000
data:
data:
```

```
retry: 90000
data:
data:
```

```
data: {"caListUpdated":{"timeInMillis":1418838003802}}
```

```
retry: 90000
data:
data:
```

```
retry: 90000
data:
data:
```

- Related Documentation**
- [Data Learning Engine API Overview on page 3](#)
 - *Cloud Analytics Engine Feature Guide for the QFX Series*

Retrieving Compute Agents Information

Use the Cloud Analytics Engine DLE API as follows to retrieve information about Compute Agent (CA) Cloud Analytics Engine components known to DLE.

- [GET /api/v1/compute-agents on page 9](#)
- [Compute Agent Transfer Object on page 9](#)
- [Inquiry Compute Agent Data Learning Engine API Example on page 10](#)

GET /api/v1/compute-agents

Get a list of all CA components known to DLE, with details about their configuration. DLE API clients use this information for activating flow traces, because CA signature information is part of the required parameters for starting a trace (see DLE API [“POST /api/v1/traces” on page 13](#)).

Clients can also subscribe to DLE event notifications and learn when DLE discovers new CA components, to know when to issue this request to get the new CA information. See DLE API [“GET /api/v1/events” on page 7](#) for details.

Request Format:

- **GET** method.

Request Parameters:

- None.

Response Data (in JSON format):

- **totalSize**: (Numeric) Number of CA components for which data is being returned.
- **caList**: List of CA components known to DLE, described in [“Compute Agent Transfer Object” on page 9](#).
- **uri**: Resource URI for this request.

See [“Inquiry Compute Agent Data Learning Engine API Example” on page 10](#) for a sample request and response.

Compute Agent Transfer Object

[Table 4 on page 10](#) describes the Compute Agent transfer object, the response data to a DLE API request for information about CA components in the network (see [“GET /api/v1/compute-agents” on page 9](#)).

Table 4: Compute Agent Transfer Object

Element Name	Description
calp	(String) CA IP address.
caPort	(String) CA management port.
hostIp	(String) CA host IP address.
hostname	(String) CA host name.
networkPorts	List of CA Network Interface data objects, defined in Table 5 on page 10 , which describe the application interfaces on the CA host with which application flows might be emulated.

[Table 5 on page 10](#) describes the CA Network Interface data object in Compute Agent response data.

Table 5: Compute Agent Network Interface Data Object

Element Name	Description
ipV4Addresses	List of IP addresses (strings) of application interfaces.
interfaceName	(String) Interface name.
macAddress	(String) Interface MAC address.

Inquiry Compute Agent Data Learning Engine API Example

The following is a DLE API example for retrieving information about CA components known to DLE. See [“GET /api/v1/compute-agents” on page 9](#). The CA information in the response is described in the [“Compute Agent Transfer Object” on page 9](#).

Request:

http://localhost:8282/api/v1/compute-agents

Response:

```
{
  "totalSize": 4,
  "caList": [
    {
      "calp": "10.94.201.11",
      "caPort": 8080,
      "hostIp": "10.94.201.11",
      "hostName": "host02-example-01",
      "networkPorts": []
    },
    {
      "calp": "192.168.55.102",
```

```

      "caPort": 8080,
      "hostIp": "192.168.55.102",
      "hostName": "host01-example-01",
      "networkPorts": []
    },
    {
      "calp": "192.168.55.178",
      "caPort": 8080,
      "hostIp": "192.168.55.178",
      "hostName": "host03-example-01",
      "networkPorts": [
        {
          "ipV4Addresses": [
            "10.0.0.2"
          ],
          "interfaceName": "eth1",
          "macAddress": "00:25:90:1a:9f:cb"
        }
      ]
    },
    {
      "calp": "10.94.201.11",
      "caPort": 8080,
      "hostIp": "10.94.201.11",
      "hostName": "host04.example.net",
      "networkPorts": [
        {
          "ipV4Addresses": [
            "10.0.0.2"
          ],
          "interfaceName": "eth1",
          "macAddress": "00:15:17:d6:d0:89"
        }
      ]
    }
  ],
  "uri": "/api/v1/compute-agents"
}

```

**Related
Documentation**

- [Data Learning Engine API Overview on page 3](#)
- *Cloud Analytics Engine Feature Guide for the QFX Series*

CHAPTER 3

Managing DLE Flow Path Traces

- [Starting and Stopping a Flow Path Trace on page 13](#)
- [Requesting Flow Path Trace Data on page 17](#)
- [Requesting Summary Probe Data for a Trace on page 22](#)
- [Requesting Detailed Probe Data for a Trace on page 30](#)
- [Requesting Interface Statistics for a Trace on page 33](#)

Starting and Stopping a Flow Path Trace

Use the DLE application flow path trace APIs as follows to:

- Start a trace of an application flow path, with or without encapsulation (VXLAN tunnel), by providing the corresponding request parameters.
- Subsequently stop an active trace for the specified trace ID.
- [POST /api/v1/traces on page 13](#)
- [TraceConfig Transfer Object on page 14](#)
- [TunnelConfig Transfer Object on page 15](#)
- [Trace Creation Data Learning Engine API Example on page 16](#)
- [PUT /api/v1/traces/{traceId} on page 16](#)

POST /api/v1/traces

Use this DLE API request to create a trace of a non-overlay or overlay (tunnel) flow data collection.

Request Format:

- **POST** method.
- **Content-Type:Application/JSON** for the request body.

Request Parameters:

- **tunnel**: (Boolean) Specify **tunnel=false** for non-overlay flow trace creation, or **tunnel=true** for overlay (tunnel) flow data collection. Default is **false** if this parameter is not specified.

Request Body (JSON format):

- [“TraceConfig Transfer Object” on page 14](#) for starting a flow trace, or [“TunnelConfig Transfer Object” on page 15](#) for starting a tunnel flow trace.

Response Data:

- If the collection was successfully created:
 - Unique trace ID of the created flow or tunnel trace.
 - Corresponding URI for retrieving trace data for that trace ID.

TraceConfig Transfer Object

[Table 6 on page 14](#) describes the TraceConfig transfer object that provides the parameters when instructing DLE to create a flow data trace collection (see [“POST /api/v1/traces” on page 13](#)). The [“TunnelConfig Transfer Object” on page 15](#) contains the same elements with additional tunnel identification elements for creating a tunnel trace collection.

Table 6: TraceConfig Transfer Object

Element Name	Data Type
source	(String) Source IPv4 address of the flow path emulation you want to start tracing. Required parameter.
destination	(String) Destination IPv4 address of the flow being traced. Required parameter.
destinationPort	(Numeric) Application destination port for the flow. Required parameter.
sourcePort	(Numeric) Application source port for the flow. Required parameter.
startTime	(Numeric) Scheduled time to start the trace (epoch time in ms), if a future time is specified. If not specified (default behavior), or the specified time is in the past, DLE starts the trace immediately upon receipt of the request.
endTime	(Numeric) Scheduled time to end the trace (epoch time in ms). If not specified, the default value is 0, which indicates the trace should run until it is manually stopped (see “PUT /api/v1/traces/{traceldj}” on page 16).

Table 6: TraceConfig Transfer Object (*continued*)

Element Name	Data Type
protocol	(String) Protocol type for the flow to be traced ("udp" or "tcp"). Required parameter.
hops	(Numeric) Number of network hops on the flow path for which to collect data. Data is gathered across all hops until the destination (<=255) if this parameter is not specified.
retries	(Numeric) Number of tries to attempt in case of a timeout on a given hop. If this parameter is not specified, only a single attempt is made to gather data for the hop.
collectionFreq	(Numeric) Sampling frequency in seconds for collecting network data. Default value if not specified is 1 second.
timeouts	(Numeric) Time to wait in ms for the device at any hop in the traced flow path to respond. Delayed responses that fall outside this timeout value are ignored.
checksum	(Boolean) Enable ("true") or disable ("false") checksum computation. Default value is "true" or enabled.
collectBandwidthIngress	(Boolean) If specified as "true", trace flow bandwidth at each hop ingress. Default value is "false" or disabled.
mirrorType	(String) Install mirroring of the indicated type for this flow. Supported mirror type values: "ERSPAN". Default if not specified is no mirror installation.
mirrorDirection	(String) Mirror direction. Supported values: "ingress".
mirrorAnalyzer	(String) Mirror analyzer IP address. No default value. If mirrorType is specified, this parameter must also be specified.

TunnelConfig Transfer Object

The TunnelConfig transfer object contains the same elements as the ["TraceConfig Transfer Object" on page 14](#) with the additional elements described in [Table 7 on page 15](#). The data in this object provides the tunnel details when instructing DLE to create an encapsulated flow data trace collection (see ["POST /api/v1/traces" on page 13](#)).

Table 7: TunnelConfig Transfer Object

Element Name	Description
All elements from "TraceConfig Transfer Object" on page 14 , plus the elements in this table.	
encapsulation	(String) Encapsulation type for tracing a tunnel flow. Supported values: "VXLAN". NOTE: For tunnel flow traces, DLE and CA assume VXLAN encapsulation by default, so this parameter is optional.

Table 7: TunnelConfig Transfer Object (*continued*)

Element Name	Description
sourceVTEP	(String) VTEP source IPv4 address. Required parameter.
destinationVTEP	(String) VTEP destination IPv4 address. Required parameter.
vnid	(Numeric) VXLAN network identifier. Required parameter.
sourceVTEPPort	(String) Source VTEP port number (represented as a string in this data object, rather than a numeric value).
destinationVTEPPort	(String) Destination VTEP port number (represented as a string in this data object, rather than a numeric value).
mcastGroupAddress	(String) Overlay network multicast group address. Required parameter.

Trace Creation Data Learning Engine API Example

The following is a DLE API example for creating a trace data collection. See [“POST /api/v1/traces” on page 13](#) for more details.

Request:

http://192.168.55.102:8282/api/v1/traces

- **POST** method
- **Content-Type** set to **Application/JSON**

Request body:

```
{
  "source": "192.0.2.8",
  "destination": "198.51.100.9",
  "protocol": "udp",
  "sourcePort": 3231,
  "destinationPort": 23451,
  "timeouts": 30,
  "collectionFreq": 20,
  "retries": 3,
  "hops": 4
}
```

PUT /api/v1/traces/{traceId}

Use this DLE API request to stop a DLE trace data collection for the specified trace ID.

Request Format:

- **PUT** method.

Request Parameters:

- None

Response Data:

- **HttpResponse** status indicating success or failure reason.

**Related
Documentation**

- [Data Learning Engine API Overview on page 3](#)
- [Cloud Analytics Engine Feature Guide for the QFX Series](#)

Requesting Flow Path Trace Data

Use the Cloud Analytics Engine DLE API as follows to retrieve analytics data from one or more flow path traces started by DLE API requests.

- [GET /api/v1/traces on page 17](#)
- [GET /api/v1/traces/{traceId} on page 18](#)
- [Trace Data Transfer Object on page 18](#)
- [Trace Inquiries Data Learning Engine API Example on page 19](#)

GET /api/v1/traces

Use this DLE API request to get trace operational data for traces managed by DLE.

Request Format:

- **GET** method.

Request Parameters:

- **status:** (String) "ACTIVE", "COMPLETED", or "SCHEDULED". Filter trace data results based on trace status, retrieving only traces that are still active (**ACTIVE**), that have reached the requested trace end time (**COMPLETED**), or have not yet started (**SCHEDULED**). If this parameter is omitted, return data for all available traces.

Response Data (JSON format):

- **totalSize:** (Numeric) Number of flow traces for which data is being returned.
- **flows:** List of detailed trace data for the traces specified in the request, described in the ["Trace Data Transfer Object" on page 18](#).

See ["Trace Inquiries Data Learning Engine API Example" on page 19](#) for sample trace data requests and responses.

GET /api/v1/traces/{traceId}

Use this DLE API request to get detailed trace data from DLE for a specified trace ID.

Request Format:

- **GET** method

Request Parameters:

- None.

Response Data:

- Detailed trace data described in the “[Trace Data Transfer Object](#)” on page 18, in JSON format, for the trace specified in the request

See “[Trace Inquiries Data Learning Engine API Example](#)” on page 19 for sample trace data requests and responses.

Trace Data Transfer Object

[Table 8 on page 18](#) describes the Trace Data transfer object, returned from DLE in response to a request for trace details for one or more traces. See “[GET /api/v1/traces](#)” on page 17 and “[GET /api/v1/traces/{traceId}](#)” on page 18.

Table 8: Trace Data Transfer Object

Element Name	Description
id	(String) Trace ID returned when the trace was requested (see “ POST /api/v1/traces ” on page 13).
uri	(String) URI for the trace data resource request related to this data.
sourceIp	(String) Source IP address of the flow path.
destinationIp	(String) Destination IP address of the flow path.
sourcePort	(String) Application source port of the flow.
destinationPort	(String) Application destination port of the flow.
protocol	(String) Protocol for the flow—“TCP” or “UDP”.
vnid	(String) Encapsulation VXLAN network identifier for a tunnel flow.
vmSourceMac	(String) VM application source MAC address for a tunnel flow.
vmDestinationMac	(String) VM application destination MAC address for a tunnel flow.
vmSourceIp	(String) VM application source IP address for a tunnel flow.

Table 8: Trace Data Transfer Object (*continued*)

Element Name	Description
vmDestinationIp	(String) VM application destination IP address for a tunnel flow.
vmSourcePort	(String) VM application source port for the flow.
vmDestinationPort	(String) VM application destination port for the flow.
averageE2ELatency	(Numeric) Average end-to-end latency in ms for the flow.
minE2ELatency	(Numeric) Maximum end-to-end latency in ms for the flow.
maxE2ELatency	(Numeric) Minimum end-to-end latency in ms for the flow.
startTime	(Numeric) Time (epoch time in ms) when trace was started or scheduled to be started (if non-active trace data is requested).
endTime	(Numeric) Time (epoch time in ms) when trace ended, or 0 if trace is still active or to be scheduled.
status	(Enumerated String) Run status of flow data trace: <ul style="list-style-type: none"> • "ACTIVE"—Trace still running. • "COMPLETED"—Finished at requested trace endTime. • "SCHEDULED"—Requested to start at a future startTime.
uriForProbeSummary	(String) URI for the probe summary resource request to retrieve information about the probes that generated this trace.
totalProbeEntries	(Numeric) Number of probe sampling instances (end-to-end flow path traversals) for the trace. This depends on the collection frequency (collectionFreq) specified when requesting to start the trace (see " POST /api/v1/traces " on page 13).

Trace Inquiries Data Learning Engine API Example

The following are DLE API examples for retrieving flow trace details for all available traces (see "[GET /api/v1/traces](#)" on page 17) or a specific trace ID (see "[GET /api/v1/traces/{traceId}](#)" on page 18). Requests use the **GET** method. The response data for these requests is in JSON format, and is defined in the "[Trace Data Transfer Object](#)" on page 18 for each trace.

- [Trace Data Inquiry Example For All Available Traces](#) on page 19
- [Trace Data Inquiry Example For a Specified Trace ID](#) on page 21

[Trace Data Inquiry Example For All Available Traces](#)

Request:

http://192.168.55.102:8282/api/v1/traces

Response:

```
{
  "totalSize": 18,
  "flows": [
    {
      "id":
"NTYuNS4zLjE0OjMxMTctNjYuNS4zLjE0OjQxMTctdWRwQDE0MjMwODgzMzQwklm36=",

      "sourceIp": "10.0.0.2",
      "destinationIp": "198.51.100.2",
      "destinationPort": 4040,
      "sourcePort": 6065,
      "protocol": "TCP",
      "vnid": 100,
      "vmSourceMac": "fa:16:3e:61:24:f2",
      "vmDestinationMac": "fa:16:3e:e7:ba:05",
      "vmSourceIp": "192.0.2.2",
      "vmDestinationIp": "198.51.100.1",
      "vmSourcePort": "4410",
      "vmDestinationPort": "4065",
      "averageE2ELatency": 7509,
      "minE2ELatency": 1863,
      "maxE2ELatency": 14437,
      "starttime": 1449159157000,
      "endtime": 1449159277000,
      "status": "COMPLETED",
      "uriForProbeSummary":

"/api/v1/traces/NTEuMC4wLjI6NDUwODQtMjE0OjMxMTctdWRwQDE0MjMwODgzMzQwklm36=/probes",

      "totalProbeEntries": 12,
      "activationFailureReason": "",
      "uri":
"/api/v1/traces/NTEuMC4wLjI6NDUwODQtMjE0OjMxMTctdWRwQDE0MjMwODgzMzQwklm36="

    },
    {
      "id":
"NTYuNS4zLjE0OjMxMTctNjYuNS4zLjE0OjQxMTctdWRwQDE0MjMwODgzMzQwklm40=",

      "sourceIp": "10.5.5.6",
      "destinationIp": "10.5.5.5",
      "destinationPort": 40,
      "sourcePort": 3065,
      "protocol": "TCP",
      "vnid": 100,
      "vmSourceMac": "fa:16:3e:e7:ba:05",
      "vmDestinationMac": "fa:16:3e:61:24:f2",
      "vmSourceIp": "198.51.100.1",
      "vmDestinationIp": "192.0.2.2",
      "vmSourcePort": "4065",
      "vmDestinationPort": "4410",
      "averageE2ELatency": 7509,
      "minE2ELatency": 1863,
      "maxE2ELatency": 14437,
```



```

        "starttime": 1446493551000,
        "endtime": 1446493671000,
        "status": "COMPLETED",
        "uriForProbeSummary":
"/api/v1/traces/NTEuMC4wLjI6NDUwODQ0MjEuMC4wLjI6NDMONSTlZHBAMTQyNjYyMzk3OTAwMA==/probes",

        "totalProbeEntries": 12,
        "activationFailureReason": "",
        "uri":

"/api/v1/traces/NTEuMC4wLjI6NDUwODQ0MjEuMC4wLjI6NDMONSTlZHBAMTQyNjYyMzk3OTAwMA=="

    },
    ...
    {
        "id":
"NTYuNS4zLjEyOjMxMTUtNjYuNS4zLjEyOjQxMTUtdWRwQDE0MjMwODgzMzI5OTk=",

        "sourceIp": "10.0.0.2",
        "destinationIp": "198.51.100.2",
        "destinationPort": 4040,
        "sourcePort": 6065,
        "protocol": "TCP",
        "vnid": 100,
        "vmSourceMac": "fa:16:3e:61:24:f2",
        "vmDestinationMac": "fa:16:3e:e7:ba:05",
        "vmSourceIp": "192.0.2.2",
        "vmDestinationIp": "198.51.100.1",
        "vmSourcePort": "4410",
        "vmDestinationPort": "4065",
        "averageE2ELatency": 8207,
        "minE2ELatency": 914,
        "maxE2ELatency": 15961,
        "starttime": 1446492480000,
        "endtime": 1446499680000,
        "status": "COMPLETED",
        "uriForProbeSummary":
"/api/v1/traces/NTEuMC4wLjI6NDUwODQ0MjEuMC4wLjI6NDMONSTlZHBAMTQyNjYyMzk3OTAwMA==/probes",

        "totalProbeEntries": 360,
        "activationFailureReason": "",
        "uri":

"/api/v1/traces/NTEuMC4wLjI6NDUwODQ0MjEuMC4wLjI6NDMONSTlZHBAMTQyNjYyMzk3OTAwMA=="

    }
  ],
  "uri": "/api/v1/traces"
}

```

Trace Data Inquiry Example For a Specified Trace ID

Request:

```

http://192.168.55.102:8282/api/v1/traces/
NTEuMC4wLjI6NDUwODQ0MjEuMC4wLjI6NDMONS11ZHBAMTQyNjYyMzk3OTAwMA==

```

Response:

```
{
  "id":
    "NTEuMC4wLjI6NDUwODQtMjEuMC4wLjI6NDMONSI1ZHBAMTQyNjYyMzk3OTAwMA==",

  "sourceIp": "10.0.0.2",
  "destinationIp": "198.51.100.2",
  "destinationPort": 6666,
  "sourcePort": 4040,
  "protocol": "TCP",
  "vmSourcePort": "None",
  "vmDestinationPort": "None",
  "averageE2ELatency": 7948,
  "minE2ELatency": 914,
  "maxE2ELatency": 15575,
  "starttime": 1429640044500,
  "endtime": 1429642044500,
  "status": "COMPLETED",
  "uriForProbeSummary":

  "/api/v1/traces/NTEuMC4wLjI6NDUwODQtMjEuMC4wLjI6NDMONSI1ZHBAMTQyNjYyMzk3OTAwMA==/probes",

  "totalProbeEntries": 100,
  "activationFailureReason": "",
  "uri":

  "/api/v1/traces/NTEuMC4wLjI6NDUwODQtMjEuMC4wLjI6NDMONSI1ZHBAMTQyNjYyMzk3OTAwMA=="
}
```

- Related Documentation**
- [Data Learning Engine API Overview on page 3](#)
 - [Cloud Analytics Engine Feature Guide for the QFX Series](#)

Requesting Summary Probe Data for a Trace

Use the Cloud Analytics Engine DLE API as follows to retrieve a summary of probe data for a DLE flow path trace, including additional per-hop summary data, if desired.

- [GET /api/v1/traces/{traceId}/probes \[?additionalInfo=hops \] on page 22](#)
- [Probe Summary Transfer Object on page 23](#)
- [HopSummary Transfer Object on page 24](#)
- [Probe Inquiry Data Learning Engine API Example on page 24](#)

GET /api/v1/traces/{traceId}/probes [?additionalInfo=hops]

Use this DLE API request to get probe summary data from DLE for the specified trace ID.

Request Format:

- **GET** method

Request Parameters:

- **None**: Instructs DLE to return only probe summary data.
- **additionalInfo**: Instructs DLE to return probe summary data and the specified additional data. Values supported: "hops"—Specifying **additionalInfo=hops** returns hop summary data for each probe in addition to probe summary data.

Response Data (JSON format):

- **id**: (String) Trace ID of the trace for which probe summary information was requested
- **totalSize**: (Numeric) Total number of probes for which results are being returned
- **probes**: List of probe summary information for each probe, described in ["Probe Summary Transfer Object" on page 23](#). The list includes hop summary information (described in ["HopSummary Transfer Object" on page 24](#)) if the request included parameter "additionalInfo=hops".

See ["Probe Inquiry Data Learning Engine API Example" on page 24](#) for a sample probe summary data request and response.

Probe Summary Transfer Object

[Table 9 on page 23](#) describes the Probe Summary transfer object, returned in JSON format in response to a request for probe summary data for a particular trace managed by DLE (see ["GET /api/v1/traces/{traceId}/probes \[?additionalInfo=hops\]" on page 22](#)).

Table 9: Probe Summary Transfer Object

Element Name	Description
differentPath	(Boolean) If "true", indicates this probe followed a different path (due to routing differences) than previous probes in the flow path for this trace. If "false", this probe's path matches that of the previous probe.
timestamp	(Numeric) Timestamp (epoch time in ms) when this probe started on its path.
maxHopLatency	(Numeric) Maximum hop latency in ms among all the hops recorded in this probe's flow path.
uri	(String) URI for the probe summary data resource request related to this data.
hopCount	(Numeric) Number of hops in this probe's flow path, and the size of the list of HopSummary objects in the hops element (described next).

Table 9: Probe Summary Transfer Object (*continued*)

Element Name	Description
hops	List of HopSummary transfer objects, which contain summary data for each hop in the probe's path, defined in the "HopSummary Transfer Object" on page 24. This data is included in the response if the request includes the additionalInfo=hops option.

HopSummary Transfer Object

Table 10 on page 24 describes the HopSummary transfer object, which is included in DLE API probe summary response data (see "GET /api/v1/traces/{traceId}/probes [?additionalInfo=hops]" on page 22 and "Probe Summary Transfer Object" on page 23) when the request specifies the **additionalInfo=hops** parameter.

Table 10: HopSummary Transfer Object

Element Name	Description
deviceSerial	(String) Device serial number for this hop.
latency	(Numeric) Latency in ms for this hop in the flow path.

Probe Inquiry Data Learning Engine API Example

The following are DLE API examples for retrieving probe summary data for a particular trace ID with and without additional hops summary information. See "GET /api/v1/traces/{traceId}/probes [?additionalInfo=hops]" on page 22. Requests use the **GET** method. The response data is in JSON format, and is described in "Probe Summary Transfer Object" on page 23 and "HopSummary Transfer Object" on page 24.

- [Probe Summary Inquiry Example on page 24](#)
- [Probe Summary Inquiry Example With Additional Hop Summary Information on page 27](#)

Probe Summary Inquiry Example

Request:

http://192.168.55.102:8282/api/v1/traces/NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQ1NTE=/probes

Response:

```
{
  "id":
    "NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQ1NTE=",
  "totalSize": 60,
  "probes": [
    {
      "id":
```

```

"NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM00",

    "differentPath": false,
    "timestamp": 1443561761500,
    "maxHopLatency": 5713,
    "hopCount": 3,
    "errorCode": "-1",
    "uri":
"/api/v1/traces/192.0.2.2-10.5.5.5:4410_40-198.51.100.1-10.5.5.4:4065_3065-tcp/probes/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM00"

    },
    {
        "id":
"NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM01",

        "differentPath": false,
        "timestamp": 1443561762500,
        "maxHopLatency": 3696,
        "hopCount": 3,
        "errorCode": "-1",
        "uri":
"/api/v1/traces/192.0.2.2-10.5.5.5:4410_40-198.51.100.1-10.5.5.4:4065_3065-tcp/probes/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM01"

        },
        {
            "id":
"NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM02",

            "differentPath": false,
            "timestamp": 1443561763500,
            "maxHopLatency": 9627,
            "hopCount": 3,
            "errorCode": "-1",
            "uri":
"/api/v1/traces/192.0.2.2-10.5.5.5:4410_40-198.51.100.1-10.5.5.4:4065_3065-tcp/probes/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM02"

            },
            {
                "id":
"NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM03",

                "differentPath": false,
                "timestamp": 1443561764500,
                "maxHopLatency": 5170,
                "hopCount": 3,
                "errorCode": "-1",
                "uri":
"/api/v1/traces/192.0.2.2-10.5.5.5:4410_40-198.51.100.1-10.5.5.4:4065_3065-tcp/probes/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM03"

                },
                {
                    "id":

```

```

"NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM04",

    "differentPath": false,
    "timestamp": 1443561765500,
    "maxHopLatency": 10572,
    "hopCount": 3,
    "errorCode": "-1",
    "uri":
"/api/v1/traces/192.0.2.2-10.5.5.5:4410_40-198.51.100.1-10.5.5.4:4065_3065-tcp/probes/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM04"

  },
  {
    "id":
"NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM05",

    "differentPath": false,
    "timestamp": 1443561766500,
    "maxHopLatency": 10541,
    "hopCount": 3,
    "errorCode": "-1",
    "uri":
"/api/v1/traces/192.0.2.2-10.5.5.5:4410_40-198.51.100.1-10.5.5.4:4065_3065-tcp/probes/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM05"

  },
  {
    "id":
"NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM06",

    "differentPath": false,
    "timestamp": 1443561767500,
    "maxHopLatency": 14437,
    "hopCount": 3,
    "errorCode": "-1",
    "uri":
"/api/v1/traces/192.0.2.2-10.5.5.5:4410_40-198.51.100.1-10.5.5.4:4065_3065-tcp/probes/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM06"

  },
  ...
  {
    "id":
"NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM3b",

    "differentPath": false,
    "timestamp": 1443561820500,
    "maxHopLatency": 15555,
    "hopCount": 3,
    "errorCode": "-1",
    "uri":
"/api/v1/traces/192.0.2.2-10.5.5.5:4410_40-198.51.100.1-10.5.5.4:4065_3065-tcp/probes/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM3b"

  }
}

```

```

    ],
    "uri":
"/api/v1/traces/NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQ1NTE=/probes"
}

```

Probe Summary Inquiry Example With Additional Hop Summary Information

Request:

```

http://192.168.55.102:8282/api/v1/traces/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQ1NTE=/
probes?additionalInfo=hops

```

Response:

```

{
  "id":
"NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQ1NTE=",
  "totalSize": 60,
  "probes": [
    {
      "id":
"NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQ1NTE=MGMMGM00",
      "differentPath": false,
      "timestamp": 1443561762500,
      "hops": [
        {
          "deviceSerial": "TA3714140387",
          "latency": 6013
        },
        {
          "deviceSerial": "TA3714141203",
          "latency": 5813
        },
        {
          "deviceSerial": "TA3714141204",
          "latency": 5713
        }
      ],
      "hopCount": 3,
      "errorCode": -1,
      "uri":
"/api/v1/traces/192.0.2.2-10.5.5.4:4410_40-198.51.100.1-10.5.5.4:4065_3065-tcp/probes/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQ1NTE=MGMMGM00"
    },
    {
      "id":
"NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQ1NTE=MGMMGM01",
      "differentPath": false,
      "timestamp": 1443561763500,
      "hops": [
        {

```

```

        "deviceSerial": "TA3714140387",
        "latency": 3996
    },
    {
        "deviceSerial": "TA3714141203",
        "latency": 3796
    },
    {
        "deviceSerial": "TA3714141204",
        "latency": 3696
    }
],
"hopCount": 3,
"errorCode": -1,
"uri":
"/api/v1/traces/192.0.2.2-10.5.5.5:4410_40-198.51.100.1-10.5.5.4:4065_3065-tcp/probes/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM01"
},
{
    "id":
"NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM02",

    "differentPath": false,
    "timestamp": 1443561764500,
    "hops": [
        {
            "deviceSerial": "TA3714140387",
            "latency": 9927
        },
        {
            "deviceSerial": "TA3714141203",
            "latency": 9727
        },
        {
            "deviceSerial": "TA3714141204",
            "latency": 9627
        }
    ],
    "hopCount": 3,
    "errorCode": -1,
    "uri":
"/api/v1/traces/192.0.2.2-10.5.5.5:4410_40-198.51.100.1-10.5.5.4:4065_3065-tcp/probes/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM02"
},
{
    "id":
"NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMMGM03",

    "differentPath": false,
    "timestamp": 1443561765500,
    "hops": [
        {
            "deviceSerial": "TA3714140387",
            "latency": 5470
        }
    ]
}

```



```

    },
    {
      "deviceSerial": "TA3714141203",
      "latency": 5270
    },
    {
      "deviceSerial": "TA3714141204",
      "latency": 5170
    }
  ],
  "hopCount": 3,
  "errorCode": -1,
  "uri":
"/api/v1/traces/192.0.2.2-10.5.5.5:4410_40-198.51.100.1-10.5.5.4:4065_3065-tcp/probes/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMGMGM03"

},
...
{
  "id":
"/api/v1/traces/192.0.2.2-10.5.5.5:4410_40-198.51.100.1-10.5.5.4:4065_3065-tcp/probes/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMGMGM3b",

  "differentPath": false,
  "timestamp": 1443561821500,
  "hops": [
    {
      "deviceSerial": "TA3714140387",
      "latency": 15855
    },
    {
      "deviceSerial": "TA3714141203",
      "latency": 15655
    },
    {
      "deviceSerial": "TA3714141204",
      "latency": 15555
    }
  ],
  "hopCount": 3,
  "errorCode": -1,
  "uri":
"/api/v1/traces/192.0.2.2-10.5.5.5:4410_40-198.51.100.1-10.5.5.4:4065_3065-tcp/probes/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=MGMGMGM3b"

}
],
"uri":
"/api/v1/traces/NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQlNTE=/probes"
}

```

**Related
Documentation**

- [Data Learning Engine API Overview on page 3](#)
- [Cloud Analytics Engine Feature Guide for the QFX Series](#)

Requesting Detailed Probe Data for a Trace

Use the Cloud Analytics Engine DLE API as follows to retrieve detailed probe data for a DLE flow path trace, including per-hop details.

- [GET /api/v1/traces/{traceId}/probes/{probeId}](#) on page 30
- [Probe Detail Transfer Object](#) on page 30
- [Hop Transfer Object](#) on page 31
- [Hops Inquiries Data Learning Engine API Example](#) on page 31

GET /api/v1/traces/{traceId}/probes/{probeId}

Use this DLE API request to get detailed probe data from DLE for the specified probe ID and the specified trace ID.

Request Format:

- **GET** method

Request Parameters:

- None

Response Data:

- [“Probe Detail Transfer Object”](#) on page 30, in JSON format, which also includes details about each hop in the flow path (see [“Hop Transfer Object”](#) on page 31).

See [“Hops Inquiries Data Learning Engine API Example”](#) on page 31 for a sample probe details request and response.

Probe Detail Transfer Object

[Table 11 on page 30](#) describes the Probe Detail transfer object, returned in JSON format in response to a request for probe details for a specific probe of a trace managed by DLE (see [“GET /api/v1/traces/{traceId}/probes/{probeId}”](#) on page 30).

Table 11: Probe Detail Transfer Object

Element Name	Description
id	(String) Probe ID. Note that the trace ID is a part of the probe ID.
differentPath	(Boolean) If “true”, indicates this probe followed a different path (due to routing differences) than previous probes in the flow path for this trace. If “false”, this probe’s path matches that of the previous probe.
timestamp	(Numeric) Timestamp (epoch time in ms) when this probe started on its path.

Table 11: Probe Detail Transfer Object (*continued*)

Element Name	Description
maxHopLatency	(Numeric) Maximum hop latency in ms among all the hops recorded in this probe's flow path.
uri	(String) URI for the probe detail resource request related to this data.
hopCount	(Numeric) Number of hops in this probe's flow path (number of objects in the hops list element).
hops	List of Hop transfer objects, which contain detailed hop and device data for the hops in the probe's path, described in " Hop Transfer Object " on page 31.

Hop Transfer Object

Table 12 on page 31 describes the Hop transfer object, which provides detailed hop and device information for the hops traversed by a specific probe for a trace managed by DLE. This information is included in the response to a probe detail API request (see "[GET /api/v1/traces/{traceId}/probes/{probeId}](#)" on page 30) as part of "[Probe Detail Transfer Object](#)" on page 30 for each hop traversed by the probe.

Table 12: Hop Transfer Object

Element Name	Description
deviceSerial	(String) Device serial number for this hop.
deviceName	(String) Device name of the device at this hop.
latency	(Numeric) Latency in ms for this hop in the flow path.
cpuUtilization	(Numeric) CPU utilization of device at this hop.
memUtilization	(Numeric) Memory utilization of device at this hop.
ingressInterface	(String) Ingress interface name of the hop.
egressList	List of egress interface name strings for the hop, in the format {"egressName": "<interface name>",...}.

Hops Inquiries Data Learning Engine API Example

The following is a DLE API example for retrieving detailed probe data for a specific probe ID. See "[GET /api/v1/traces/{traceId}/probes/{probeId}](#)" on page 30, "[Probe Detail Transfer Object](#)" on page 30, and "[Hop Transfer Object](#)" on page 31. Requests use the GET method, and response data is in JSON format.

Request:

```
http://192.168.55.102:8282/api/v1/traces/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQ1NTE=/probes/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQ1NTE=MGMMGM00
```

Response:

```
{
  "id":

  "NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQ1NTE=MGMMGM00",

  "timestamp": 1443561761500,
  "maxHopLatency": 6013,
  "differentPath": false,
  "hopCount": 3,
  "hops": [
    {
      "deviceSerial": "TA3714140387",
      "deviceName": "host-example-03",
      "latency": 6013,
      "cpuUtilization": 11,
      "memUtilization": 21,
      "ingressInterface": "ge-0/0/8.0",
      "egressList": [
        {
          "egressName": "et-0/0/48.0"
        }
      ]
    },
    {
      "deviceSerial": "TA3714141203",
      "deviceName": "host--example-04",
      "latency": 5813,
      "cpuUtilization": 10,
      "memUtilization": 22,
      "ingressInterface": "et-0/0/50.0",
      "egressList": [
        {
          "egressName": "et-0/0/48.0"
        }
      ]
    },
    {
      "deviceSerial": "TA3714141204",
      "deviceName": "host-example-01",
      "latency": 5713,
      "cpuUtilization": 10,
      "memUtilization": 20,
      "ingressInterface": "ge-0/0/9.0",
      "egressList": [
        {
          "egressName": "et-0/0/48.0"
        }
      ]
    }
  ]
}
```

```
]
}
```

**Related
Documentation**

- [Data Learning Engine API Overview on page 3](#)
- *Cloud Analytics Engine Feature Guide for the QFX Series*

Requesting Interface Statistics for a Trace

Use the Cloud Analytics Engine DLE API as follows to retrieve detailed device interface packet transmit and receive statistics for a DLE flow path trace.

- [GET /api/v1/traces/{traceId}/interface-stats on page 33](#)
- [Trace Interface Statistics Transfer Object on page 34](#)
- [Trace Interface Statistics Data Learning Engine API Example on page 36](#)

GET /api/v1/traces/{traceId}/interface-stats

Use this DLE API request to get detailed interface statistics samples collected by DLE for the specified trace ID.

Request Format:

- **GET** method

Request Parameters:

- [Table 13 on page 34](#) describes the request parameters, including the devices to sample.

Table 13: Trace Interface Statistics Request Parameters

Request Parameter	Description
device-serials	(String) Comma-separated list of device serial number strings in the trace's flow path from which to retrieve interface statistics.
interface-name	(String) Device interface name.
start-time	(Numeric) Starting time (epoch time in ms) from which to retrieve interface statistics samples.
end-time	(Numeric) Ending time (epoch time in ms) up until which to retrieve interface statistics samples.
trace-id	(String) Trace ID from which to retrieve statistics for the specified interface.

Response Data:

- Detailed interface statistics data samples for the specified trace and interface(s) in the requested interval, described in "[Trace Interface Statistics Transfer Object](#)" on [page 34](#).

See "[Trace Interface Statistics Data Learning Engine API Example](#)" on [page 36](#) for a sample trace interface statistics request and response.

Trace Interface Statistics Transfer Object

[Table 14 on page 34](#) describes the Trace Interface Statistics transfer object, which provides detailed interface statistics from the devices in the flow path for a particular trace managed by DLE, in JSON format. See "[GET /api/v1/traces/{traceId}/interface-stats](#)" on [page 33](#).

Table 14: Trace Interface Statistics Transfer Object

Element Name	Description
id	(String) Trace ID.
totalSize	(Numeric) Number of interfaces for which statistics are returned in this response in the traceStats list element.
traceStats	List of Trace Interface Statistics for the requested device(s), described in Table 15 on page 35 .

Table 14: Trace Interface Statistics Transfer Object (*continued*)

Element Name	Description
uri	(String) URI of the API request for this response data.

[Table 15 on page 35](#) describes the Trace Interface Statistics data elements.

Table 15: Trace Interface Statistics Data Object

Element Name	Description
deviceSerialNumber	(String) Serial number of the device.
iflCount	(Numeric) Number of iflTrafficRates list elements in this response.
iflTrafficRates	List of Trace Interface Traffic Statistics Samples data objects for the requested device(s) and interface(s), described in Table 16 on page 35 , which includes receive and transmit traffic statistics samples.

[Table 16 on page 35](#) describes the Trace Interface Traffic Statistics Samples data elements.

Table 16: Trace Interface Traffic Statistics Samples Data Object

Element Name	Description
interfaceName	(String) Interface name.
sampleCount	(Numeric) Number of samples list elements in this response.
samples	List of Trace Interface Packet Statistics data objects, described in Table 17 on page 35 .

[Table 17 on page 35](#) describes the Trace Interface Packet Statistics data elements, representing packet transmit and receive statistics samples.

Table 17: Trace Interface Packet Statistics Data Object

Element Name	Description
timestamp	Timestamp (epoch time in ms) of this sample in the requested sampling period.
tx_Pkts	(Numeric) Total transmitted (Tx) packets on the interface.
rx_Pkts	(Numeric) Total received (Rx) packets on the interface.
tx_UcPkts	(Numeric) Total number of egress/transmitted unicast packets on the interface.
rx_UcPkts	(Numeric) Total number of ingress/received unicast packets on the interface.
tx_BcPkts	(Numeric) Total number of outgoing broadcast packets.

Table 17: Trace Interface Packet Statistics Data Object (*continued*)

Element Name	Description
rx_BcPkts	(Numeric) Total number of incoming broadcast packets.
tx_McPkts	(Numeric) Total number of outgoing multicast packets.
rx_McPkts	(Numeric) Total number of incoming multicast packets.
tx_Pps	(Numeric) Transmit bandwidth (packets per second).
rx_Pps	(Numeric) Receive bandwidth (packets per second).
tx_CrcErrors	(Numeric) Total CRC align errors while transmitting.
rx_CrcErrors	(Numeric) Total CRC align errors while receiving.
tx_DropPkt	(Numeric) Cumulative Tx packet drop in all the queues for the interface.
rx_DropPkt	(Numeric) Ingress Rx packet drops on the interface. This could be due to a buffer full condition.
tx_Bytes	(Numeric) Total transmit bytes.
rx_Bytes	(Numeric) Total receive bytes.
tx_Bw	(Numeric) Transmit bandwidth (bytes per second), the average rate at which packet bytes are transmitted.
rx_Bw	(Numeric) Receive bandwidth (bytes per second), the average rate at which packet bytes are received.
errorCntr	(Numeric) Error counter.

Trace Interface Statistics Data Learning Engine API Example

The following is a DLE API example for retrieving trace interface statistics. See [“GET /api/v1/traces/{traceId}/interface-stats” on page 33](#). The response data for these requests is in JSON format.

Request:

```
http://192.168.55.102:8282/api/v1/traces/
NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQ1NTE=/
interface-stats?end-time=1429641444500&device-serials=TA3714140387
&start-time=1429641004500&interface-name=et-0%2F0%2F48.0
&trace-id=NTYuNS4zLjEIOjMxMTgtNjYuNS4zLjEIOjQxMTgtdWRwQDE0MjMwODgzMzQ1NTE=
```


Response data:

```
{
  "id":
    "NTYyNS4zLjE1OjMxMTgtNjYuNS4zLjE1OjQxMTgtdWRwQDE0MjMwODgzMzQ1NTE=",
  "totalSize": 1,
  "traceStats": [
    {
      "deviceSerialNumber": "TA3714140387",
      "iflCount": 1,
      "iflTrafficRates": [
        {
          "interfaceName": "et-0/0/48.0",
          "sampleCount": 60,
          "samples": [
            {
              "timestamp": 1443561762500,
              "txPkts": 74,
              "rxPkts": 26,
              "txUcpPkts": 104,
              "rxUcpPkts": 63,
              "txBcPkts": 73,
              "rxBcPkts": 53,
              "txMcPkts": 53,
              "rxMcPkts": 65,
              "txPps": 21,
              "rxPps": 71,
              "txCrcErrors": 1,
              "rxCrcErrors": 1,
              "txDropPkts": 7,
              "rxDropPkts": 4,
              "txBytes": 62164,
              "rxBytes": 46491,
              "txBw": 44428,
              "rxBw": 39250,
              "errorCntr": 6
            },
            {
              "timestamp": 1443561763500,
              "txPkts": 51,
              "rxPkts": 56,
              "txUcpPkts": 103,
              "rxUcpPkts": 70,
              "txBcPkts": 85,
              "rxBcPkts": 70,
              "txMcPkts": 37,
              "rxMcPkts": 81,
              "txPps": 45,
              "rxPps": 85,
              "txCrcErrors": 7,
              "rxCrcErrors": 7,
              "txDropPkts": 1,
              "rxDropPkts": 5,
              "txBytes": 68454,
              "rxBytes": 52287,
            }
          ]
        }
      ]
    }
  ]
}
```

```
"txBw": 51048,
"rxBw": 36367,
"errorCnt": 0
},
{
  "timestamp": 1443561764500,
  "txPkts": 107,
  "rxPkts": 44,
  "txUcpPkts": 70,
  "rxUcpPkts": 59,
  "txBcPkts": 73,
  "rxBcPkts": 66,
  "txMcPkts": 53,
  "rxMcPkts": 81,
  "txPps": 38,
  "rxPps": 80,
  "txCrcErrors": 8,
  "rxCrcErrors": 9,
  "txDropPkts": 8,
  "rxDropPkts": 8,
  "txBytes": 61960,
  "rxBytes": 44943,
  "txBw": 53186,
  "rxBw": 36718,
  "errorCnt": 9
},
{
  "timestamp": 1443561765500,
  "txPkts": 72,
  "rxPkts": 24,
  "txUcpPkts": 83,
  "rxUcpPkts": 77,
  "txBcPkts": 96,
  "rxBcPkts": 44,
  "txMcPkts": 55,
  "rxMcPkts": 77,
  "txPps": 54,
  "rxPps": 63,
  "txCrcErrors": 9,
  "rxCrcErrors": 7,
  "txDropPkts": 7,
  "rxDropPkts": 3,
  "txBytes": 64032,
  "rxBytes": 33888,
  "txBw": 50079,
  "rxBw": 34569,
  "errorCnt": 5
},
{
  "timestamp": 1443561766500,
  "txPkts": 106,
  "rxPkts": 65,
  "txUcpPkts": 80,
  "rxUcpPkts": 52,
  "txBcPkts": 70,
  "rxBcPkts": 69,
```

```

        "txMcPkts": 62,
        "rxMcPkts": 82,
        "txPps": 23,
        "rxPps": 87,
        "txCrcErrors": 1,
        "rxCrcErrors": 0,
        "txDropPkts": 4,
        "rxDropPkts": 7,
        "txBytes": 56748,
        "rxBytes": 34659,
        "txBw": 61612,
        "rxBw": 35628,
        "errorCntr": 5
    },
    ...
]
}
]
}
]
}
}

```

- Related Documentation**
- [Data Learning Engine API Overview on page 3](#)
 - *Cloud Analytics Engine Feature Guide for the QFX Series*

CHAPTER 4

Managing DLE Streaming Data Subscription

- [Subscribing to and Unsubscribing from DLE Flow Path Data Streaming on page 41](#)

Subscribing to and Unsubscribing from DLE Flow Path Data Streaming

Use the DLE flow data subscription APIs as follows to:

- Subscribe to DLE's flow data streaming service and automatically receive flow data for all flows being traced by DLE.
- Unsubscribe from the DLE flow data streaming service.

DLE streams the flow path data to subscription service clients in the same format received from CA, as described in the CA **data/flow** and **data/tunnel** API response data object.

- [POST /api/v1/subscription/subscribeAll on page 41](#)
- [DELETE /api/v1/subscription/unsubscribe on page 42](#)
- [Flow Path Response Data Object on page 42](#)

POST /api/v1/subscription/subscribeAll

Use this DLE API request to subscribe to the DLE flow data streaming service, which enables clients to automatically receive bulk CA flow path analytics data for all flows being traced by DLE. DLE collects and streams the response data returned from CA, in JSON format (see [“Flow Path Response Data Object” on page 42](#)—the same flow path data objects returned from CA **data/flow** and **data/tunnel** API requests).

Request Format:

- **POST** method.

Request Parameters :

- receiverIP: (String) IP address of the receiver to which DLE will stream collected flow data. Required parameter.
- receiverPort: (Numeric, Integer) Receiver port number. Required parameter.

Response Data:

- Unique *subscription ID* that DLE associates with the given **receiverIP** and **receiverPort** to maintain the list of active subscriptions. Specify this value as the **subscriberId** parameter when subsequently requesting to unsubscribe (see [“DELETE /api/v1/subscription/unsubscribe” on page 42](#)).

Example Request (**POST** method):

http://localhost:8282/api/v1/subscription/subscribeAll?receiverIP=192.168.1.1&receiverPort=2000

Response:

192.168.1.1:2000@15645677

DELETE /api/v1/subscription/unsubscribe

Use this DLE API request to unsubscribe from the DLE flow data streaming service, and stop automatically receiving bulk CA flow path analytics data for flows being traced by DLE. This request requires the subscription ID returned by the original request to subscribe to the service (see [“POST /api/v1/subscription/subscribeAll” on page 41](#)).

Request Format:

- **DELETE** method.

Request Parameters:

- **subscriberId**: (String) Subscription ID of the subscription to stop. Required parameter.

Response Data:

- Status of request: (Boolean) **true** if client was successfully unsubscribed, and **false** otherwise.

Example Request (**DELETE** method):

http://localhost:8282/api/v1/subscription/unsubscribe?subscriberId=192.168.1.1:2000@15645677

Response:

true

Flow Path Response Data Object

[Table 18 on page 43](#) lists the elements in the response data object (JSON format) for requests to retrieve CA flow path analytics data. See [GET data/flow Resource](#) and [GET data/tunnel Resource](#), which describe the CA API resource requests that return this data.

Note that the Cloud Analytics Engine DLE component collects, stores, and processes this data as a client of CA **data/flow** and **data/tunnel** resources, and supports an analytics data subscription service that can stream this CA flow path data in bulk directly to subscribed DLE clients. See [“POST /api/v1/subscription/subscribeAll” on page 41](#) for

details on the DLE API you can use to subscribe to this service to receive this data automatically.

Table 18: Flow Path Analytics Data JSON Response Object

Element Name	Description
Path	Path data object that contains statistics from the devices in the flow path, containing the elements listed in Table 19 on page 43 .
ID	<p>A 5-tuple in JSON format identifying the flow or tunnel flow. The tuple values are as follows, in the order shown:</p> <ul style="list-style-type: none"> For non-overlay (data/flow resource): <i>{source IP address, destination IP address, protocol, source port, destination port}</i> For overlay (data/tunnel resource): <i>{VTEP source IP address, VTEP destination IP address, VXLAN ID, application source IP address, application destination IP address}</i>
Time	(Numeric) Time (epoch time in seconds) when the probe run for the flow was started.

[Table 19 on page 43](#) lists the elements in the Path data object of flow path data responses.

Table 19: Flow Path Analytics JSON Response Data: Path Object Elements

Element Name	Description
HOP	(Numeric) Number identifying the position of node in the path.
DevMEMUtil	(Numeric, decimal) Memory utilization of the node expressed as a percentage.
DevCPUUtil	(Numeric, decimal) CPU utilization of the node expressed as a percentage.
Bandwidth_Counter	(Numeric) Counter to track the flow packet count at the node.
Bandwidth_ID	(Numeric) Bandwidth measurement ID to track reset of bandwidth filters.
Latency	(Numeric, decimal) Latency for the node from CA server.
DevName	(String) Node name.
DevSerialID	(String) Node serial id.
TIMESTAMP	Timestamp object representing the time at which flow packet was received at the node. See Table 20 on page 44 .
IngressIFLAttribute	Ingress Logical Interface Attribute data object. See Table 21 on page 44 .
EgressIFLAttribute	Egress Logical Interface Attribute data object. See Table 22 on page 44 .

[Table 20 on page 44](#) lists the elements in the Timestamp data object of the Path data object in flow path data responses.

Table 20: Flow Path Analytics JSON Response Data: Timestamp Object Elements

Element Name	Description
TimeStamp	(Numeric) Seconds portion of the timestamp.
TimeStamp_usec	(Numeric) Microseconds portion of the timestamp.
Type	(String) Timestamp type. Accepted values are "PTP", "LCPU" or "NTP".

[Table 21 on page 44](#) lists the elements in the Ingress Logical Interface Attribute data object, part of the Path data object of flow path data responses.

Table 21: Flow Path Analytics JSON Response Data: Ingress Logical Interface Attribute Object Elements

Element Name	Description
IFLList	List of Logical Interface data objects. See Table 25 on page 45 .

[Table 22 on page 44](#) lists the elements in the Egress Logical Interface Attribute data object, part of the Path data object of flow path data responses.

Table 22: Flow Path Analytics JSON Response Data: Egress Logical Interface Attribute Object Elements

Element Name	Description
L3Ecmp	Layer 3 ECMP data object. See Table 23 on page 44 .

[Table 23 on page 44](#) lists the elements in the Layer 3 ECMP data object in the Egress Logical Interface Attribute data object (part of the Path data object) in flow path data responses.

Table 23: Flow Path Analytics JSON Response Data: Layer 3 ECMP Object Elements

Element Name	Description
IFLList	List of Logical Interface data objects. See Table 25 on page 45 .
NumBuckets	(Numeric) Number of buckets in the ECMP.
L2EcmpList	List of Layer 2 ECMP data objects. See Table 24 on page 45 .

[Table 24 on page 45](#) lists the elements in the Layer 2 ECMP List data object within the Layer 3 ECMP data object of flow path data responses.

Table 24: Flow Path Analytics JSON Response Data: Layer 2 ECMP List Object Elements

Element Name	Description
AE_IFL_NAME	(String) Aggregated logical interface name.
NumBuckets	(Numeric) Number of buckets in the ECMP.
IFLList	List of Logical Interface data objects. See Table 25 on page 45 .

[Table 25 on page 45](#) lists the elements in the Logical Interface data object (part of the Layer 2 ECMP and Layer 3 ECMP data objects) of flow path data responses.

Table 25: Flow Path Analytics JSON Response Data: Logical Interface Object Elements

Element Name	Description
IFLName	(String) Interface name.
IFLStats	Interface Packet Statistics data object. See Table 26 on page 45 .

[Table 26 on page 45](#) lists the elements in the Interface Packet Statistics data object (part of the Logical Interface data object) of flow path data responses.

Table 26: Flow Path Analytics JSON Response Data: Interface Packet Statistics Object Elements

Element Name	Description
TX_PKTS	(Numeric) Total transmitted (Tx) packets on the interface.
RX_PKTS	(Numeric) Total received (Rx) packets on the interface.
TX_UCPKTS	(Numeric) Total number of egress/transmitted unicast packets on the interface.
RX_UCPKTS	(Numeric) Total number of ingress/received unicast packets on the interface.
TX_BCPKTS	(Numeric) Total number of outgoing broadcast packets.
RX_BCPKTS	(Numeric) Total number of incoming broadcast packets.
TX_MCPKTS	(Numeric) Total outgoing multicast packets.
RX_MCPKTS	(Numeric) Total incoming multicast packets.
TX_PPS	(Numeric) Transmit bandwidth (packets per second).
RX_PPS	(Numeric) Receive bandwidth (packets per second).
TX_CRCERROR	(Numeric) Total CRC align errors while transmitting.
RX_CRCERROR	(Numeric) Total CRC align errors while receiving.

Table 26: Flow Path Analytics JSON Response Data: Interface Packet Statistics Object Elements (*continued*)

Element Name	Description
TX_DROPPKT	(Numeric) Cumulative Tx packet drop in all the queues for the interface.
RX_DROPPKT	(Numeric) Ingress Rx packet drops on the interface. This could be due to a buffer full condition.
TX_BYTES	(Numeric) Total transmit bytes.
RX_BYTES	(Numeric) Total receive bytes.
TX_BW	(Numeric) Transmit bandwidth (bytes per second), the average rate at which packet bytes are transmitted.
RX_BW	(Numeric) Receive bandwidth (bytes per second), the average rate at which packet bytes are received.

- Related Documentation**
- [Data Learning Engine API Overview on page 3](#)
 - [Compute Agent API Resources Overview](#)
 - [Cloud Analytics Engine Feature Guide for the QFX Series](#)

CHAPTER 5

Retrieving Network Traffic Analysis Statistics

- [Requesting NTA Top Application Statistics on page 47](#)
- [Requesting NTA Top Conversations Statistics on page 51](#)
- [Requesting NTA Application Flow Statistics on page 54](#)
- [Requesting NTA Conversation Flow Statistics on page 57](#)

Requesting NTA Top Application Statistics

Use the Cloud Analytics Engine DLE API as follows to retrieve Network Traffic Analysis (NTA) top-performing application statistics from network devices known to DLE.

- [GET /api/v1/nta/top-applications on page 47](#)
- [NTA Top Stats Response Transfer Object on page 49](#)
- [NTA Top Applications Data Learning Engine API Example on page 49](#)

GET /api/v1/nta/top-applications

Use this DLE API request to get NTA top application flow statistics collected by DLE. Response data is in JSON format.

Request Format:

- **GET** method

Request Parameters:

- [Table 27 on page 48](#) describes the request parameters.

Table 27: NTA Top Applications Request Parameters

Request Parameter	Description
device	(String) Device IP address.
if-index	(String) Interface SNMP index.
start	(Numeric) Start time in ms (epoch time). Query DLE for data starting at this timestamp.
end	(Numeric) Optional end time in ms (epoch time). Query DLE for data up until this timestamp.
count	(Numeric) Requested limit on number of top application statistics results returned (out of total available).

Response Data:

- [Table 28 on page 48](#) describes the response data.

Table 28: NTA Top Applications Response Data Object

Element Name	Description
total-applications	(Numeric) Total number of applications for which statistics were collected.
avg-byte-count	(Numeric) Average byte count for all applications for which statistics were collected.
limit	(Numeric) Number of applications for which statistics results are provided in the response. This might be limited based on the requested application count compared to total-applications.
top-applications	List of top application statistics data objects (see “NTA Top Stats Response Transfer Object” on page 49) by application identifier pair (port, protocol), based on the request parameters and total applications for which statistics were collected.

See [“NTA Top Applications Data Learning Engine API Example” on page 49](#) for a sample request and response.

NTA Top Stats Response Transfer Object

The DLE API returns the NTA top statistics transfer object in response to a request for NTA top application or conversation statistics. (See [“GET /api/v1/nta/top-applications” on page 47](#) and [“GET /api/v1/nta/top-conversations” on page 51](#).) The response data is a list of top statistics results, in JSON format, with each result identified by the following elements:

- For top applications statistics, an application identifier element pair:
 - port: (Numeric) Application port number.
 - protocol: (String) Application transport protocol - “TCP” | “UDP”.
- Or for top conversation statistics, a conversation identifier element pair:
 - src-ip: (String) Conversation source IP address.
 - dst-ip: (String) Conversation destination IP address.
- And the remaining elements in [Table 29 on page 49](#).

Table 29: NTA Top Stats Response Object Common Elements

Element Name	Description
ingress-bytes	(Numeric) Total number of ingress bytes generated by the application or on the conversation on the specified interface.
egress-bytes	(Numeric) Total number of egress bytes generated by the application or conversation on the specified interface.
total-bytes	(Numeric) Total number of bytes (ingress + egress) generated by the application or conversation on the specified interface.
bytes-percent	(Numeric, Decimal) Percent of bytes transferred by the specified application or on the specified conversation out of all traffic flow on the specified interface.
rank	(Numeric) Rank of this top application or conversation results among all the top applications or conversations in this response.

NTA Top Applications Data Learning Engine API Example

The following is a DLE API example for retrieving NTA top applications statistics from DLE. See [“GET /api/v1/nta/top-applications” on page 47](#) and [“NTA Top Stats Response Transfer Object” on page 49](#). This example requests 5 top applications flow statistics results, and the DLE response includes the top 5 applications out of a total of 20 that were collected in the requested time period. If statistics from fewer than 5 applications had been collected, the response would return results for as many applications as were available up to the requested count.

Request:

http://192.168.55.122:8282/api/v1/nta/top-applications?
device=192.168.55.93&if-index=521&start=1441057546726&count=5

Response:

```
{
  "total-applications": 20,
  "avg-byte-count": 12865412,
  "limit": 5,
  "top-applications": [
    {
      "port": 30583,
      "protocol": "udp",
      "rank": 1,
      "ingress-bytes": 42212544,
      "egress-bytes": 42165486,
      "total-bytes": 84378030,
      "bytes-percent": 32.7926
    },
    {
      "port": 21074,
      "protocol": "udp",
      "rank": 2,
      "ingress-bytes": 18364500,
      "egress-bytes": 18201480,
      "total-bytes": 36565980,
      "bytes-percent": 14.211
    },
    {
      "port": 113,
      "protocol": "udp",
      "rank": 3,
      "ingress-bytes": 3741736,
      "egress-bytes": 3972480,
      "total-bytes": 7714216,
      "bytes-percent": 2.998
    },
    {
      "port": 53,
      "protocol": "udp",
      "rank": 4,
      "ingress-bytes": 3882406,
      "egress-bytes": 3827858,
      "total-bytes": 7710264,
      "bytes-percent": 2.9965
    },
    {
      "port": 991,
      "protocol": "udp",
      "rank": 5,
      "ingress-bytes": 3915080,
      "egress-bytes": 3782410,
      "total-bytes": 7697490,
      "bytes-percent": 2.9915
    }
  ]
}
```

```
]
}
```

**Related
Documentation**

- [Data Learning Engine API Overview on page 3](#)
- *Cloud Analytics Engine Feature Guide for the QFX Series*

Requesting NTA Top Conversations Statistics

Use the Cloud Analytics Engine DLE API as follows to retrieve Network Traffic Analysis (NTA) top-performing conversation statistics from network devices known to DLE.

- [GET /api/v1/nta/top-conversations on page 51](#)
- [NTA Top Stats Response Transfer Object on page 53](#)
- [NTA Top Conversations Data Learning Engine API Example on page 53](#)

GET /api/v1/nta/top-conversations

Use this DLE API request to get NTA top conversation flow statistics collected by DLE. Response data is in JSON format.

Request Format:

- **GET** method

Request Parameters:

- [Table 30 on page 52](#) describes the request parameters.

Table 30: NTA Top Conversations Request Parameters

Request Parameter	Description
device	(String) Device IP address.
if-index	(String) Interface SNMP index.
start	(Numeric) Start time in ms (epoch time). Query DLE for data starting at this timestamp.
end	(Numeric) Optional end time in ms (epoch time). Query DLE for data up until this timestamp
count	(Numeric) Requested limit on number of top conversation statistics results returned (out of total available).

Response Data:

- [Table 31 on page 52](#) describes the response data.

Table 31: NTA Top Conversations Response Data Object

Element Name	Description
total-conversations	(Numeric) Total number of conversations for which statistics were collected.
avg-byte-count	(Numeric) Average byte count for all conversations for which statistics were collected.
limit	(Numeric) Number of conversations for which statistics results are provided in the response (might be limited based on the requested conversation count compared to total-conversations).
top-conversations	List of top conversation statistics data objects (see “NTA Top Stats Response Transfer Object” on page 49) by conversation identifier pair (source IP address, destination IP address), based on the request parameters and total conversations for which statistics were collected.

See [“NTA Top Conversations Data Learning Engine API Example” on page 53](#) for a sample request and response.

NTA Top Stats Response Transfer Object

The DLE API returns the NTA top statistics transfer object in response to a request for NTA top application or conversation statistics. (See [“GET /api/v1/nta/top-applications” on page 47](#) and [“GET /api/v1/nta/top-conversations” on page 51](#).) The response data is a list of top statistics results, in JSON format, with each result identified by the following elements:

- For top applications statistics, an application identifier element pair:
 - port: (Numeric) Application port number.
 - protocol: (String) Application transport protocol - “TCP” | “UDP”.
- Or for top conversation statistics, a conversation identifier element pair:
 - src-ip: (String) Conversation source IP address.
 - dst-ip: (String) Conversation destination IP address.
- And the remaining elements in [Table 29 on page 49](#).

Table 32: NTA Top Stats Response Object Common Elements

Element Name	Description
ingress-bytes	(Numeric) Total number of ingress bytes generated by the application or on the conversation on the specified interface.
egress-bytes	(Numeric) Total number of egress bytes generated by the application or conversation on the specified interface.
total-bytes	(Numeric) Total number of bytes (ingress + egress) generated by the application or conversation on the specified interface.
bytes-percent	(Numeric, Decimal) Percent of bytes transferred by the specified application or on the specified conversation out of all traffic flow on the specified interface.
rank	(Numeric) Rank of this top application or conversation results among all the top applications or conversations in this response.

NTA Top Conversations Data Learning Engine API Example

The following is a DLE API example for retrieving NTA top conversations statistics from DLE. See [“GET /api/v1/nta/top-conversations” on page 51](#) and [“NTA Top Stats Response Transfer Object” on page 49](#). In this example, although the request asks for 5 top conversation results, DLE collected fewer than that number in the requested time period, so the DLE response includes only the 2 results that are available.

Request:

```
http://192.168.55.122:8282/api/v1/nta/top-conversations?
device=192.168.55.93&if-index=521&start=1441057546726&count=5
```

Response:

```
{
  "total-conversations": 2,
  "avg-byte-count": 132476682,
  "limit": 5,
  "top-conversations": [
    {
      "src-ip": "198.51.100.2",
      "dst-ip": "10.0.0.2",
      "ingress-bytes": 0,
      "egress-bytes": 132571326,
      "total-bytes": 132571326,
      "bytes-percent": 50.0357,
      "rank": 1
    },
    {
      "src-ip": "10.0.0.2",
      "dst-ip": "198.51.100.2",
      "ingress-bytes": 132382038,
      "egress-bytes": 0,
      "total-bytes": 132382038,
      "bytes-percent": 49.9643,
      "rank": 2
    }
  ]
}
```

- Related Documentation**
- [Data Learning Engine API Overview on page 3](#)
 - [Cloud Analytics Engine Feature Guide for the QFX Series](#)

Requesting NTA Application Flow Statistics

Use the Cloud Analytics Engine DLE API as follows to retrieve Network Traffic Analysis (NTA) application statistics from network devices known to DLE.

- [GET /api/v1/nta/application-stats on page 54](#)
- [NTA Application Stats Samples Transfer Object on page 55](#)
- [NTA Application Stats Data Learning Engine API Example on page 56](#)

GET /api/v1/nta/application-stats

Use this DLE API request to get NTA application flow statistics samples collected by DLE. Response data is in JSON format.

Request Format:

- **GET** method

Request Parameters:

- [Table 33 on page 55](#) describes the request parameters.

Table 33: NTA Application Stats Request Parameters

Request Parameter	Description
device	(String) Device IP address.
if-index	(String) Interface SNMP index.
start	(Numeric) Start time in ms (epoch time). Query DLE for data starting at this timestamp.
end	(Numeric) Optional end time in ms (epoch time). Query DLE for data up until this timestamp.
aggregation-interval	(Numeric) Aggregation interval in seconds. Instructs DLE to aggregate (average) the sample data produced within the specified timestamps, and return the averaged samples per interval.
filter	<p>Applications statistics request data object consisting of a list of application identifiers (port, protocol) for which to filter the results and return statistics for only the specified applications, in the following JSON format:</p> <pre>{ "applications": [{ "protocol"=<"TCP" "UDP">,"port"="<port number>" }, ...] }</pre>

Response Data:

- application-stats: “[NTA Application Stats Samples Transfer Object](#)” on [page 55](#) in JSON format—A list of application identifiers and corresponding application flow statistics samples results for each application, as specified in the request parameters.

See “[NTA Application Stats Data Learning Engine API Example](#)” on [page 56](#) for a sample request and response.

NTA Application Stats Samples Transfer Object

The DLE API returns the Network Traffic Analysis (NTA) Application Statistics Samples transfer object in response to a DLE API request for NTA application statistics. See “[GET /api/v1/nta/application-stats](#)” on [page 54](#). The response data lists each application identifier pair specified in the request, followed by the corresponding list of samples. Results for each application are represented by the following elements in JSON format:

- protocol: (String) Application transport protocol - “TCP” | “UDP”.

- port: (Numeric) Application port number.
- samples: List of application statistics samples, described in [Table 34 on page 56](#), that were collected and aggregated by DLE according to the request parameters.

Table 34: NTA Application Samples Transfer Object

Element Name	Data Type	Description
timestamp	Numeric	Timestamp of sample in ms (epoch time).
total-bytes	Numeric	Total bytes of traffic transferred in the applications sample.

NTA Application Stats Data Learning Engine API Example

The following is a DLE API example for retrieving NTA application flow samples from DLE. See [“GET /api/v1/nta/application-stats” on page 54](#) and [“NTA Application Stats Samples Transfer Object” on page 55](#). The example request identifies two applications for which to filter the results, and the DLE response provides the available samples results for each application, aggregated over the specified interval from the specified starting timestamp.

Request:

```
http://192.168.55.122:8282/api/v1/nta/application-stats?
filter={{"applications":[{"protocol":"UDP","port":"20"},{"protocol":"UDP","port":"109"}]}}
&start=1442463323552&aggregation-interval=4&device=192.168.55.93&if-index=521
```

Response:

```
{
  "application-stats": [
    {
      "samples": [
        {
          "timestamp": 1442463323552,
          "total-bytes": 1429198
        },
        {
          "timestamp": 1442463623552,
          "total-bytes": 1070
        },
        {
          "timestamp": 1442463923552,
          "total-bytes": 466736
        },
        {
          "timestamp": 1442464223552,
          "total-bytes": 86026
        }
      ],
      "protocol": "udp",
      "port": 20
    }
  ]
}
```

```

    },
    {
      "samples": [
        {
          "timestamp": 1442463323552,
          "total-bytes": 1386124
        },
        {
          "timestamp": 1442463623552,
          "total-bytes": 1518
        },
        {
          "timestamp": 1442463923552,
          "total-bytes": 402036
        },
        {
          "timestamp": 1442464223552,
          "total-bytes": 85852
        }
      ],
      "protocol": "udp",
      "port": 109
    }
  ]
}

```

- Related Documentation**
- [Data Learning Engine API Overview on page 3](#)
 - [Cloud Analytics Engine Feature Guide for the QFX Series](#)

Requesting NTA Conversation Flow Statistics

Use the Cloud Analytics Engine DLE API as follows to retrieve Network Traffic Analysis conversation statistics from network devices known to DLE.

- [GET /api/v1/nta/conversation-stats on page 57](#)
- [NTA Conversation Stats Samples Transfer Object on page 58](#)
- [NTA Conversation Stats Data Learning Engine API Example on page 59](#)

GET /api/v1/nta/conversation-stats

Use this DLE API request to get NTA conversation flow statistics samples collected by DLE. Response data is in JSON format.

Request Format:

- **GET** method

Request Parameters:

- [Table 35 on page 58](#) describes the request parameters.

Table 35: NTA Conversation Stats Request Parameters

Request Parameter	Description
device	(String) Device IP address.
if-index	(String) Interface SNMP index.
start	(Numeric) Start time in ms (epoch time). Query DLE for data starting at this timestamp.
end	(Numeric) Optional end time in ms (epoch time). Query DLE for data up until this timestamp.
aggregation-interval	(Numeric) Aggregation interval in seconds. Instructs DLE to aggregate (average) the sample data produced within the specified timestamps, and return the averaged samples per interval.
filter	<p>Conversations statistics request transfer object consisting of a list of conversation identifiers (source IP address, destination IP address) for which to filter the results and return statistics for only the specified conversations, in the following JSON format:</p> <pre>{ "conversations": [{ "src-ip"="<IPv4 addr>","dst-ip"="<IPv4 addr>" }, ...] }</pre>

Response Parameters:

- conversation-stats: “[NTA Conversation Stats Samples Transfer Object](#)” on page 58 in JSON format—A list of conversation identifiers and corresponding conversation flow statistics samples results for each conversation, as specified in the request parameters.

See “[NTA Conversation Stats Data Learning Engine API Example](#)” on page 59 for a sample request and response.

NTA Conversation Stats Samples Transfer Object

The DLE API returns the NTA Conversation Statistics Samples transfer object in response to a DLE API request for NTA conversation statistics. See “[GET /api/v1/nta/conversation-stats](#)” on page 57. The response data lists each conversation identifier pair specified in the request, followed by the corresponding list of samples. The results for each conversation are represented by the following elements in JSON format:

- src-ip: (String) Conversation source IP address

- `dst-ip`: (String) Conversation destination IP address
- `samples`: List of conversation statistics samples, described in [Table 36 on page 59](#), that were collected and aggregated by DLE according to the request parameters.

Table 36: NTA Conversation Samples Transfer Object

Element Name	Description
<code>timestamp</code>	(Numeric) Timestamp of sample in ms (epoch time).
<code>total-bytes</code>	(Numeric) Total bytes of traffic transferred in the conversations sample.

NTA Conversation Stats Data Learning Engine API Example

The following is a DLE API example for retrieving NTA conversation flow samples from DLE. See [“GET /api/v1/nta/conversation-stats” on page 57](#) and [“NTA Conversation Stats Samples Transfer Object” on page 58](#). The example request identifies two conversations that make up the bidirectional traffic between two hosts for which to filter the results. The DLE response provides the available samples results for each conversation, aggregated over the specified interval from the specified starting timestamp.

Request:

```
http://192.168.55.122:8282/api/v1/nta/conversation-stats?
filter={"conversations":[{"src-ip":"198.51.100.2","dst-ip":"10.0.0.2"}, {"src-ip":"10.0.0.2","dst-ip":"198.51.100.2"}]}
&start=1441057546726&aggregation-interval=4&device=192.168.55.93&if-index=521
```

Response:

```
{
  "conversation-stats": [
    {
      "src-ip": "198.51.100.2",
      "dst-ip": "10.0.0.2",
      "samples": [
        {
          "timestamp": 1441057546726,
          "total-bytes": 208908
        },
        {
          "timestamp": 1441057550726,
          "total-bytes": 222132
        },
        {
          "timestamp": 1441057554726,
          "total-bytes": 206142
        },
        {
          "timestamp": 1441057558726,
          "total-bytes": 246626
        }
      ]
    }
  ]
}
```

```
        "timestamp": 1441057562726,
        "total-bytes": 222996
    },
    {
        "timestamp": 1441057566726,
        "total-bytes": 186214
    },
    {
        "timestamp": 1441057570726,
        "total-bytes": 210944
    },
    {
        "timestamp": 1441057574726,
        "total-bytes": 210866
    },
    {
        "timestamp": 1441057578726,
        "total-bytes": 215196
    }
  ]
},
{
  "src-ip": "10.0.0.2",
  "dst-ip": "198.51.100.2",
  "samples": [
    {
      "timestamp": 1441057546726,
      "total-bytes": 208232
    },
    {
      "timestamp": 1441057550726,
      "total-bytes": 220512
    },
    {
      "timestamp": 1441057554726,
      "total-bytes": 206642
    },
    {
      "timestamp": 1441057558726,
      "total-bytes": 249726
    },
    {
      "timestamp": 1441057562726,
      "total-bytes": 223846
    },
    {
      "timestamp": 1441057566726,
      "total-bytes": 187924
    },
    {
      "timestamp": 1441057570726,
      "total-bytes": 214044
    },
    {
      "timestamp": 1441057574726,
      "total-bytes": 216056
    }
  ],
}
```



```
    {  
      "timestamp": 1441057578726,  
      "total-bytes": 209136  
    }  
  ]  
}
```

- Related Documentation**
- [Data Learning Engine API Overview on page 3](#)
 - *Cloud Analytics Engine Feature Guide for the QFX Series*

