NetWitness® Platform

Version 12.4.1.0

API Core User Guide

Contact Information

NetWitness Community at https://community.netwitness.com contains a knowledge base that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

RSA, NetWitness, and other trademarks are trademarks of RSA Security LLC or its affiliates ("RSA"). For a list of RSA trademarks, go to https://www.rsa.com/en-us/company/rsa-trademarks. Other trademarks are trademarks of their respective owners.

License Agreement

The licensed software and the associated documentation, including this document, are proprietary and confidential to RSA and may be used and copied only in accordance with the terms of such license and with the inclusion of the copyright notice below. The licensed software and the documentation, and any copies thereof, may not be provided or otherwise made available to any other person.

No title to or ownership of the software or documentation or any intellectual property rights thereto is hereby transferred. Any unauthorized use or reproduction of this software and the documentation may be subject to civil and/or criminal liability.

The licensed software is subject to change without notice and should not be construed as a commitment by RSA.

Third-Party Licenses

This product may include software developed by parties other than RSA. The text of the license agreements applicable to third-party software in this product may be viewed on the product documentation page on NetWitness Community. By using this product, a user of this product agrees to be fully bound by terms of the license agreements.

Note on Encryption Technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when using, importing, or exporting this product.

Distribution

Use, copying, and distribution of any RSA software described in this publication requires an applicable software license.

RSA believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." RSA MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Miscellaneous

All products, software, and associated documentation are subject to the agreement between the parties or if none, to NetWitness' standard Terms and Conditions in effect as of the issuance date of this document which can be found at https://www.netwitness.com/standard-form-agreements/.

June, 2024

Contents

Overview

Netwitness Core Services API can be accessed through the hos and port of each core service.The core services comprises Decoder, Concentrator, LogDecoder, Broker, Archiver and Appliance.
The API can be accessed via REST through their respective ports.

Current Version

All request to the API will use lates version of API available with the service.

```
Netwitness-Version: 12.4.1
```

Schema

The core services API are sructured in tree form which includes nodes and node messages, it has predictable service and node oriented URLs.
The format of the URL consiss of the following components:

```
https://<hostname or IP>:<port>/[node1][/node2][...]?msg=<message name>[&param1=value1][&param2=value2][...]
```

The API accepts input request by using URL parameters and by POSTing application/json.

- The special content type, application/x-netwitness-string-params passes parameters as: plain text in the
  - format: param1=value1 param2="value \"2\"
  - `Note` : Quotes, as part of the value, must be preceded by the backslash \ character. Any character can be escaped in this manner, including the backslash itself `\\` .

The following are acceptable output formats:

- application/json text/plain
- text/xml text/html
- application/octet-stream for binary content
-
- The content type that is returned can be controlled through the HTTP Accept header. It is possible to set the parameter force-content-type to one of the previous values.

Any felds containing human readable timesamps or dates would be in the format

```
YYYY-MM-DD HH:MM:SS
```

HTTP Usage

API tries to adhere as closely as possible to sandard HTTP and REST conventions in its use of HTTP verbs and satus codes.

Code in the 2xx range indicates success.

Code in the 4xx range indicates an error that failed for the given information supplied (e.g a required parameter was omitted, URL path doesn't exist etc..).

Code in the 5xx range indicates an error with server (these are rare).

HTTP Verbs

Verb	Usage
`GET`	Used to retrieve resource. Along with message options on the URL `GET` can also be used to create, update and delete a resource.
`POST`	Used to create or upload a new resource.
`DELETE`	Used to delete an existing resource.

HTTP Status Codes

Stauts Code	Usage
`200 OK`	Request completed successfully.
`400 Bad Request`	The request was malformed. The response body will include an error providing further information.
`401 Unauthorized`	Similar to `403 Forbidden`, but specifically for use when authentication is required and has failed or has not yet been provided.
`403 Forbidden`	The request was valid, but the server is refusing the action. The user might not have the necessary permissions for a resource.
`404 Not Found`	The requested resource does not exist.
`500 Internal Server Error`	An unexpected error has occurred. The response body will include a message providing further information.

Service API Ports

The following are default ports that each core service lisen for REST requess.

Service	Port
Decoder	50104
LogDecoder	50102
Concentrator	50105
Broker	50103
Archiver	50108
Appliance	50106

Case Sensitive

All URLs, reques parameters and JSON felds are case sensitive.

Error Response

Error response along with HTTP satus code would be returned in the JSON format

Authentication and Authorization

All requess mus include the username and password in the `Authentication` parameters or as `Authorization` header (base64 encoded).

An example `curl` Authentication would be one of the following:

- Using `username and password`

```
curl -u username:password
```

- Using `Basic Authorization Header`

```
curl -H 'Authorization: Basic <Base64 encoded username:password>'
```

Where:

- `username` : The username of the service account. `password` : The
- password of that account.

The API supports both HTTP & HTTPS protocols and for secure communication prefer to use HTTPS and certifcates.

> This document refers `curl` commands to access REST API and to use ssl certificates for curl commands refer https://curl.se/docs/sslcerts.html
>
> (Optional) To configure custom certificates for the REST interface refer https://community.netwitness.com document `RESTful API User Guide` section `(Optional) Configure Custom SSLCertificate for the REST Interface`

API access over HTTPS is secured using TLS, all credentials will be encrypted in transit.

Authorization

In order to make requess through the API, users mus belong to groups associated with roles that have the access permissions, as well as any underlying permissions required to fulfll the reques. For details on service user roles and permissions refer https://community.netwitness.com section `Services Security View - Service User Roles and Permissions Note:` Through this document the API message details also include required security.roles permissions for accessing the API message.

Decoder service

Introduction

Decoder captures packets, sessionizes them, parses sessions, generates meta and index, and then writes the results to disk. The results are written to a session database, meta database, packet database and index (query engine). Typically the index on a decoder is jus time and no other felds are indexed, though they certainly can be.

The metadata generated from raw packet data capture is enriched with security context through session parsing.

API Usage

Decoder service provides API interface through its tree form as nodes, node messages and child nodes.

The following are few examples of the several API nodes and messages supported by Decoder:

- /decoder node messages:
  - `select` capture adapters `start` and `stop` packet capture `reconfig` decoder for optimal
  - settings `sslkeys` to push ssl crypto information for ssl decryption before session parsing
  - `/decoder/stats` node for decoder capture stats information /database node
  - messages:
  - `reconfig` database for optimal database settings `dbState` to get comprehensive
  - information about current databases
  - `/database/stats` node for database stats information /sdk node
  - messages:
  - `reconfig` sdk for optimal settings
  - `summary` for retrieving summary information from the databases `query` for
  - performing query against meta database
  - `/sdk/stats` node for queries active, pending and other sdk operations /sys node
  - messagges:
  - `statHist` to retrieve historical stats from stats database `/sys/stats` node for
  - service and system stats information

Sample Reques

The following is a sample reques to get lis of capture adapters on the decoder using select message:

Reques path is `/decoder` and message is `msg=select`

```
$ curl 'https://decoder-hos:50104/decoder?msg=select'  \
    -H 'Accept: application/json;charset=UTF-8' \
    -H 'Content-Type: application/x-www-form-urlencoded;charset=ISO-8859-1' \
    -u 'username:password'
```

Sample Response

HTTP Headers

```
HTTP/1.1 200 OK
Content-Length: 887
Connection: Keep-Alive
Pragma: no-cache
Expires: -1
Cache-Control: no-cache, no-sore, mus-revalidate
Content-Type: application/json
```

JSON Response

```
{
   "fags": 1073938433,
   "params": {
       "1": "packet_mmap_,em0 (bpf)",
       "2": "packet_mmap_,em1 (bpf)",
       "3": "packet_mmap_,lo (bpf)",
       "4": "packet_mmap_,ALL (bpf)",
       "5": "PFRINGZC,em0 (bpf)",
       "6": "PFRINGZC,em1 (bpf)",
       "7": "PFRINGZC,lo (bpf)",
       ...
    }
}
```

`Note`: The API Messages section for the path nodes provide more details on API messages available and any associated manual pages.

/connections node

API Messages

closeAll

Description: Closes all connections

Security.roles: connections.manage

Parameters:

Parameter	Type, Options	Description
type	sring, optional, {enum-any:The value mus be one or more of the following: native|res|amqp}	Indicates the type of connection to close, either native, res or amqp. Default is all types.
limit	uint32, optional	If non-zero, will sop closing connections after limit is reached. NOTE: It sarts closing the mos idle connections frs.
idleLongerThan	sring, optional	A time duration in HH:MM:SS or ##{d,h,m,s} format. Any connection that has been idle longer than the passed in duration is eligible to be closed. Connections that have seen activity within the duration window will not be closed. Examples: "01:05:10" (1 hour, 5 mins, 10 secs) or "65m" (65 minutes)
exceptMe	bool, optional	If true, will not close the connection that initiated the command.

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or
		the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

netSpeed

Description: Network Speed Tes **Security.roles:**

connections.manage,aggregate

Parameters:

Parameter	Type, Options	Description
size	size, optional, {byte-range:The value (in bytes, size modifers like MB are accepted) mus be between 1024 and 536870912, inclusive}	The size of each message sent, default is 64k
count	uint32, optional, {unsigned:The value mus be between 1 and 536870912, inclusive}	The number of messages to send, default is 16k

/database node

Manual for /database

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

dbState

Description: Returns comprehensive information about the current database sate or persiss the current sate to disk for fas reload **Security.roles:**

database.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: flelis|summary|save}	The operation to perform (flelis, summary or save)
type	sring, {enum-any:The value mus be one or more of the following: session|meta|packet}	The database(s) to operate on
options	sring, optional, {enum-one:The value mus be one of the following: bytes|pretty-print}	Output display options like 'bytes' or 'pretty-print'. Pretty print (the default) will convert all sizes from bytes to human readable quantities like MB or GB.

Manual for dbState

dump

Description: Dumps information out of the database in nwd formatted fles

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
session	uint64	The session id to dump
type	sring, optional, {enum-one:The value mus be one of the following: db|nwd}	The dump type
source	sring, optional, {enum-any:The value mus be one or more of the following: s|m|p}	The types of data to dump, default is all
verbose	bool, optional	If true (default is false), dumps more information
fle	sring, optional	Optional flename to use for NWD type, otherwise flename is assumed to be _sessionid_.nwd
limit	uint32, optional	If assigned, will only output the number of metas/packets passed in. Otherwise will print everything.

hashInfo

Description: Retrieves hash information for database fles that containing session/meta/packet objects for a set of sessions or date range. **Security.roles:**

database.manage **Parameters:**

Parameter	Type, Options	Description
sessions	sring, optional	A comma delimited lis of sessions and session ranges to retrieve fle hashes for.
beginDate	sring, optional	The beginning of a date range (YYYY-MM-DD HH:MM:SS) of database fle creation dates to retrieve fle hashes for.
endDate	sring, optional	The end of a date range (YYYY-MM-DD HH:MM:SS) of database fle creation dates to retrieve fle hashes for.
directories	sring, optional	A semi-colon delimited lis of additional directories to search for hash fles. This may be used to account for changes to the hash.dir confguration.

Manual for hashInfo

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

manifes

Description: If a manifes directory is defned, it will allow operations on the manifes fles (such as a time based query) for database fles in cold sorage.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
op	sring, optional, {enum-one:The value mus be one of the following: query|compress}	The operation to perform (defaults to query)
time1	date-time, optional	The beginning time (UTC) for matching ofine database fles
time2	date-time, optional	The ending time (UTC) for matching ofine database fles
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	Specify the time format that is returned (posix, simple), default is posix

Manual for manifest

optimize

Description: Runs a series of tess to determine the optimimal *.write.block.size based on confgured drives and using data from exising databases.

Security.roles: database.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, optional, {enum-any:The value mus be one or more of the following: session|meta|packet}	The database types to optimize, default is all
size	size, optional, {byte-range:The value (in bytes, size modifers like MB are accepted) mus be between 104857600 and 107374182400, inclusive}	The amount of data to write for each tes
compression	sring, optional, {enum-one:The value mus be one of the following: none|gzip|bzip2|lzma|zsd}	Determines if compression block sizes should be tesed, default is none
compressionLevel	uint32, optional, {unsigned:The value mus be between 0 and 22, inclusive}	Determines the compression level, 0-9, where 1 is fases, 9 is the bes compression (for zsd 22 is the bes compression) and zero is a predetermined balance between speed and compression

reconfg

Description: Calculates new drive sizes and free space for the session, meta and/or packet directories. No directories are removed and the assumption is each directory is mounted on a separate flesysem and will only be used for sorage of that database.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the respective *.dir and *.free.space.min settings, otherwise it will jus output the calculations
type	sring, optional, {enum-any:The value mus be one or more of the following: session|meta|packet}	The database types to reconfgure, default is all
percent	uint32, optional, {unsigned:The value mus be between 1 and 99, inclusive}	The drive percentage to use for the calculated size per directory, default is 98
op	sring, optional, {enum-one:The value mus be one of the following: normal|10g}	The operating mode, 'normal' or '10g'. Default is normal.

Manual for reconfig

resetMax

Description: Resets all max database sats or jus the ones lised.

Security.roles: database.manage **Parameters:**

Parameter	Type, Options	Description
sats	sring, optional, {enum-any:The value mus be one or more of the following: session.rate.max|meta.rate.max|packet.rate.max}	The sats to reset, default is all

sizeRoll

Description: Delete database fles based on the total size of all databases (passed with 'type' parameter) or space remaining on shared volume(s). This command should not be used on databases that don't share sorage.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
type	sring, {enum-any:The value mus be one or more of the following: session|meta|packet}	The databases to consider for removing the oldes data based on total size or space remaining
log	bool, optional	If true, will log all actions taken, otherwise all activity will happen silently. Default is true.
maxSize	size, optional	The maximum size of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxSize. This applies only to the Hot tier.
maxSizeWarm	size, optional	The maximum size of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxSize. This applies only to the Warm tier.
maxPercent	sring, optional	The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxPercent of total volumes. This applies only to the Hot tier.
maxPercentWarm	sring, optional	The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxPercent of total volumes. This applies only to the Warm tier.
minFree	size, optional	The minimum allowed free space on all volumes for all databases in 'type' parameter before oldes data is deleted. When minimum free space drops below target, the oldes data is deleted frs until free space once again exceeds target. This only applies to the Hot tier
minFreeWarm	size, optional	The minimum allowed free space on all volumes for all databases in 'type' parameter before oldes data is deleted. When minimum free space drops below target, the oldes data is deleted frs until free space once again exceeds target. This only applies to the Warm tier

Manual for sizeRoll

sagger

Description: Staggers database fles to optimize read/write performance across multiple volumes. Can be used after adding an empty mount point

Security.roles: database.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, {enum-one:The value mus be one of the following: session|meta|packet}	The databases fles to sagger
dryRun	bool, optional	If true (the default), will only return a description of the operations that would be performed. If false, then the fles will actually be moved to an optimal read/write pattern. Please turn of capture or aggregation before actually performing the sagger operation.

Manual for stagger

timeRoll

Description: Delete database fles that exceed a given age

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
type	sring, {enum-any:The value mus be one or more of the following: session|meta|packet}	The database fles to remove
timeCalc	sring, optional, {enum-one:The value mus be one of the following: current|las-write}	The time calculation to use (current date or las write date), default is current
minutes	uint32, optional, {unsigned:The value mus be between 0 and 10000, inclusive}	Remove database fles older than the given number of minutes
hours	uint32, optional, {unsigned:The value mus be between 0 and 10000, inclusive}	Remove database fles older than the given number of hours
days	uint32, optional, {unsigned:The value mus be between 0 and 10000, inclusive}	Remove database fles older than the given number of days
date	sring, optional	Remove database fles older than the given UTC date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

Manual for timeRoll

wipe

Description: Overwrites all packets and/or meta for a session with a pattern (for eliminating sensitive information). Meta keys sessionid, time and size always remain untouched.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
session	uint64	The session id whose packets will be wiped
payloadOnly	bool, optional	If true (default), will only overwrite the packet payload
pattern	sring, optional	The pattern to use, by default it uses all zeros
metaLis	sring, optional	Comma separated lis of meta to wipe, default (empty) is all meta
source	sring, optional, {enum-any:The value mus be one or more of the following: m|p}	The types of data to wipe, meta and/or packets, default is jus packets

/decoder node

Manual for /decoder

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

dpdk

Description: perform DPDK adminisration

Security.roles: decoder.manage **Parameters:**

Parameter	Type, Options	Description
migrate	sring, {not-empty:Value cannot be empty}	perform migration of exising capture confguration to DPDK:
commit	bool, optional	commit changes

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node

Security.roles: everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

reconfg

Description: Calculates optimal settings for decoder pools and bufers based on the insalled hardware.

Security.roles: decoder.manage **Parameters:**

Parameter	Type, Options	Description
update	bool, optional, {bool:The value mus be one of the following acceptable boolean values: 0,1,yes,no,true,false,on,of }	If true (default is false), will automatically update the respective decoder settings, otherwise it will jus output the calculations
op	sring, optional, {enum-one:The value mus be one of the following: normal|10g|ndr}	The operating mode, 'normal', '10g', or 'ndr'. Default is normal.
memory	size, optional	If non-zero, will use that setting as insalled physical RAM (e.g., memory="24 GiB"). This is useful for hybrids and/or all-in-one machines where the service will need to share memory.

Manual for reconfig

reset

Description: Reset data, index, manifess, sats, confguration, or logs for this service. Data automatically deletes index and sats, unless flesCreatedAfter is specifed. Service is automatically resarted. Example arguments:d ata=1 confg=1 log=1T his example will reset data, index, logs, and confguration index=1T his example will reset the index only manifes=1T his example will delete everything in the manifes directory flesCreatedAfter="2015-12-01 14:00:00"T his example will delete all session, meta and packet fles created on or after Dec 1s, 2015 2pm
(UTC) from the service. All other fles will remain. The index will not be touched, but upon resart will be truncated to match the las session

in the session database. **Security.roles:**

decoder.manage **Parameters:**

Parameter	Type, Options	Description
data	bool, optional	Reset data, automatically resets index (may not be applicable to all services)
index	bool, optional	Reset index, only valid if data is not reset (may not be applicable to all services)
manifes	bool, optional	Reset all manifess sored in the long term manifes directory (may not be applicable to all services)
confg	bool, optional	Reset confguration settings to default
sats	bool, optional	Reset the sats database
log	bool, optional	Reset the log database
flesCreatedAfter	date-time, optional	Delete all database fles (session, meta, packets) created after a certain date. The index will automatically get rolled back on resart. Not valid with option index.
cl	sring, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next resart
parseStats	bool, optional	Reset the parse sats database (PD/LD only)
force	bool, optional	Force reset without confrmation

Manual for reset

resetMax

Description: Resets all max sats to zero. **Security.roles:**

decoder.manage

select

Description: Selected a new capture device

Security.roles: decoder.manage **Parameters:**

Parameter	Type, Options	Description
adapter	sring, optional	The new adapter. If omitted will display a lis of available adapters. You may specify a comma-separated lis of adapter indexes to enable multi-interface capture.

Manual for select

sslKeys

Description: Push SSL crypto information to enable SSL decryption of a session's packets prior to parsing

Security.roles: decoder.manage **Parameters:**

Parameter	Type, Options	Description
clear	bool, optional	Clears all exising keys from sorage. Cannot be used with any other parameters.
maxKeys	uint32, optional	Sets the total number of keys that can be held in memory before aging out begins. Cannot be used with any other parameters.
counts	bool, optional	Returns key counts for sslKeys. Cannot be used with any other parameters.
random	sring, optional	Adds the random that identifes the session key exchange.
premaser	sring, optional	Adds the encryption key for the random.
CETS	sring, optional	Adds the TLS 1.3 client early trafc secret key for the random.
CHTS	sring, optional	Adds the TLS 1.3 client handshake trafc secret key for the random.
SHTS	sring, optional	Adds the TLS 1.3 server handshake trafc secret key for the random.
CTS0	sring, optional	Adds the TLS 1.3 client trafc secret 0 key for the random.
STS0	sring, optional	Adds the TLS 1.3 server trafc secret 0 key for the random.
pemFilename	sring, optional	Gives the name of a PEM fle that is the binary part of the reques. This should be a private key that can be used for decrypting SSL trafc.
lisPems	bool, optional	Returns the lis of all insalled PEM fles used for decryption.
deletePem	sring, optional	Given the PEM name, deletes it from the service. Can be repeated.

Manual for sslKeys

sart

Description: Starts aggregation

Security.roles: decoder.manage Manual for start

sop

Description: Stops aggregation

Security.roles: decoder.manage

Parameters:

Parameter	Type, Options	Description
fush	bool, optional	If true (default), process all sessions and packets currently in memory. Otherwise, sop immediately, discarding unprocessed sessions and packets.

Manual for stop

whoAgg

Description: Returns information on who is aggregating from this service

Security.roles: decoder.manage

Parameters:

Parameter	Type, Options	Description
clean	bool, optional	If true, will delete any aggregation trackers that no longer have a valid channel

Manual for whoAgg

/decoder/parsers node

API Messages

attrib

Description: Manage attributes for content fles.

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, optional, {enum-one:The value mus be one of the following: lis|update}	The operation to perform. If lis (default), return a lis of attributes for specifed type of contents. If update, set the attributes for specifed content fle.
type	sring, {enum-one:The value mus be one of the following: feeds|parsers|devices}	The content type (feeds|parsers|devices).
fle	sring, optional	The content fle name.
values	sring, optional	Lis of attributes to set for the specifed content fle in name value pairs.

content

Description: Get parsing content fle information.

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
fle	sring, optional	The content flename (type mus be specifed)
type	sring, optional, {enum-one:The value mus be one of the following: feeds|parsers}	The content type
uuid	sring, optional	The content uuid

count

Description: Returns the number of child nodes **Security.roles:** everyone

delete

Description: Delete parsing content fles.

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
fle	sring, optional	The content flename (type mus be specifed)
type	sring, optional, {enum-one:The value mus be one of the following: feeds|parsers}	The content type (feeds|parsers)
uuid	sring, optional	The content uuid

devices

Description: Returns the log parsers

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
uuid	sring, optional	The content uuid

disable

Description: Disable the specifed content.

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, {enum-one:The value mus be one of the following: feeds|parsers|devices}	The content type (feeds|parsers|devices)
uuid	sring	The content uuid

enable

Description: Enable the specifed content.

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, {enum-one:The value mus be one of the following: feeds|parsers|devices}	The content type (feeds|parsers|devices)
uuid	sring	The content uuid

export

Description: Returns the content as a package.

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, {enum-one:The value mus be one of the following: feeds|parsers|devices}	The content type (feeds|parsers|devices)
uuid	sring, optional	The content uuid
fle	sring, optional	The content fle name

feed

Description: Manages the feed parsers

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: notify|reload|remove|delete}	The feed operation to perform
fle	sring, optional	The feed flename

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

reload

Description: Reloads all meta parsers **Security.roles:**

parsers.manage

reset

Description: Reset content of the specifed type (feeds|parsers|devices|all or 1). Content type can be specifed in csv format e.g content="feeds,parsers,devices". Content type 'devices' is only valid for LogDecoder. Example arguments:c ontent="feeds,parsers"T his example will reset content for types feeds and parsers. content=1 or content=allT his example will reset content for all types i.e feeds,parsers,devices. **Security.roles:** parsers.manage

Parameters:

Parameter	Type, Options	Description
content	sring	The content type (feeds|parsers|devices|all or 1)
force	bool, optional	Force content reset without confrmation

schema

Description: Returns the parsers meta schema

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
onlyShowEnabled	bool, optional	If true, only return information on parsers that are currently enabled
showRegiseredMeta	bool, optional	If true (the default), show the regisered meta for each parser
showOptions	bool, optional	If true, shows the options supported by each parser. Passing true changes the XML sructure returned so each parser (potentially) has
prettyPrint	bool, optional	If true, returns the XML formatted to be easier to read. Default is false.

snrtStat

Description: Get Snort rule satisics **Security.roles:**

parsers.manage

satHis

Description: Retrieve hisorical sats from the sats db. Don't send time1/time2 to get bounding times about sats db. Supported wildcards are ? to match any single char, * to match zero or more characters, not including slash /, ** to match zero or more characters including slash /. **Security.roles:** parsers.manage

Parameters:

Parameter	Type, Options	Description
time1	date-time, optional	The sarting time for retrieving sats
time2	date-time, optional	The ending time for retrieving sats
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	Specify the time format for each sat snapshot, default is posix (seconds since 1970)
include	sring, optional	Comma separated lis of sats to include (wildcards allowed)
exclude	sring, optional	Comma separated lis of sats to exclude (wildcards allowed, has precedence over include)
onChange	sring, optional	Comma separated lis of sats that reduce the result set to only when at leas one of the sats change from their previous value
reduce	bool, optional	If true, reduces data transmission by replacing sat pathnames with a shorthand and provides a lookup table as frs result
showAll	bool, optional	If true, changes each result to show all sats previously seen, even if the value hasn't changed. It has a moderate performance impact and is bes used with include, exclude or onChange parameters to reduce the returned result set size.

upload

Description: Upload feed/parser/geoip2 fles to this service

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: sart|attributes|cancel|fnished}	The operation to perform (sart, cancel, fnished)
size	uint64, optional	The size of the incoming fle
flename	sring, optional	The name of the incoming fle
fnalCount	uint32, optional	Las chunk count. To be used with fnished operation
reload	uint32, optional	Reload all parsers when upload is complete for the appropriate fle types (default). A setting of 0 when sending a fnished op will disable the reload.

/index node

API Messages

cacheClr

Description: Clear index internal caches **Security.roles:**

index.manage

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

inspect

Description: Provides overall information about the index or can inspect specifc keys and values

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
key	sring, optional	The key to inspect
value	sring, optional	The value to inspect
format	sring, optional, {enum-one:The value mus be one of the following: Text|IPv4|Int8|UInt8|Int16|UInt16|Int32|UInt32|Int64|UInt64|UInt128|Float32|Float64|TimeT|DayOfWeek|HourOfDay|Binary|IPv6|MAC}	Only include keys that have this format
valueCount	uint64, optional	The number of values to return from each key
allSlices	bool, optional	Return information on all slices, insead of jus mos recently created slice
summarizeAllValues	bool, optional	Calculate total summary and page usage from all values in each slice

language

Description: Returns language information

Security.roles: index.manage **Parameters:**

Parameter	Type, Options	Description
values	bool, optional	If set to 1, will describe the value overrides defned in the language

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

profle

Description: Returns information about the index used by the Index Profle tool

Security.roles: index.manage **Parameters:**

Parameter	Type, Options	Description
key	sring, optional	The key to profle
value	sring, optional	The value to profle
max	uint32, optional	Max number of pages to return
fag	uint32, optional	Bitwise option when key is specifed. 1: Break report to values. 2: Lis page sizes. 4: Categorize page counts in diferent range. 8: Sessions count Per page with each algorithm. 16: Verify session page

reconfg

Description: Calculates new sizes for the summary and page directories. NOTE: Any size changes require a reset of the index before they take efect.

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the respective confg node settings, otherwise it will jus output the calculations
percent	uint32, optional, {unsigned:The	The drive percentage to use for the index database, default is 90
	value mus be between 1 and 99, inclusive}	

save

Description: Saves the index to disk **Security.roles:**

index.manage

schema

Description: Introspects all open Meta DB fles for meta keys and merges those keys with the defned language to return a fairly comprehensive

lis of all meta keys in exisence. **Security.roles:** index.manage

sizeRoll

Description: Delete index and other DB fles (see 'type' param) based on the total size or space remaining on shared volumes. This command is not intended to be used on databases that do not share sorage.

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
log	bool, optional	If true, will log all actions taken, otherwise all activity will happen silently. Default is true.
type	sring, optional, {enum-any:The value mus be one or more of the following: session|meta|packet}	The databases to consider for removing the oldes data based on total size or space remaining
maxSize	size, optional	The maximum size of the index hot sorage. When exceeded, oldes data is deleted frs until total size is less than maxSize.
maxSizeWarm	size, optional	The maximum size of the index warm sorage. When exceeded, oldes data is delete frs until total size is less that maxSizeWarm
maxPercent	sring, optional	The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxPercent of total volumes. This parameter applies to hot sorage only.
maxPercentWarm	sring, optional	The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxPercent of total volumes. This parameter applies to warm sorage only.
minFree	size, optional	The minimum allowed free space on the volumes before oldes data is deleted. This parameter applies to hot sorage only.
minFreeWarm	size, optional	The minimum allowed free space on the volumes before oldes data is deleted. This parameter applies to warm sorage only.

values

Description: Returns the top n values for the given key

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
key	sring	The index key for values
top	uint32, {unsigned:The value mus be between 0 and 10000, inclusive}	The number of top entries to return

/logs node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

download

Description: Downloads log entries

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The sarting log number to download
id2	uint64, optional	The ending log number to download
time1	date-time, optional	The sarting log date to download, can be posix time (seconds since 1970) or DD-MM-YY HH:MM:SS
time2	date-time, optional	The ending log date to download, can be posix time (seconds since 1970) or DD-MM-YY HH:MM:SS
op	sring, {enum-one:The value mus be one of the following: sart|next|cancel}	The operation to perform (sart, next, cancel)
logTypes	sring, optional	The type(s) of log messages to retrieve (debug,info,audit,warning,failure)
match	sring, optional	Case insensitive sring to match in each log message
regex	sring, optional	Regular expression to match in each log message
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	The time format used in each log message, default is posix time (seconds since 1970)
batchSize	uint32, optional, {unsigned:The value mus be between 2000 and 10000, inclusive}	The number of logs to retrieve per batch, default is 2000

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

pull

Description: Downloads N log entries

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The frs log id number to retrieve, this is mutually exclusive with id2
id2	uint64, optional	The las log id number that will be sent, defaults to mos recent log message when id1 or id2 is not sent
count	uint32, optional, {unsigned:The value mus be between 1 and 10000, inclusive}	The number of logs to pull
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	The time format used in each log message, default is posix time (seconds since 1970)

timeRoll

Description: Delete log entries that exceed a given age

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
timeCalc	sring, optional, {enum-any:The value mus be one or more of the following: current|las-write}	The time calculation to use (current date or las write date), default is current
minutes	uint32, optional, {unsigned:The value mus be between 0 and 60, inclusive}	Remove log entries older than the given number of minutes
hours	uint32, optional, {unsigned:The value mus be between 0 and 24, inclusive}	Remove log entries older than the given number of hours
days	uint32, optional, {unsigned:The value mus be between 0 and 365, inclusive}	Remove log entries older than the given number of days
date	sring, optional	Remove log entries older than the given date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

/res node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated
		values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

/sdk node

Manual for /sdk

API Messages

aliases

Description: Retrieves aliases for the specifed key

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
key	sring	The language key to retrieve aliases for
value	sring, optional	An optional value to return a specifc alias for

cancel

Description: Cancel a running query, regardless of the user

Security.roles: sdk.manage,sdk.meta **Parameters:**

Parameter	Type, Options	Description
handle	uint32	The channel handle running the query
uuid	sring, optional	The device UUID if taking a device ofine insead of canceling the entire query (Broker only). Passing uuid will propagate the reques upsream to fnd the frs matching device to cancel.

content

Description: Returns the packet content for a session

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
session	uint64, optional	The session id to retrieve packet content for
packet	uint64, optional	The packet id of the frs packet of the session, in case the session has rolled out
pinId	sring, optional	The Pin ID of the session to retrieve from long term sorage. If provided, the packet and session parameters mus not be

		provided.
maxSize	size, optional	The max number of bytes to return, zero means no limit
failOnCacheMiss	bool, optional	If true, the content call fails if the NWD is not in the cache. This is used by optimized caching code only.
renderType	sring, optional	The render type (see NwContentTypes in NwSDK.h), pass zero to pick bes option. Not specifying this parameter always results in an NWD and renderFlags and renderOptions are not considered. The following values are also supported: 'pcap', 'fle-lis-json', 'session-meta-fle-lis-json', 'nwd', 'log', 'fles'
renderFlags	uint32, optional	Bitwise mask to control options, (see NwContentFlags in NwSDK.h)
renderOptions	sring, optional	An encoded sring params containing additional options for controlling the operation.

Manual for content

count

Description: Returns the number of child nodes **Security.roles:** everyone

delCache

Description: Removes all cached NWD fles from either the normal NWD cache or the pin cache

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
type	sring, optional, {enum-one:The value mus be one of the following: nwd|pin|all}	Specifes which cache to delete all NWD fles from. The default is 'nwd'. Passing 'all' will delete from all caches.

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

hierarch

Description: Report the set of devices attached to this service

Security.roles: sdk.meta Manual for

hierarch **info**

Description: Returns detailed information about the node **Security.roles:** everyone

keyrefs

Description: Retrieves the set of entity keys

Security.roles: sdk.meta Manual for

keyrefs

language

Description: Retrieves the language defnition

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The sarting language id
id2	uint64, optional	The ending language id
time1	date-time, optional	The sarting time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS)
time2	date-time, optional	The ending time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS)
options	sring, optional	Extra options, currently unused
fags	sring, optional	Optional fags to confgure how the results are returned. Can be a number (bitwise mask) or comma separated values like default, no-count, quick-count or full-count.
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query

Manual for language **ls**

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

	~~regex|sp|sm|ci|pre|pos|ds|precache|si}~~	~~regex,sp,sm,ci,pre,pos,ds,precache,si~~
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

msearch

Description: Search for pattern matches in many sessions or packets

Security.roles: sdk.content & sdk.meta

Parameters:

Parameter	Type, Options	Description
sessions	sring, optional	The session ID ranges to search
packets	sring, optional	The packet ID ranges to search
search	sring	The search sring to use
where	sring, optional	The where clause used to identify sessions to consider for the search
limit	uint64, optional	The maximum number of sessions to traverse for this search
size	uint64, optional	The maximum number of results to return for this search
fags	sring, {enum-any:The value mus be one or more of the following:	Flags to use for search. This is a comma separated lis of one or more of the fag values:

Manual for msearch

packets

Description: Stream packets back based on the input parameters provided

Security.roles: sdk.content & sdk.packets **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: setup|sart|cancel|processed}	The operation to perform (sart, cancel, processed)
sessions	sring, optional	A comma separated lis of session ids or session id ranges (#-#) whose packets will be sreamed back.
packets	sring, optional	A comma separated lis of packet ids or session/packet ids (#&#) that will be sreamed back.
where	sring, optional	Where clause which will be evaluated to determine which sessions to sream back
time1	date-time, optional	A sarting time range (UTC) where matching packets will be sreamed back ("2010-Apr-20 09:00:00")
time2	date-time, optional	An ending time range (UTC) where matching packets will be sreamed back ("2010-Apr-20 10:00:00")
fags	uint32, optional	Additional fags as defned by the NwPackets SDK function
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query

Manual for packets

pin

Description: Pin a session in the long term cache so it can be retrieved even after the meta or packets have rolled out. To retrieve, you mus make a content call with the Pin ID that is returned from this command. The op={unpin,validate} commands all take one or more comma separated Pin IDs in the pinId parameter and returns the satus for the Pin IDs that were recognized by a service.

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
session	uint64, optional	The session ID to pin. This cannot be used with the 'sessions' parameter.
sessions	sring, optional	A comma separated lis of session ids or session id ranges (#-#) to pin. This cannot be used with the 'session' parameter.
pinId	sring, optional	One or more Pin IDs, separated by commas
op	sring, optional, {enum-one:The value mus be one of the following: ls|unpin|validate}	Various operations that can be performed. Will be submitted to upsream devices if not confgured for long term cache.

Manual for pin

precache

Description: Efciently caches NWD fles for future retrieval via the content message

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
sessions	sring, optional	A comma separated lis of session ids or session id ranges (#-#) that will be saved in the NWD cache directory
packets	sring, optional	A comma separated lis of session and packet ids (#p#) that will be saved in the NWD cache directory
waitOnComplete	bool, optional	If true, will wait for precache to complete before returning a response.

query

Description: Performs a query agains the meta database

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The sarting meta id (to run the query from mos recent to oldes meta, make id1 larger than id2)
id2	uint64, optional	The ending meta id
size	uint32, optional	The max number of entries to return, or jus sream back all results if zero
query	sring, optional	The query sring to use
fags	sring, optional	The fags to use for query. Can be a number (bitwise mask) or comma separated values like query-log.
threshold	uint64, optional	Query optimization to sop processing results after the threshold is reached (useful with select aggregate functions). Zero means no threshold (the default).
partition	uint32, optional	Used to partition the result set across multiple clients. Mus be greater than zero and less than or equal to totalPartitions
totalPartitions	uint32, optional	The total number of clients that will be submitting the same query for partitioning.
search	sring, optional	A text search clause to apply to the where clause of this query
sreamLimit	uint32, optional	Limit the number of sessions returned by a sreaming query
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query

Manual for query

reconfg

Description: Calculates default values for some of the confg nodes.

search	sring	The search sring to use

Security.roles: sdk.manage **Parameters:**

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the confg nodes with the defaults
memory	size, optional	If non-zero, will use that setting as insalled physical RAM (e.g., memory="24 GiB"). This is useful for hybrids and/or all-in-one machines where the service will need to share memory.

search

Description: Searches for matches in session/packet content

Security.roles: sdk.content

Parameters:
Manual for search

session

Description: Retrieves the meta id range for the session range

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
id1	uint64	The sarting session id (inclusive). Pass zero to scope to lowes valid id.
id2	uint64	The ending session id (inclusive). Pass zero to scope to highes valid id.

Manual for session

summary

Description: Retrieves summary information from the databases

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
session	uint64	The session id to search
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return

Parameter	Type, Options	Description
fags	sring, optional	Optional SDK fags. Can be a number (bitwise mask) or comma separated values like default or ignore-cache.

Manual for summary

timeline

Description: Returns the count of sessions/size/packets in discrete time intervals

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
time1	date-time, optional	The sarting time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS), if greater than time2, then the whole time range is returned
time2	date-time, optional	The ending time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS)
timezone	int32, optional	The timezone of the local computer to receive the data, hour

		ofset from GMT [-13 to 13]
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
fags	sring, optional	The fags to use for the timeline. Can be a number (bitwise mask) or comma separated values like sessions, size, packets, order-ascending, order-descending, ignore-cache or clear-cache.
where	sring, optional	Optional where clause for fltering the data
bookendHours	uint32, optional	The number of hours outside the given time1, time2 which will become part of the cache. Default is 24 hours.
feldName	sring, optional	The time feld to retrieve values for, defaults to 'time'
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
search	sring, optional	A text search clause to apply to the where clause of this query

Manual for timeline **validate**

Description: Validate a query or values call without execution

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: query|values}	The operation to validate
validateLanguage	bool, optional	If true (default is false), language tokens and syntax will be validated, otherwise jus syntax will be validated
feldName	sring, optional	Required if validating a 'values' operation. The feld to retrieve values for.
where	sring, optional	Required if validating a 'values' operation. The flter criteria to be validated.
query	sring, optional	Required if validating a 'query' operation. The query sring to be validated.
json	bool, optional	Structure response in JSON syntax

Manual for validate

values

Description: Performs a value count query and returns the matching values for a report

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The sarting meta id
id2	uint64, optional	The ending meta id
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
fags	sring, optional	The fags to use for values. Can be a number (bitwise mask) or comma separated values like sessions, size, packets, aggregate, sort-total, sort-value, order-ascending, order-descending, ignore-cache, clear-cache, or database-scan.
threshold	uint64, optional	Query optimization to sop processing large session counts
feldName	sring	The feld to retrieve values for
where	sring, optional	The flter criteria for the values
aggregateFunction	sring, optional, {enum-one:The value mus be one of the following: count|sum}	The aggregate operation to be applied where the aggregate fag is set
aggregateFieldName	sring, optional	The meta feld to aggregate in the aggregateFunction
min	sring, optional	The lower limit of values to return. Only values greater than this value will be returned.
max	sring, optional	The upper limit of values to return. Only values less than this value will be returned.
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
search	sring, optional	A text search clause to apply to the where clause of this query

Manual for values

xforms

Description: Retrieves transforms for the specifed key

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
key	sring, optional	An optional language key to retrieve transform for, all transforms returned if not provided

/sys node

Manual for /sys

API Messages

caCert

Description: Display or delete trused CA certs

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: lis|delete|add}	CA action to perform (lis,delete,add)
id	sring, optional	ID of CA certifcate to remove when op=delete

Manual for caCert

count

Description: Returns the number of child nodes **Security.roles:** everyone

fleEdit

Description: View and edit sysem confguration fles

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: dir|get|getBackup|put|create|delete}	The operation to perform (dir, get, getBackup, put, create, delete)
type	sring, optional, {enum-one:The value mus be one of the following: parser|flter}	The type of fle, if not provided will check fle extension
flename	sring, optional, {length:The value length mus be between 1 and 64, inclusive}	File to create, edit, get or delete

Manual for fileEdit

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

peerCert

Description: Display or modify trused peer certifcates

Security.roles: sys.manage
Parameters:

Parameter	Type, Options	Description
op	sring, optional, {enum-one:The value mus be one of the following: lis|delete|add}	Operation to perform on the lis of trused peers (lis,delete,add)
id	sring, optional	ID of peer certifcate to remove when op=delete

Manual for peerCert

save

Description: Forces a save of sysem settings

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
force	uint32, optional	If 1, forces the confguration fle to saved, regardless if changes are detected, default is 0

Manual for save

servCert

Description: Display the current server certifcate in PEM format.

Security.roles: sys.manage Manual for servCert

shutdown

Description: Stop the service

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
reason	sring, optional, {length:The value length mus be between 0 and 512, inclusive}	The reason for shutting down the service
cl	sring, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next resart

Manual for shutdown

satHis

Description: Retrieve hisorical sats from the sats db. Don't send time1/time2 to get bounding times about sats db. Supported wildcards are ? to match any single char, * to match zero or more characters, not including slash /, ** to match zero or more characters including slash /. **Security.roles:** sys.manage

Parameters:

Parameter	Type, Options	Description
time1	date-time, optional	The sarting time for retrieving sats
time2	date-time, optional	The ending time for retrieving sats
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	Specify the time format for each sat snapshot, default is posix (seconds since 1970)
include	sring, optional	Comma separated lis of sats to include (wildcards allowed)
exclude	sring, optional	Comma separated lis of sats to exclude (wildcards allowed, has precedence over include)
onChange	sring, optional	Comma separated lis of sats that reduce the result set to only when at leas one of the sats change from their previous value
reduce	bool, optional	If true, reduces data transmission by replacing sat pathnames with a shorthand and provides a lookup table as frs result
showAll	bool, optional	If true, changes each result to show all sats previously seen, even if the value hasn't changed. It has a moderate performance impact and is bes used with include, exclude or onChange parameters to reduce the returned result set size.

Manual for statHist

telemtry

Description: Returns telemetry information about the service

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
level	uint32, optional, {signed:The value mus be between 1 and 3, inclusive}	The level of information to be sent by the telemetry reques. A value of 1 (default) will send all sat nodes, a value of 2 will send sat and confg nodes, and a value of 3 will send sat nodes, confg nodes and usage metrics for rules, parsers and devices (where applicable).
include	sring, optional	Comma separated lis of nodes to include (wildcard usage is ok).
exclude	sring, optional	Comma separated lis of nodes to exclude (wildcard usage is ok). If "include" is also specifed, exclusions are applied to the set of nodes specifed by "include".
url	sring, optional	Endpoint URL to send the JSON response.
op	sring, optional, {enum-one:The value mus be one of the following: get|confgure}	The telemetry operation to execute. Default is 'get'.

/users node

API Messages

addOrMod

Description: Add a new user or update an exising user in the sysem

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#$%&'+-=?^_`{|}~.[]\}	The username to add or update, mus be alphanumeric or @!#$%&'+-=?^_`{|}~.[]\
password	sring, optional	The user's password
pwdIsHashed	bool, optional	Set to true if the password has already been hashed
groups	sring, optional	The groups the user belongs to
authType	sring, optional, {enum-one:The value mus be one of the following: netwitness|pam}	The authentication sysem to use, "netwitness" is the default
queryTimeout	uint32, optional, {unsigned:The value mus be between 0 and 43200, inclusive}	The maximum number of minutes that a query is allowed to execute for this user. Zero means unlimited.
displayName	sring, optional	Display name for this user account
email	sring, optional	Email address

		for this account
description	sring, optional	Description of this user account
queryPrefx	sring, optional	Query flter applied to every query performed by this user account
sessionThreshold	uint32, optional, {unsigned:The value mus be between 0 and 4294967295, inclusive}	Query optimization which will extrapolate the remaining session counts when they exceed this value

auths

Description: Get supported authentication types **Security.roles:** everyone

count

Description: Returns the number of child nodes **Security.roles:** everyone

delete

Description: Delete a user from the sysem

Security.roles: users.manage **Parameters:**

Parameter	Type, Options	Description
name	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#$%&'+-=?^_`{|}~.[]\}	The username to delete

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone
Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

unlock

Description: Unlock a locked out account

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	sring	The username or account to unlock

whoAmI

Description: Returns information about the authenticated user

Security.roles: everyone

LogDecoder service

Introduction

LogDecoder captures logs, parses the logs into meta, indexes them and writes the results to disk. It also has a session, meta and packet database (which is really the log message) and index. Similar to a Decoder the LogDecoder also has minimal index with fewer felds along with time.

The metadata generated from raw log capture is enriched with security context through session parsing.

API Usage

LogDecoder service provides API interface through its tree form as nodes, node messages and child nodes.

The following are few examples of the several API nodes and messages supported by LogDecoder:

- /decoder node messages:
  - `start` and `stop` logs capture `reconfig` logdecoder for
  - optimal settings
  - `/decoder/stats` node for logdecoder capture stats information /decoder/parsers node
- messages:
  - `content` to get parsing file contents `devices` list of log
  - parser devices `ipdevice` map IP to Device type in log parsing
  - `schema` log parsers meta schema /database node messages:
- `reconfig` database for optimal database settings `dbState` to get comprehensive information about current databases
  - `/database/stats` node for database stats information /sdk node
  - messages:
- `reconfig` sdk for optimal settings
  - `summary` for retrieving summary information from the databases `query` for
  - performing query against meta database
  - `/sdk/stats` node for queries active, pending and other sdk operations /sys node
  - messagges:
- `statHist` to retrieve historical stats from stats database. `/sys/stats` node for service and system stats information
  -

Sample Reques

The following is sample reques to describe ip device mappings on the logdecoder.

Reques path is `/decoder/parsers`, message is `ipdevice` and parameters are `op=describe` .

```
$ curl 'https://logdecoder-hos:50102/decoder/parsers?msg=ipdevice&op=describe'  \
    -H 'Accept: application/json;charset=UTF-8' \
    -H 'Content-Type: application/x-www-form-urlencoded;charset=ISO-8859-1' \
    -u 'username:password'
```

Sample Response

HTTP Headers

```
HTTP/1.1 200 OK
Content-Length: 278
Connection: Keep-Alive
Pragma: no-cache
Expires: -1
Cache-Control: no-cache, no-sore, mus-revalidate
Content-Type: application/json
```

JSON Response

The sample response here is JSON encoded ip-device mapping schema in xml format. It is possible for response values to contain xml content for certain requess.

```
{
  "fags": 1073872897,
  "sring": "<addressmap-schema>
    <DeviceEntry ipv4=\"X.X.X.X\" device=\"ciscomeraki\" lasUpdated=\"1586804116\"
soft=\"false\"/>
    <DeviceEntry ipv6=\"::1\" device=\"ciscomeraki\" lasUpdated=\"1596696544\"
soft=\"true\"/>
    </addressmap-schema>"
}
```

`Note`: The API Messages section for the path nodes provide more details on API messages available and any associated manual pages.

/collections node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

create

Description: Create an empty collection

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
name	sring, {collection-name:Collection names mus be valid path names}	The name of the collection to be created.
path	sring, optional, {not-empty:Value cannot be empty}	The path of the collection. All database and index folders will be created under this path.

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

mounts

Description: Returns the common mount points for all collections with disk satisics

Security.roles: everyone

/connections node

API Messages

closeAll

Description: Closes all connections

Security.roles: connections.manage

Parameters:

Parameter	Type, Options	Description
type	sring, optional, {enum-any:The value mus be one or more of the following: native|res|amqp}	Indicates the type of connection to close, either native, res or amqp. Default is all types.
limit	uint32, optional	If non-zero, will sop closing connections after limit is reached. NOTE: It sarts closing the mos idle connections frs.
idleLongerThan	sring, optional	A time duration in HH:MM:SS or ##{d,h,m,s} format. Any connection that has been idle longer than the passed in duration is eligible to be closed. Connections that have seen activity within the duration window will not be closed. Examples: "01:05:10" (1 hour, 5 mins, 10 secs) or "65m" (65 minutes)
exceptMe	bool, optional	If true, will not close the connection that initiated the command.

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

netSpeed

Description: Network Speed Tes **Security.roles:**

connections.manage,aggregate

Parameters:

Parameter	Type, Options	Description
size	size, optional, {byte-range:The value (in bytes, size modifers like MB are accepted) mus be between 1024 and 536870912, inclusive}	The size of each message sent, default is 64k
count	uint32, optional, {unsigned:The value mus be between 1 and 536870912, inclusive}	The number of messages to send, default is 16k

/database node

Manual for /database

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

dbState

Description: Returns comprehensive information about the current database sate or persiss the current sate to disk for fas reload **Security.roles:**

database.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: flelis|summary|save}	The operation to perform (flelis, summary or save)
type	sring, {enum-any:The value mus be one or more of the following: session|meta|packet}	The database(s) to operate on
options	sring, optional, {enum-one:The value mus be one of the following: bytes|pretty-print}	Output display options like 'bytes' or 'pretty-print'. Pretty print (the default) will convert all sizes from bytes to human readable quantities like MB or GB.

Manual for dbState

dump

Description: Dumps information out of the database in nwd formatted fles

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
session	uint64	The session id to dump
type	sring, optional, {enum-one:The value mus be one of the following: db|nwd}	The dump type
source	sring, optional, {enum-any:The value mus be one or more of the following: s|m|p}	The types of data to dump, default is all
verbose	bool, optional	If true (default is false), dumps more information
fle	sring, optional	Optional flename to use for NWD type, otherwise flename is assumed to be _sessionid_.nwd
limit	uint32, optional	If assigned, will only output the number of metas/packets passed in. Otherwise will print everything.

hashInfo

Description: Retrieves hash information for database fles that containing session/meta/packet objects for a set of sessions or date range. **Security.roles:**

database.manage **Parameters:**

Parameter	Type, Options	Description
sessions	sring, optional	A comma delimited lis of sessions and session ranges to retrieve fle hashes for.
beginDate	sring, optional	The beginning of a date range (YYYY-MM-DD HH:MM:SS) of database fle creation dates to retrieve fle hashes for.
endDate	sring, optional	The end of a date range (YYYY-MM-DD HH:MM:SS) of database fle creation dates to retrieve fle hashes for.
directories	sring, optional	A semi-colon delimited lis of additional directories to search for hash fles. This may be used to account for changes to the hash.dir confguration.

Manual for hashInfo

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

manifes

Description: If a manifes directory is defned, it will allow operations on the manifes fles (such as a time based query) for database fles in cold sorage.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
op	sring, optional, {enum-one:The value mus be one of the following: query|compress}	The operation to perform (defaults to query)
time1	date-time, optional	The beginning time (UTC) for matching ofine database fles
time2	date-time, optional	The ending time (UTC) for matching ofine database fles
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	Specify the time format that is returned (posix, simple), default is posix

Manual for manifest

optimize

Description: Runs a series of tess to determine the optimimal *.write.block.size based on confgured drives and using data from exising databases.

Security.roles: database.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, optional, {enum-any:The value mus be one or more of the following: session|meta|packet}	The database types to optimize, default is all
size	size, optional, {byte-range:The value (in bytes, size modifers like MB are accepted) mus be between 104857600 and 107374182400, inclusive}	The amount of data to write for each tes
compression	sring, optional, {enum-one:The value mus be one of the following: none|gzip|bzip2|lzma|zsd}	Determines if compression block sizes should be tesed, default is none
compressionLevel	uint32, optional, {unsigned:The value mus be between 0 and 22, inclusive}	Determines the compression level, 0-9, where 1 is fases, 9 is the bes compression (for zsd 22 is the bes compression) and zero is a predetermined balance between speed and compression

reconfg

Description: Calculates new drive sizes and free space for the session, meta and/or packet directories. No directories are removed and the assumption is each directory is mounted on a separate flesysem and will only be used for sorage of that database.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the respective *.dir and *.free.space.min settings, otherwise it will jus output the calculations
type	sring, optional, {enum-any:The value mus be one or more of the following: session|meta|packet}	The database types to reconfgure, default is all
percent	uint32, optional, {unsigned:The value mus be between 1 and 99, inclusive}	The drive percentage to use for the calculated size per directory, default is 98
op	sring, optional, {enum-one:The value mus be one of the following: normal|10g}	The operating mode, 'normal' or '10g'. Default is normal.

Manual for reconfig

resetMax

Description: Resets all max database sats or jus the ones lised.

Security.roles: database.manage **Parameters:**

Parameter	Type, Options	Description
sats	sring, optional, {enum-any:The value mus be one or more of the following: session.rate.max|meta.rate.max|packet.rate.max}	The sats to reset, default is all

sizeRoll

Description: Delete database fles based on the total size of all databases (passed with 'type' parameter) or space remaining on shared volume(s). This command should not be used on databases that don't share sorage.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
type	sring, {enum-any:The value mus be one or more of the following: session|meta|packet}	The databases to consider for removing the oldes data based on total size or space remaining
log	bool, optional	If true, will log all actions taken, otherwise all activity will happen silently. Default is true.
maxSize	size, optional	The maximum size of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxSize. This applies only to the Hot tier.
maxSizeWarm	size, optional	The maximum size of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxSize. This applies only to the Warm tier.
maxPercent	sring, optional	The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxPercent of total volumes. This applies only to the Hot tier.
maxPercentWarm	sring, optional	The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxPercent of total volumes. This applies only to the Warm tier.
minFree	size, optional	The minimum allowed free space on all volumes for all databases in 'type' parameter before oldes data is deleted. When minimum free space drops below target, the oldes data is deleted frs until free space once again exceeds target. This only applies to the Hot tier
minFreeWarm	size, optional	The minimum allowed free space on all volumes for all databases in 'type' parameter before oldes data is deleted. When minimum free space drops below target, the oldes data is deleted frs until free space once again exceeds target. This only applies to the Warm tier

Manual for sizeRoll

sagger

Description: Staggers database fles to optimize read/write performance across multiple volumes. Can be used after adding an empty mount point

Security.roles: database.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, {enum-one:The value mus be one of the following: session|meta|packet}	The databases fles to sagger
dryRun	bool, optional	If true (the default), will only return a description of the operations that would be performed. If false, then the fles will actually be moved to an optimal read/write pattern. Please turn of capture or aggregation before actually performing the sagger operation.

Manual for stagger

timeRoll

Description: Delete database fles that exceed a given age

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
type	sring, {enum-any:The value mus be one or more of the following: session|meta|packet}	The database fles to remove
timeCalc	sring, optional, {enum-one:The value mus be one of the following: current|las-write}	The time calculation to use (current date or las write date), default is current
minutes	uint32, optional, {unsigned:The value mus be between 0 and 10000, inclusive}	Remove database fles older than the given number of minutes
hours	uint32, optional, {unsigned:The value mus be between 0 and 10000, inclusive}	Remove database fles older than the given number of hours
days	uint32, optional, {unsigned:The value mus be between 0 and 10000, inclusive}	Remove database fles older than the given number of days
date	sring, optional	Remove database fles older than the given UTC date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

Manual for timeRoll

wipe

Description: Overwrites all packets and/or meta for a session with a pattern (for eliminating sensitive information). Meta keys sessionid, time and size always remain untouched.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
session	uint64	The session id whose packets will be wiped
payloadOnly	bool, optional	If true (default), will only overwrite the packet payload
pattern	sring, optional	The pattern to use, by default it uses all zeros
metaLis	sring, optional	Comma separated lis of meta to wipe, default (empty) is all meta
source	sring, optional, {enum-any:The value mus be one or more of the following: m|p}	The types of data to wipe, meta and/or packets, default is jus packets

/decoder node

Manual for /decoder

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

logStats

Description: Retrieve count and las receive times for log devices, sources and forwarders.

Security.roles: decoder.manage

Parameters:

Parameter	Type, Options	Description
op	sring, optional, {enum-one:The value mus be one of	The log sats operation. If clear, all log sats sate will be reset and no results will be returned. If save, the current sats are

	the following: clear|remove|save|buckets}	saved for the next time the log decoder is resarted.
deviceFilter	sring, optional	A regex that will only return matching devices.
forwarderFilter	sring, optional	A regex that will only return matching forwarders.
sourceFilter	sring, optional	A regex that will only return matching sources.
discoveredFilter	sring, optional	A regex that will only return matching discovered device types.
timeFilter	sring, optional	This parameter forms a window from the current time which will return any device that falls outside that window. So if you pass 300, it will return all devices that have not sent a log in over 5 minutes. You can pass a hard datetime (YYYY-MM-DD HH:MM:SS) or a time duration (HH:MM:SS or jus number of seconds). Not compatible with the lasReceived parameter.
lasReceived	sring, optional	Compare the las received log time from each device with the passed in operator (
lasUpdated	sring, optional	Compare the las updated log time from each device with the passed in operator (
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	Specify the time format that is returned (posix, simple), default is simple (YYYY-MM-DD HH:MM:SS)

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

reconfg

Description: Calculates optimal settings for decoder pools and bufers based on the insalled hardware.

Security.roles: decoder.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional, {bool:The value mus be one of the following acceptable boolean values: 0,1,yes,no,true,false,on,of }	If true (default is false), will automatically update the respective decoder settings, otherwise it will jus output the calculations
op	sring, optional, {enum-one:The value mus be one of the following: normal|10g|ndr}	The operating mode, 'normal', '10g', or 'ndr'. Default is normal.
memory	size, optional	If non-zero, will use that setting as insalled physical RAM (e.g., memory="24 GiB"). This is useful for hybrids and/or all-in-one machines where the service will need to share memory.

Manual for reconfig

reparse

Description: Reparse a specifed set of packet IDs

Security.roles: decoder.manage **Parameters:**

Parameter	Type, Options	Description
packets	sring	A comma separated lis of packet ids or packet id ranges to reparse
collection	sring, optional	The target collection into which to write the newly created sessions. If this is left blank then the new sessions will be written to the collections determined by the collections rules.
source	sring, optional	The source collection from which to read packets. If left blank then packets will be read from the default collection

reset

Description: Reset data, index, manifess, sats, confguration, or logs for this service. Data automatically deletes index and sats, unless flesCreatedAfter is specifed. Service is automatically resarted. Example arguments:d ata=1 confg=1 log=1T his example will reset data, index, logs, and confguration index=1T his example will reset the index only manifes=1T his example will delete everything in the manifes directory flesCreatedAfter="2015-12-01 14:00:00"T his example will delete all session, meta and packet fles created on or after Dec 1s, 2015 2pm (UTC) from the service. All other fles will remain. The index will not be touched, but upon resart will be truncated to match the las session in the session database. **Security.roles:** decoder.manage

Parameters:

Parameter	Type, Options	Description
data	bool, optional	Reset data, automatically resets index (may not be applicable to

		all services)
index	bool, optional	Reset index, only valid if data is not reset (may not be applicable to all services)
manifes	bool, optional	Reset all manifess sored in the long term manifes directory (may not be applicable to all services)
confg	bool, optional	Reset confguration settings to default
sats	bool, optional	Reset the sats database
log	bool, optional	Reset the log database
flesCreatedAfter	date-time, optional	Delete all database fles (session, meta, packets) created after a certain date. The index will automatically get rolled back on resart. Not valid with option index.
cl	sring, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next resart
parseStats	bool, optional	Reset the parse sats database (PD/LD only)
force	bool, optional	Force reset without confrmation

Manual for reset

resetMax

Description: Resets all max sats to zero. **Security.roles:**

decoder.manage

select

Description: Selected a new capture device

Security.roles: decoder.manage **Parameters:**

Parameter	Type, Options	Description
adapter	sring, optional	The new adapter. If omitted will display a lis of available adapters. You may specify a comma-separated lis of adapter indexes to enable multi-interface capture.

Manual for select

sart

Description: Starts aggregation

Security.roles: decoder.manage Manual for start

sop

Description: Stops aggregation

Security.roles: decoder.manage

Parameters:

Parameter	Type, Options	Description
fush	bool, optional	If true (default), process all sessions and packets currently in memory. Otherwise, sop immediately, discarding unprocessed sessions and packets.

Manual for stop

whoAgg

Description: Returns information on who is aggregating from this service

Security.roles: decoder.manage

Parameters:

Parameter	Type, Options	Description
clean	bool, optional	If true, will delete any aggregation trackers that no longer have a valid channel

Manual for whoAgg

/decoder/parsers node

Manual for /decoder/parsers

API Messages

attrib

Description: Manage attributes for content fles.

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, optional, {enum-one:The value mus be one of the following: lis|update}	The operation to perform. If lis (default), return a lis of attributes for specifed type of contents. If update, set the attributes for specifed content fle.
type	sring, {enum-one:The value mus be one of the following: feeds|parsers|devices}	The content type (feeds|parsers|devices).
fle	sring, optional	The content fle name.
values	sring, optional	Lis of attributes to set for the specifed content fle in name value pairs.

content

Description: Get parsing content fle information.

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
fle	sring, optional	The content flename (type mus be specifed)
type	sring, optional, {enum-one:The value mus be one of the following: feeds|parsers}	The content type
uuid	sring, optional	The content uuid

count

Description: Returns the number of child nodes

Security.roles: everyone

delete

Description: Delete parsing content fles.

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
fle	sring, optional	The content flename (type mus be specifed)
uuid	sring, optional	The content uuid
type	sring, optional, {enum-one:The value mus be one of the following: feeds|parsers|device}	The content type (feeds|parsers|device)

devices

Description: Returns the log parsers

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
uuid	sring, optional	The content uuid

disable

Description: Disable the specifed content.

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, {enum-one:The value mus be one of the following: feeds|parsers|devices}	The content type (feeds|parsers|devices)
uuid	sring	The content uuid

dyndvmap

Description: Retrieve dynamic mappings of sources to device types in log parsing.

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
op	sring, optional, {enum-one:The value mus be one of the following: describe|clear|save}	The operation to perform. If describe (default), return a lising of mappings. If clear, the all mappings will be removed. If save, the mappings will be persised.
deviceFilter	sring, optional	A regex that will only return matching devices.
sourceFilter	sring, optional	A regex that will only return matching sources.
forwarderFilter	sring, optional	A regex that will only return matching forwarders.

Manual for dyndvmap

enable

Description: Enable the specifed content.

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, {enum-one:The value mus be one of the following: feeds|parsers|devices}	The content type (feeds|parsers|devices)
uuid	sring	The content uuid

export

Description: Returns the content as a package.

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, {enum-one:The value mus be one of the following: feeds|parsers|devices}	The content type (feeds|parsers|devices)
uuid	sring, optional	The content uuid
fle	sring, optional	The content fle name

feed

Description: Manages the feed parsers **Security.roles:**

parsers.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: notify|reload|remove|delete}	The feed operation to perform
fle	sring, optional	The feed flename

headers

Description: Return header information about particular devices.

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
devices	sring, optional	A CSV of devices for which to get header information. If omitted, all header information is returned.

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ipdevice

Description: Map IP to Device type in log parsing. Multiple device types mapped to the same ip/hos are prioritized in the order in which they are lised. Takes efect immediately.

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: add|edit|remove|describe}	The operation to perform.
entries	sring, optional	The IP entries. StringParam in format of 'ip=device'. The edit op accepts a preceding '+' or '-' for each param (e.g. '+ip=device').The add op or edit op with + means adding or editing a map entry, the remove op or edit op with
time	sring, optional	Used with 'describe' to return the mapped devices that have been updated AFTER the time specifed (eg. "2010-Apr-20 10:00:00" will return all elements updated after April 20, 2010 10:00).
sources	sring, optional	Used with 'describe' to return the mapped devices of the specifed CSV of sources. (eg. sources="192.168.1.114,192.168.112" will return all mappings from those devices only).
soft	bool, optional	Used with 'op=add'. Indicates that the mappings to be added are soft mappings.

iptmzone

Description: Map IP to time zone in log parsing. Takes efect after parser reload.

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: add|edit|remove|describe}	The operation to perform (edit|describe).
entries	sring, optional	The IP entries. StringParam in format of '+/-ip=
mdformat	sring, optional, {enum-one:The value mus be one of the following: none|mdy|dmy}	Override the parsing order of Month/Day or Day/Month (none|mdy|dmy). In Month/Day/Year or Day/Month/Year sequences, the parsing order of Month and Day can reversed to match the order observed in the messages. A value of 'dmy' will override parsing to be Day/Month/Year. A value of 'mdy' will perform the reverse.

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

patterns

Description: API for interfacing UI managed content nodes on the Log Decoder

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
op	sring, optional	Use 'parse' to get detailed information about how parse rules are parsed, otherwise leave blank for default behavior.
devices	sring, optional	A CSV of device names or '*'. If empty, a lis of all devices that have at leas one content node defned is returned. If '*', a lis of content nodes for all devices. Otherwise, the content nodes of the requesed devices.
sems	sring, optional, {enum-any:The value mus be one or more of the following: default|cusom|tokens}	A CSV of source content flename sems to lis. Default is all sems.
showDisabled	bool, optional	Include disabled rules. Default is false.
type	sring, optional, {enum-one:The value mus be one of the following: RULE|DataType}	If not specifed then returns all content types, otherwise the name of the content type requesed.

reload

Description: Reloads all meta parsers **Security.roles:**

parsers.manage

reset

Description: Reset content of the specifed type (feeds|parsers|devices|all or 1). Content type can be specifed in csv format e.g content="feeds,parsers,devices".

Content type 'devices' is only valid for LogDecoder. Example arguments:c ontent="feeds,parsers"T his example will reset content for types feeds and parsers.

content=1 or content=allT his example will reset content for all types i.e feeds,parsers,devices. **Security.roles:** parsers.manage **Parameters:**

Parameter	Type, Options	Description
content	sring	The content type (feeds|parsers|devices|all or 1)
force	bool, optional	Force content reset without confrmation

schema

Description: Returns the parsers meta schema

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
onlyShowEnabled	bool, optional	If true, only return information on parsers that are currently enabled
showRegiseredMeta	bool, optional	If true (the default), show the regisered meta for each parser
showOptions	bool, optional	If true, shows the options supported by each parser. Passing true changes the XML sructure returned so each parser (potentially) has
prettyPrint	bool, optional	If true, returns the XML formatted to be easier to read. Default is false.

satHis

Description: Retrieve hisorical sats from the sats db. Don't send time1/time2 to get bounding times about sats db. Supported wildcards are ? to match any single char, * to match zero or more characters, not including slash /, ** to match zero or more characters including slash /. **Security.roles:** parsers.manage

Parameters:

Parameter	Type, Options	Description
time1	date-time, optional	The sarting time for retrieving sats
time2	date-time, optional	The ending time for retrieving sats
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	Specify the time format for each sat snapshot, default is posix (seconds since 1970)
include	sring, optional	Comma separated lis of sats to include (wildcards allowed)
exclude	sring, optional	Comma separated lis of sats to exclude (wildcards allowed, has precedence over include)
onChange	sring, optional	Comma separated lis of sats that reduce the result set to only when at leas one of the sats change from their previous value
reduce	bool, optional	If true, reduces data transmission by replacing sat pathnames with a shorthand and provides a lookup table as frs result
showAll	bool, optional	If true, changes each result to show all sats previously seen, even if the value hasn't changed. It has a moderate performance impact and is bes used with include, exclude or onChange parameters to reduce the returned result set size.

tzinfo

Description: Timezones available for normalizing timesamps parsed from log events.

Security.roles: parsers.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: version|tznames|abbreviations|duplicates|overrides|add|remove}	The operation to perform.
abbr	sring, optional	A timezone abbreviation to add or remove to the overrides lis. This parameter is required for the add and remove operations.
gmtofset	sring, optional	The ofset from GMT of a timezone abbreviation to add to the override lis. Format is [-+]HH[MM].
tzname	sring, optional	A timezone name to override the default timezone for a duplicate abbreviation. The named timezone mus exis and the abbreviation mus be valid for that timezone (see tznames).

Manual for tzinfo

upload

Description: Upload feed/parser/geoip2 fles to this service

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: sart|attributes|cancel|fnished}	The operation to perform (sart, cancel, fnished)
size	uint64, optional	The size of the incoming fle
flename	sring, optional	The name of the incoming fle
fnalCount	uint32, optional	Las chunk count. To be used with fnished operation
reload	uint32, optional	Reload all parsers when upload is complete for the appropriate fle types (default). A setting of 0 when sending a fnished op will disable the reload.

/index node

API Messages

cacheClr

Description: Clear index internal caches **Security.roles:**

index.manage

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

inspect

Description: Provides overall information about the index or can inspect specifc keys and values

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
key	sring, optional	The key to inspect
value	sring, optional	The value to inspect
format	sring, optional, {enum-one:The value mus be one of the following: Text|IPv4|Int8|UInt8|Int16|UInt16|Int32|UInt32|Int64|UInt64|UInt128|Float32|Float64|TimeT|DayOfWeek|HourOfDay|Binary|IPv6|MAC}	Only include keys that have this format
valueCount	uint64, optional	The number of values to return from each key
allSlices	bool, optional	Return information on all slices, insead of jus mos recently created slice
summarizeAllValues	bool, optional	Calculate total summary and page usage from all values in each slice

language

Description: Returns language information

Security.roles: index.manage **Parameters:**

Parameter	Type, Options	Description
values	bool, optional	If set to 1, will describe the value overrides defned in the language

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

profle

Description: Returns information about the index used by the Index Profle tool

Security.roles: index.manage **Parameters:**

Parameter	Type, Options	Description
key	sring, optional	The key to profle
value	sring, optional	The value to profle
max	uint32, optional	Max number of pages to return
fag	uint32, optional	Bitwise option when key is specifed. 1: Break report to values. 2: Lis page sizes. 4: Categorize page counts in diferent range. 8: Sessions count Per page with each algorithm. 16: Verify session page

reconfg

Description: Calculates new sizes for the summary and page directories. NOTE: Any size changes require a reset of the index before they take efect.

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the respective confg node settings, otherwise it will jus output the calculations
percent	uint32, optional, {unsigned:The	The drive percentage to use for the index database, default is 90
	value mus be between 1 and 99, inclusive}	

save

Description: Saves the index to disk **Security.roles:**

index.manage

schema

Description: Introspects all open Meta DB fles for meta keys and merges those keys with the defned language to return a fairly comprehensive

lis of all meta keys in exisence. **Security.roles:** index.manage

sizeRoll

Description: Delete index and other DB fles (see 'type' param) based on the total size or space remaining on shared volumes. This command is not intended to be used on databases that do not share sorage.

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
log	bool, optional	If true, will log all actions taken, otherwise all activity will happen silently. Default is true.
type	sring, optional, {enum-any:The value mus be one or more of the following: session|meta|packet}	The databases to consider for removing the oldes data based on total size or space remaining
maxSize	size, optional	The maximum size of the index hot sorage. When exceeded, oldes data is deleted frs until total size is less than maxSize.
maxSizeWarm	size, optional	The maximum size of the index warm sorage. When exceeded, oldes data is delete frs until total size is less that maxSizeWarm
maxPercent	sring, optional	The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxPercent of total volumes. This parameter applies to hot sorage only.
maxPercentWarm	sring, optional	The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxPercent of total volumes. This parameter applies to warm sorage only.
minFree	size, optional	The minimum allowed free space on the volumes before oldes data is deleted. This parameter applies to hot sorage only.
minFreeWarm	size, optional	The minimum allowed free space on the volumes before oldes data is deleted. This parameter applies to warm sorage only.

values

Description: Returns the top n values for the given key

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
key	sring	The index key for values
top	uint32, {unsigned:The value mus be between 0 and 10000, inclusive}	The number of top entries to return

/logs node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

download

Description: Downloads log entries

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The sarting log number to download
id2	uint64, optional	The ending log number to download
time1	date-time, optional	The sarting log date to download, can be posix time (seconds since 1970) or DD-MM-YY HH:MM:SS
time2	date-time, optional	The ending log date to download, can be posix time (seconds since 1970) or DD-MM-YY HH:MM:SS
op	sring, {enum-one:The value mus be one of the following: sart|next|cancel}	The operation to perform (sart, next, cancel)
logTypes	sring, optional	The type(s) of log messages to retrieve (debug,info,audit,warning,failure)
match	sring, optional	Case insensitive sring to match in each log message
regex	sring, optional	Regular expression to match in each log message
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	The time format used in each log message, default is posix time (seconds since 1970)
batchSize	uint32, optional, {unsigned:The value mus be between 2000 and 10000, inclusive}	The number of logs to retrieve per batch, default is 2000

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

pull

Description: Downloads N log entries

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The frs log id number to retrieve, this is mutually exclusive with id2
id2	uint64, optional	The las log id number that will be sent, defaults to mos recent log message when id1 or id2 is not sent
count	uint32, optional, {unsigned:The value mus be between 1 and 10000, inclusive}	The number of logs to pull
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	The time format used in each log message, default is posix time (seconds since 1970)

timeRoll

Description: Delete log entries that exceed a given age

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
timeCalc	sring, optional, {enum-any:The value mus be one or more of the following: current|las-write}	The time calculation to use (current date or las write date), default is current
minutes	uint32, optional, {unsigned:The value mus be between 0 and 60, inclusive}	Remove log entries older than the given number of minutes
hours	uint32, optional, {unsigned:The value mus be between 0 and 24, inclusive}	Remove log entries older than the given number of hours
days	uint32, optional, {unsigned:The value mus be between 0 and 365, inclusive}	Remove log entries older than the given number of days
date	sring, optional	Remove log entries older than the given date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

/res node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated
		values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

/sdk node

Manual for /sdk

API Messages

aliases

Description: Retrieves aliases for the specifed key

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
key	sring	The language key to retrieve aliases for
value	sring, optional	An optional value to return a specifc alias for

cancel

Description: Cancel a running query, regardless of the user

Security.roles: sdk.manage,sdk.meta **Parameters:**

Parameter	Type, Options	Description
handle	uint32	The channel handle running the query
uuid	sring, optional	The device UUID if taking a device ofine insead of canceling the entire query (Broker only). Passing uuid will propagate the reques upsream to fnd the frs matching device to cancel.

content

Description: Returns the packet content for a session

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
session	uint64, optional	The session id to retrieve packet content for
packet	uint64, optional	The packet id of the frs packet of the session, in case the session has rolled out
pinId	sring, optional	The Pin ID of the session to retrieve from long term sorage. If provided, the packet and session parameters mus not be

		provided.
maxSize	size, optional	The max number of bytes to return, zero means no limit
failOnCacheMiss	bool, optional	If true, the content call fails if the NWD is not in the cache. This is used by optimized caching code only.
renderType	sring, optional	The render type (see NwContentTypes in NwSDK.h), pass zero to pick bes option. Not specifying this parameter always results in an NWD and renderFlags and renderOptions are not considered. The following values are also supported: 'pcap', 'fle-lis-json', 'session-meta-fle-lis-json', 'nwd', 'log', 'fles'
renderFlags	uint32, optional	Bitwise mask to control options, (see NwContentFlags in NwSDK.h)
renderOptions	sring, optional	An encoded sring params containing additional options for controlling the operation.

Manual for content

count

Description: Returns the number of child nodes **Security.roles:** everyone

delCache

Description: Removes all cached NWD fles from either the normal NWD cache or the pin cache

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
type	sring, optional, {enum-one:The value mus be one of the following: nwd|pin|all}	Specifes which cache to delete all NWD fles from. The default is 'nwd'. Passing 'all' will delete from all caches.

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')

op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

hierarch

Description: Report the set of devices attached to this service

Security.roles: sdk.meta Manual for

hierarch **info**

Description: Returns detailed information about the node **Security.roles:** everyone

keyrefs

Description: Retrieves the set of entity keys

Security.roles: sdk.meta Manual for

keyrefs

language

Description: Retrieves the language defnition

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The sarting language id
id2	uint64, optional	The ending language id
time1	date-time, optional	The sarting time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS)
time2	date-time, optional	The ending time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS)
options	sring, optional	Extra options, currently unused
fags	sring, optional	Optional fags to confgure how the results are returned. Can be a number (bitwise mask) or comma separated values like default, no-count, quick-count or full-count.
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query

Manual for language **ls**

Description: Returns the lis of child nodes

	regex|sp|sm|ci|pre|pos|ds|precache|si}	regex,sp,sm,ci,pre,pos,ds,precache,si
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

msearch

Description: Search for pattern matches in many sessions or packets

Security.roles: sdk.content & sdk.meta

Parameters:

Parameter	Type, Options	Description
sessions	sring, optional	The session ID ranges to search
packets	sring, optional	The packet ID ranges to search
search	sring	The search sring to use
where	sring, optional	The where clause used to identify sessions to consider for the search
limit	uint64, optional	The maximum number of sessions to traverse for this search
size	uint64, optional	The maximum number of results to return for this search
fags	sring, {enum-any:The value mus be one or more of the following:	Flags to use for search. This is a comma separated lis of one or more of the fag values:

Manual for msearch

packets

Description: Stream packets back based on the input parameters provided

Security.roles: sdk.content & sdk.packets **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: setup|sart|cancel|processed}	The operation to perform (sart, cancel, processed)
sessions	sring, optional	A comma separated lis of session ids or session id ranges (#-#) whose packets will be sreamed back.
packets	sring, optional	A comma separated lis of packet ids or session/packet ids (#&#) that will be sreamed back.
where	sring, optional	Where clause which will be evaluated to determine which sessions to sream back
time1	date-time, optional	A sarting time range (UTC) where matching packets will be sreamed back ("2010-Apr-20 09:00:00")
time2	date-time, optional	An ending time range (UTC) where matching packets will be sreamed back ("2010-Apr-20 10:00:00")
fags	uint32, optional	Additional fags as defned by the NwPackets SDK function
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query

Manual for packets

pin

Description: Pin a session in the long term cache so it can be retrieved even after the meta or packets have rolled out. To retrieve, you mus make a content call with the Pin ID that is returned from this command. The op={unpin,validate} commands all take one or more comma separated Pin IDs in the pinId parameter and returns the satus for the Pin IDs that were recognized by a service.

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
session	uint64, optional	The session ID to pin. This cannot be used with the 'sessions' parameter.
sessions	sring, optional	A comma separated lis of session ids or session id ranges (#-#) to pin. This cannot be used with the 'session' parameter.
pinId	sring, optional	One or more Pin IDs, separated by commas
op	sring, optional, {enum-one:The value mus be one of the following: ls|unpin|validate}	Various operations that can be performed. Will be submitted to upsream devices if not confgured for long term cache.

Manual for pin

precache

Description: Efciently caches NWD fles for future retrieval via the content message

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
sessions	sring, optional	A comma separated lis of session ids or session id ranges (#-#) that will be saved in the NWD cache directory
packets	sring, optional	A comma separated lis of session and packet ids (#p#) that will be saved in the NWD cache directory
waitOnComplete	bool, optional	If true, will wait for precache to complete before returning a response.

query

Description: Performs a query agains the meta database

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The sarting meta id (to run the query from mos recent to oldes meta, make id1 larger than id2)
id2	uint64, optional	The ending meta id
size	uint32, optional	The max number of entries to return, or jus sream back all results if zero
query	sring, optional	The query sring to use
fags	sring, optional	The fags to use for query. Can be a number (bitwise mask) or comma separated values like query-log.
threshold	uint64, optional	Query optimization to sop processing results after the threshold is reached (useful with select aggregate functions). Zero means no threshold (the default).
partition	uint32, optional	Used to partition the result set across multiple clients. Mus be greater than zero and less than or equal to totalPartitions
totalPartitions	uint32, optional	The total number of clients that will be submitting the same query for partitioning.
search	sring, optional	A text search clause to apply to the where clause of this query
sreamLimit	uint32, optional	Limit the number of sessions returned by a sreaming query
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query

Manual for query

reconfg

Description: Calculates default values for some of the confg nodes.

Security.roles: sdk.manage **Parameters:**

Parameter	Type, Options	Description
session	uint64	The session id to search
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the confg nodes with the defaults
memory	size, optional	If non-zero, will use that setting as insalled physical RAM (e.g., memory="24 GiB"). This is useful for hybrids and/or all-in-one machines where the service will need to share memory.

search

search	sring	The search sring to use

Description: Searches for matches in session/packet content

Security.roles: sdk.content

Parameters:
Manual for search

session

Description: Retrieves the meta id range for the session range

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
id1	uint64	The sarting session id (inclusive). Pass zero to scope to lowes valid id.
id2	uint64	The ending session id (inclusive). Pass zero to scope to highes valid id.

Manual for session

summary

Description: Retrieves summary information from the databases

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
fags	sring, optional	Optional SDK fags. Can be a number (bitwise mask) or comma separated values like default or ignore-cache.

Manual for summary

timeline

Description: Returns the count of sessions/size/packets in discrete time intervals

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
time1	date-time, optional	The sarting time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS), if greater than time2, then the whole time range is returned
time2	date-time, optional	The ending time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS)
timezone	int32, optional	The timezone of the local computer to receive the data, hour

		ofset from GMT [-13 to 13]
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
fags	sring, optional	The fags to use for the timeline. Can be a number (bitwise mask) or comma separated values like sessions, size, packets, order-ascending, order-descending, ignore-cache or clear-cache.
where	sring, optional	Optional where clause for fltering the data
bookendHours	uint32, optional	The number of hours outside the given time1, time2 which will become part of the cache. Default is 24 hours.
feldName	sring, optional	The time feld to retrieve values for, defaults to 'time'
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
search	sring, optional	A text search clause to apply to the where clause of this query

Manual for timeline **validate**

Description: Validate a query or values call without execution

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: query|values}	The operation to validate
validateLanguage	bool, optional	If true (default is false), language tokens and syntax will be validated, otherwise jus syntax will be validated
feldName	sring, optional	Required if validating a 'values' operation. The feld to retrieve values for.
where	sring, optional	Required if validating a 'values' operation. The flter criteria to be validated.
query	sring, optional	Required if validating a 'query' operation. The query sring to be validated.
json	bool, optional	Structure response in JSON syntax

Manual for validate

values

Description: Performs a value count query and returns the matching values for a report

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The sarting meta id
id2	uint64, optional	The ending meta id
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
fags	sring, optional	The fags to use for values. Can be a number (bitwise mask) or comma separated values like sessions, size, packets, aggregate, sort-total, sort-value, order-ascending, order-descending, ignore-cache, clear-cache, or database-scan.
threshold	uint64, optional	Query optimization to sop processing large session counts
feldName	sring	The feld to retrieve values for
where	sring, optional	The flter criteria for the values
aggregateFunction	sring, optional, {enum-one:The value mus be one of the following: count|sum}	The aggregate operation to be applied where the aggregate fag is set
aggregateFieldName	sring, optional	The meta feld to aggregate in the aggregateFunction
min	sring, optional	The lower limit of values to return. Only values greater than this value will be returned.
max	sring, optional	The upper limit of values to return. Only values less than this value will be returned.
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
search	sring, optional	A text search clause to apply to the where clause of this query

Manual for values

xforms

Description: Retrieves transforms for the specifed key

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
key	sring, optional	An optional language key to retrieve transform for, all transforms returned if not provided

/sys node

Manual for /sys

API Messages

caCert

Description: Display or delete trused CA certs

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: lis|delete|add}	CA action to perform (lis,delete,add)
id	sring, optional	ID of CA certifcate to remove when op=delete

Manual for caCert

count

Description: Returns the number of child nodes **Security.roles:** everyone

fleEdit

Description: View and edit sysem confguration fles

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: dir|get|getBackup|put|create|delete}	The operation to perform (dir, get, getBackup, put, create, delete)
type	sring, optional, {enum-one:The value mus be one of the following: parser|flter}	The type of fle, if not provided will check fle extension
flename	sring, optional, {length:The value length mus be between 1 and 64, inclusive}	File to create, edit, get or delete

Manual for fileEdit

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

peerCert

Description: Display or modify trused peer certifcates

Security.roles: sys.manage
Parameters:

Parameter	Type, Options	Description
op	sring, optional, {enum-one:The value mus be one of the following: lis|delete|add}	Operation to perform on the lis of trused peers (lis,delete,add)
id	sring, optional	ID of peer certifcate to remove when op=delete

Manual for peerCert

save

Description: Forces a save of sysem settings

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
force	uint32, optional	If 1, forces the confguration fle to saved, regardless if changes are detected, default is 0

Manual for save

servCert

Description: Display the current server certifcate in PEM format.

Security.roles: sys.manage Manual for

servCert

shutdown

Description: Stop the service

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
reason	sring, optional, {length:The value length mus be between 0 and 512, inclusive}	The reason for shutting down the service
cl	sring, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next resart

Manual for shutdown

satHis

Description: Retrieve hisorical sats from the sats db. Don't send time1/time2 to get bounding times about sats db. Supported wildcards are ? to match any single char, * to match zero or more characters, not including slash /, ** to match zero or more characters including slash /. **Security.roles:** sys.manage

Parameters:

Parameter	Type, Options	Description
time1	date-time, optional	The sarting time for retrieving sats
time2	date-time, optional	The ending time for retrieving sats
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	Specify the time format for each sat snapshot, default is posix (seconds since 1970)
include	sring, optional	Comma separated lis of sats to include (wildcards allowed)
exclude	sring, optional	Comma separated lis of sats to exclude (wildcards allowed, has precedence over include)
onChange	sring, optional	Comma separated lis of sats that reduce the result set to only when at leas one of the sats change from their previous value
reduce	bool, optional	If true, reduces data transmission by replacing sat pathnames with a shorthand and provides a lookup table as frs result
showAll	bool, optional	If true, changes each result to show all sats previously seen, even if the value hasn't changed. It has a moderate performance impact and is bes used with include, exclude or onChange parameters to reduce the returned result set size.

Manual for statHist

telemtry

Description: Returns telemetry information about the service

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
level	uint32, optional, {signed:The value mus be between 1 and 3, inclusive}	The level of information to be sent by the telemetry reques. A value of 1 (default) will send all sat nodes, a value of 2 will send sat and confg nodes, and a value of 3 will send sat nodes, confg nodes and usage metrics for rules, parsers and devices (where applicable).
include	sring, optional	Comma separated lis of nodes to include (wildcard usage is ok).
exclude	sring, optional	Comma separated lis of nodes to exclude (wildcard usage is ok). If "include" is also specifed, exclusions are applied to the set of nodes specifed by "include".
url	sring, optional	Endpoint URL to send the JSON response.
op	sring, optional, {enum-one:The value mus be one of the following: get|confgure}	The telemetry operation to execute. Default is 'get'.

/users node

API Messages

addOrMod

Description: Add a new user or update an exising user in the sysem

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#$%&'+-=?^_`{|}~.[]\}	The username to add or update, mus be alphanumeric or @!#$%&'+-=?^_`{|}~.[]\
password	sring, optional	The user's password
pwdIsHashed	bool, optional	Set to true if the password has already been hashed
groups	sring, optional	The groups the user belongs to
authType	sring, optional, {enum-one:The value mus be one of the following: netwitness|pam}	The authentication sysem to use, "netwitness" is the default
queryTimeout	uint32, optional, {unsigned:The value mus be between 0 and 43200, inclusive}	The maximum number of minutes that a query is allowed to execute for this user. Zero means unlimited.
displayName	sring, optional	Display name for this user account
email	sring, optional	Email address

		for this account
description	sring, optional	Description of this user account
queryPrefx	sring, optional	Query filter applied to every query performed by this user account
sessionThreshold	uint32, optional, {unsigned:The value mus be between 0 and 4294967295, inclusive}	Query optimization which will extrapolate the remaining session counts when they exceed this value

auths

Description: Get supported authentication types **Security.roles:** everyone

count

Description: Returns the number of child nodes **Security.roles:** everyone

delete

Description: Delete a user from the sysem

Security.roles: users.manage **Parameters:**

Parameter	Type, Options	Description
name	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#$%&'+-=?^_`{|}~.[]\}	The username to delete

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone
Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

unlock

Description: Unlock a locked out account

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	sring	The username or account to unlock

whoAmI

Description: Returns information about the authenticated user

Security.roles: everyone

Concentrator service

Introduction

Concentrator sores and indexes metadata for fas queries and retrieval of raw data capture. It does not capture data, it aggregates the sessions and meta from a Decoder/LogDecoder and indexes them.
Typically the index on a concentrator is much larger and this meant to do the heavy lifting for queries, leaving the Decoders focused on capture and parsing. When the raw data (packets or logs) are requesed from a Concentrator, the reques is forwarded to the Decoder with the data. The response (with the raw data) is then forwarded back to the client. In this respect, raw data access is seamless from the client's point of view, even though the Concentrator never sores the data. It simply knows where the data is kept, requess the data on the client's behalf and sreams it back seamlessly.

API Usage

Concentrator service provides API interface through its tree form as nodes, node messages and child nodes.

The following are few examples of the several API nodes and messages supported by Concentrator:

- /concentrator node messages:
  - `start` and `stop` aggregation `reconfig` concentrator for optimal settings based on memory
  - and hardware `add` a device to concentrator
  - `/concentrator/stats` node for concentrator aggregation stats information /sdk node
- messages:
  - `reconfig` sdk for optimal settings
  - `summary` to retrieve summary information from the databases
  - `values` to perform a value count query and return the matching values for a report `query` to
  - perform query against meta database `language` of meta keys for key info, format, description
  - etc...
  - `/sdk/stats` node for queries active, pending and other sdk operations /index node
- messages:
  - `language` of meta keys index level, valueMax, format, description etc...
  - `inspect` for overall information about the index `save` for saving
  - index to disk
  - `/index/stats` node for index stats , slices, memory usage, reindex status etc...
- /database node messages:
  - `reconfig` database for optimal database settings `dbState` to get comprehensive
  - information about current databases
  - `/database/stats` node for database stats information /sys node
- messages:
  - `statHist` to retrieve historical stats from stats database `/sys/stats` node for
  - service and system stats information

Sample Reques

The following is a sample reques to get session count in sdk query on the concentrator.

Reques path is `/sdk` and URL reques parameters for message query are `msg=query query="select count(sessionid) where did='decoder-pdec-0'" size=5`.

For queries that are expected to take longer time on execution an additional URL parameter `expiry=0` can be added to disable default REST reques timeout and continue with user profle timeout.

Using URL encoded reques

```
$ curl "https://concentrator-hos:50105/sdk?
msg=query&query=select%20count(sessionid)%20where%20did%3D%27decoder-pdec-
0%27&size%3D5&expiry=0" \
    -H 'Accept: application/json;charset=UTF-8' \
    -H 'Content-Type: application/x-www-form-urlencoded; charset=ISO-8859-1' \
    -u 'username:password'
```

Using special content type

```
Content Type: application/x-netwitness-sring-params

$ curl 'https://concentrator-hos:50105/sdk?msg=query&expiry=0' \
    -H 'Accept: application/json;charset=UTF-8' \
    -H 'Content-Type: application/x-netwitness-sring-params' \
    -d "query=\"select count(sessionid) where did='decoder-pdec-0'\" size=5"
    -u 'username:password'
```

Sample Response

HTTP Headers

```
HTTP/1.1 200 OK
Content-Length: 186
Connection: Keep-Alive
Pragma: no-cache
Expires: -1
Cache-Control: no-cache, no-sore, mus-revalidate
Content-Type: application/json
```

JSON Response

```
{
    "fags": 1074200577,
    "results": {
        "id1": 3216363401111,
```

```
        "id2": 3216363401110,
        "felds": [
            {
                "id1": 0,
                "id2": 0,
                "count": 6864785,
                "format": 8,
                "type": "sessionid",
                "fags": 2,
                "group": 0,
                "value": "6864785"
            }
        ]
    }
}
```

`Note` : The API Messages section for the path nodes provide more details on API messages available and any associated manual pages.

/concentrator node

API Messages

add

Description: Add a new device for aggregation

Security.roles: concentrator.manage **Parameters:**

Parameter	Type, Options	Description
device	sring, optional, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to add (ip:port or [ipv6]:port), either this or ip mus be set
ip	sring, optional	The IP address of the device, either this or device mus be set
port	uint32, optional, {unsigned:The value mus be between 0 and 65535, inclusive}	The port number the device is lisening on
username	sring	The username to run as
password	sring, optional	The account password. Assumes service trus aggregation if no password is provided.
options	sring, optional	The device options
ssl	sring, optional	If true, the device requires SSL to connect
backup	sring, optional	If true, the device will be used as a backup
reconnect	sring, optional	If true, will automatically reconnect when not connected
name	sring, optional	Device collection name if device is a collection

count

Description: Returns the number of child nodes **Security.roles:** everyone

delete

Description: Delete an exising aggregation device

Security.roles: concentrator.manage

Parameters:

Parameter	Type, Options	Description
device	sring, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to delete
name	sring, optional	Device collection name if device is a collection

edit

Description: Edit an exising aggregation device

Security.roles: concentrator.manage **Parameters:**

Parameter	Type, Options	Description
device	sring, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to edit
username	sring, optional	The username to run as
password	sring, optional	The account password. For service trus aggregation you mus not provide this parameter
options	sring, optional	The device options
ssl	sring, optional	If true, the device requires SSL to connect
backup	sring, optional	If true, the device will be used as a backup
reconnect	sring, optional	If true, will automatically reconnect when not connected
name	sring, optional	Device collection name if device is a collection
switchToTrusedAuth	bool, optional	If true, removes any password associated with this account and will use trused authentication henceforth. When switching to trused, only the 'device' parameter is allowed.

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

reconfg

Description: Calculates optimal settings for concentrator pools and bufers based on the insalled hardware.

Security.roles: concentrator.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the respective concentrator settings, otherwise it will jus output the calculations
memory	size, optional	If non-zero, will use that setting as insalled physical RAM (e.g., memory="24 GiB"). This is useful for hybrids and/or all-in-one machines where the service will need to share memory.

reset

Description: Reset data, index, manifess, sats, confguration, or logs for this service. Data automatically deletes index and sats, unless flesCreatedAfter is specifed. Service is automatically resarted. Example arguments: data=1 confg=1 log=1 This example will reset data, index, logs, and confguration index=1 This example will reset the index only manifes=1 This example will delete everything in the manifes directory flesCreatedAfter="2015-12-01 14:00:00" This example will delete all session, meta and packet fles created on or after Dec 1s, 2015 2pm (UTC) from the service. All other fles will remain. The index will not be touched, but upon resart will be truncated to match the las session in the session database. **Security.roles:** concentrator.manage **Parameters:**

Parameter	Type, Options	Description
data	bool, optional	Reset data, automatically resets index (may not be applicable to all services)
index	bool, optional	Reset index, only valid if data is not reset (may not be applicable to all services)
manifes	bool, optional	Reset all manifess sored in the long term manifes directory (may not be applicable to all services)
confg	bool, optional	Reset confguration settings to default
sats	bool, optional	Reset the sats database
log	bool, optional	Reset the log database
flesCreatedAfter	date-time, optional	Delete all database fles (session, meta, packets) created after a certain date. The index will automatically get rolled back on resart. Not valid with option index.
cl	sring, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next resart
parseStats	bool, optional	Reset the parse sats database (PD/LD only)
force	bool, optional	Force reset without confrmation

resetMax

Description: Resets all max sats, including device max sats back to zero. **Security.roles:**

concentrator.manage

sart

Description: Starts aggregation

Security.roles: concentrator.manage

satus

Description: Change the online/ofine satus of an aggregation device

Security.roles: concentrator.manage **Parameters:**

Parameter	Type, Options	Description
device	sring, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to change satus
satus	sring, optional, {enum-one:The value mus be one of the following: online|ofine}	Whether to take device ofine or put online
name	sring, optional	Device collection name if device is a collection

sop

Description: Stops aggregation

Security.roles: concentrator.manage

whoAgg

Description: Returns information on who is aggregating from this service

Security.roles: concentrator.manage

Parameters:

Parameter	Type, Options	Description
clean	bool, optional	If true, will delete any aggregation trackers that no longer have a valid channel

/connections node

API Messages

closeAll

Description: Closes all connections

Security.roles: connections.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, optional, {enum-any:The value mus be one or more of the following: native|res|amqp}	Indicates the type of connection to close, either native, res or amqp. Default is all types.
limit	uint32, optional	If non-zero, will sop closing connections after limit is reached. NOTE: It sarts closing the mos idle connections frs.
idleLongerThan	sring, optional	A time duration in HH:MM:SS or ##{d,h,m,s} format. Any connection that has been idle longer than the passed in duration is eligible to be closed. Connections that have seen activity within the duration window will not be closed. Examples: "01:05:10" (1 hour, 5 mins, 10 secs) or "65m" (65 minutes)
exceptMe	bool, optional	If true, will not close the connection that initiated the command.

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

netSpeed

Description: Network Speed Tes **Security.roles:**

connections.manage,aggregate

Parameters:

Parameter	Type, Options	Description
size	size, optional, {byte-range:The value (in bytes, size modifers like MB are accepted) mus be between 1024 and 536870912, inclusive}	The size of each message sent, default is 64k
count	uint32, optional, {unsigned:The value mus be between 1 and 536870912, inclusive}	The number of messages to send, default is 16k

/database node

Manual for /database

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

dbState

Description: Returns comprehensive information about the current database sate or persiss the current sate to disk for fas reload **Security.roles:**

database.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: flelis|summary|save}	The operation to perform (flelis, summary or save)
type	sring, {enum-any:The value mus be one or more of the following: session|meta|packet}	The database(s) to operate on
options	sring, optional, {enum-one:The value mus be one of the following: bytes|pretty-print}	Output display options like 'bytes' or 'pretty-print'. Pretty print (the default) will convert all sizes from bytes to human readable quantities like MB or GB.

Manual for dbState

dump

Description: Dumps information out of the database in nwd formatted fles

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
session	uint64	The session id to dump
type	sring, optional, {enum-one:The value mus be one of the following: db|nwd}	The dump type
source	sring, optional, {enum-any:The value mus be one or more of the following: s|m|p}	The types of data to dump, default is all
verbose	bool, optional	If true (default is false), dumps more information
fle	sring, optional	Optional flename to use for NWD type, otherwise flename is assumed to be _sessionid_.nwd
limit	uint32, optional	If assigned, will only output the number of metas/packets passed in. Otherwise will print everything.

hashInfo

Description: Retrieves hash information for database fles that containing session/meta/packet objects for a set of sessions or date range. **Security.roles:**

database.manage **Parameters:**

Parameter	Type, Options	Description
sessions	sring, optional	A comma delimited lis of sessions and session ranges to retrieve fle hashes for.
beginDate	sring, optional	The beginning of a date range (YYYY-MM-DD HH:MM:SS) of database fle creation dates to retrieve fle hashes for.
endDate	sring, optional	The end of a date range (YYYY-MM-DD HH:MM:SS) of database fle creation dates to retrieve fle hashes for.
directories	sring, optional	A semi-colon delimited lis of additional directories to search for hash fles. This may be used to account for changes to the hash.dir confguration.

Manual for hashInfo

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

manifes

Description: If a manifes directory is defned, it will allow operations on the manifes fles (such as a time based query) for database fles in cold sorage.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
op	sring, optional, {enum-one:The value mus be one of the following: query|compress}	The operation to perform (defaults to query)
time1	date-time, optional	The beginning time (UTC) for matching ofine database fles
time2	date-time, optional	The ending time (UTC) for matching ofine database fles
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	Specify the time format that is returned (posix, simple), default is posix

Manual for manifest

optimize

Description: Runs a series of tess to determine the optimimal *.write.block.size based on confgured drives and using data from exising databases.

Security.roles: database.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, optional, {enum-any:The value mus be one or more of the following: session|meta}	The database types to optimize, default is all
size	size, optional, {byte-range:The value (in bytes, size modifers like MB are accepted) mus be between 104857600 and 107374182400, inclusive}	The amount of data to write for each tes
compression	sring, optional, {enum-one:The value mus be one of the following: none|gzip|bzip2|lzma|zsd}	Determines if compression block sizes should be tesed, default is none
compressionLevel	uint32, optional, {unsigned:The value mus be between 0 and 22, inclusive}	Determines the compression level, 0-9, where 1 is fases, 9 is the bes compression (for zsd 22 is the bes compression) and zero is a predetermined balance between speed and compression

reconfg

Description: Calculates new drive sizes and free space for the session, meta and/or packet directories. No directories are removed and the assumption is each directory is mounted on a separate flesysem and will only be used for sorage of that database.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the respective *.dir and *.free.space.min settings, otherwise it will jus output the calculations
type	sring, optional, {enum-any:The value mus be one or more of the following: session|meta}	The database types to reconfgure, default is all
percent	uint32, optional, {unsigned:The value mus be between 1 and 99, inclusive}	The drive percentage to use for the calculated size per directory, default is 98
op	sring, optional, {enum-one:The value mus be one of the following: normal|10g}	The operating mode, 'normal' or '10g'. Default is normal.

Manual for reconfig

resetMax

Description: Resets all max database sats or jus the ones lised.

Security.roles: database.manage **Parameters:**

Parameter	Type, Options	Description
sats	sring, optional, {enum-any:The value mus be one or more of the following: session.rate.max|meta.rate.max}	The sats to reset, default is all

sizeRoll

Description: Delete database fles based on the total size of all databases (passed with 'type' parameter) or space remaining on shared volume(s). This command should not be used on databases that don't share sorage.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
type	sring, {enum-any:The value mus be one or more of the following: session|meta}	The databases to consider for removing the oldes data based on total size or space remaining
log	bool, optional	If true, will log all actions taken, otherwise all activity will happen silently. Default is true.
maxSize	size, optional	The maximum size of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxSize. This applies only to the Hot tier.
maxSizeWarm	size, optional	The maximum size of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxSize. This applies only to the Warm tier.
maxPercent	sring, optional	The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxPercent of total volumes. This applies only to the Hot tier.
maxPercentWarm	sring, optional	The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxPercent of total volumes. This applies only to the Warm tier.
minFree	size, optional	The minimum allowed free space on all volumes for all databases in 'type' parameter before oldes data is deleted. When minimum free space drops below target, the oldes data is deleted frs until free space once again exceeds target. This only applies to the Hot tier
minFreeWarm	size, optional	The minimum allowed free space on all volumes for all databases in 'type' parameter before oldes data is deleted. When minimum free space drops below target, the oldes data is deleted frs until free space once again exceeds target. This only applies to the Warm tier

Manual for sizeRoll

sagger

Description: Staggers database fles to optimize read/write performance across multiple volumes. Can be used after adding an empty mount point

Security.roles: database.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, {enum-one:The value mus be one of the following: session|meta}	The databases fles to sagger
dryRun	bool, optional	If true (the default), will only return a description of the operations that would be performed. If false, then the fles will actually be moved to an optimal read/write pattern. Please turn of capture or aggregation before actually performing the sagger operation.

Manual for stagger

timeRoll

Description: Delete database fles that exceed a given age

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
type	sring, {enum-any:The value mus be one or more of the following: session|meta}	The database fles to remove
timeCalc	sring, optional, {enum-one:The value mus be one of the following: current|las-write}	The time calculation to use (current date or las write date), default is current
minutes	uint32, optional, {unsigned:The value mus be between 0 and 10000, inclusive}	Remove database fles older than the given number of minutes
hours	uint32, optional, {unsigned:The value mus be between 0 and 10000, inclusive}	Remove database fles older than the given number of hours
days	uint32, optional, {unsigned:The value mus be between 0 and 10000, inclusive}	Remove database fles older than the given number of days
date	sring, optional	Remove database fles older than the given UTC date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

Manual for timeRoll

wipe

Description: Overwrites all packets and/or meta for a session with a pattern (for eliminating sensitive information). Meta keys sessionid, time and size always remain untouched.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
session	uint64	The session id whose packets will be wiped
payloadOnly	bool, optional	If true (default), will only overwrite the packet payload
pattern	sring, optional	The pattern to use, by default it uses all zeros
metaLis	sring, optional	Comma separated lis of meta to wipe, default (empty) is all meta
source	sring, optional, {enum-any:The value mus be one or more of the following: m|p}	The types of data to wipe, meta and/or packets, default is jus packets

/index node

API Messages

cacheClr

Description: Clear index internal caches **Security.roles:**

index.manage

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

inspect

Description: Provides overall information about the index or can inspect specifc keys and values

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
key	sring, optional	The key to inspect
value	sring, optional	The value to inspect
format	sring, optional, {enum-one:The value mus be one of the following: Text|IPv4|Int8|UInt8|Int16|UInt16|Int32|UInt32|Int64|UInt64|UInt128|Float32|Float64|TimeT|DayOfWeek|HourOfDay|Binary|IPv6|MAC}	Only include keys that have this format
valueCount	uint64, optional	The number of values to return from each key
allSlices	bool, optional	Return information on all slices, insead of jus mos recently created slice
summarizeAllValues	bool, optional	Calculate total summary and page usage from all values in each slice

language

Description: Returns language information

Security.roles: index.manage **Parameters:**

Parameter	Type, Options	Description
values	bool, optional	If set to 1, will describe the value overrides defned in the language

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

profle

Description: Returns information about the index used by the Index Profle tool

Security.roles: index.manage **Parameters:**

Parameter	Type, Options	Description
key	sring, optional	The key to profle
value	sring, optional	The value to profle
max	uint32, optional	Max number of pages to return
fag	uint32, optional	Bitwise option when key is specifed. 1: Break report to values. 2: Lis page sizes. 4: Categorize page counts in diferent range. 8: Sessions count Per page with each algorithm. 16: Verify session page

reconfg

Description: Calculates new sizes for the summary and page directories. NOTE: Any size changes require a reset of the index before they take efect.

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the respective confg node settings, otherwise it will jus output the calculations
percent	uint32, optional, {unsigned:The	The drive percentage to use for the index database, default is 90
	value mus be between 1 and 99, inclusive}	

save

Description: Saves the index to disk **Security.roles:**

index.manage

schema

Description: Introspects all open Meta DB fles for meta keys and merges those keys with the defned language to return a fairly comprehensive

lis of all meta keys in exisence. **Security.roles:** index.manage

sizeRoll

Description: Delete index and other DB fles (see 'type' param) based on the total size or space remaining on shared volumes. This command is not intended to be used on databases that do not share sorage.

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
log	bool, optional	If true, will log all actions taken, otherwise all activity will happen silently. Default is true.
type	sring, optional, {enum-any:The value mus be one or more of the following: session|meta}	The databases to consider for removing the oldes data based on total size or space remaining
maxSize	size, optional	The maximum size of the index hot sorage. When exceeded, oldes data is deleted frs until total size is less than maxSize.
maxSizeWarm	size, optional	The maximum size of the index warm sorage. When exceeded, oldes data is delete frs until total size is less that maxSizeWarm
maxPercent	sring, optional	The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxPercent of total volumes. This parameter applies to hot sorage only.
maxPercentWarm	sring, optional	The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When exceeded, oldes data is deleted frs until total size is less than maxPercent of total volumes. This parameter applies to warm sorage only.
minFree	size, optional	The minimum allowed free space on the volumes before oldes data is deleted. This parameter applies to hot sorage only.
minFreeWarm	size, optional	The minimum allowed free space on the volumes before oldes data is deleted. This parameter applies to warm sorage only.

values

Description: Returns the top n values for the given key

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
key	sring	The index key for values
top	uint32, {unsigned:The value mus be between 0 and 10000, inclusive}	The number of top entries to return

/logs node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

download

Description: Downloads log entries

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The sarting log number to download
id2	uint64, optional	The ending log number to download
time1	date-time, optional	The sarting log date to download, can be posix time (seconds since 1970) or DD-MM-YY HH:MM:SS
time2	date-time, optional	The ending log date to download, can be posix time (seconds since 1970) or DD-MM-YY HH:MM:SS
op	sring, {enum-one:The value mus be one of the following: sart|next|cancel}	The operation to perform (sart, next, cancel)
logTypes	sring, optional	The type(s) of log messages to retrieve (debug,info,audit,warning,failure)
match	sring, optional	Case insensitive sring to match in each log message
regex	sring, optional	Regular expression to match in each log message
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	The time format used in each log message, default is posix time (seconds since 1970)
batchSize	uint32, optional, {unsigned:The value mus be between 2000 and 10000, inclusive}	The number of logs to retrieve per batch, default is 2000

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

pull

Description: Downloads N log entries

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The frs log id number to retrieve, this is mutually exclusive with id2
id2	uint64, optional	The las log id number that will be sent, defaults to mos recent log message when id1 or id2 is not sent
count	uint32, optional, {unsigned:The value mus be between 1 and 10000, inclusive}	The number of logs to pull
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	The time format used in each log message, default is posix time (seconds since 1970)

timeRoll

Description: Delete log entries that exceed a given age

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
timeCalc	sring, optional, {enum-any:The value mus be one or more of the following: current|las-write}	The time calculation to use (current date or las write date), default is current
minutes	uint32, optional, {unsigned:The value mus be between 0 and 60, inclusive}	Remove log entries older than the given number of minutes
hours	uint32, optional, {unsigned:The value mus be between 0 and 24, inclusive}	Remove log entries older than the given number of hours
days	uint32, optional, {unsigned:The value mus be between 0 and 365, inclusive}	Remove log entries older than the given number of days
date	sring, optional	Remove log entries older than the given date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

/res node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated
		values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

/sdk node

Manual for /sdk

API Messages

aliases

Description: Retrieves aliases for the specifed key

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
key	sring	The language key to retrieve aliases for
value	sring, optional	An optional value to return a specifc alias for

cancel

Description: Cancel a running query, regardless of the user

Security.roles: sdk.manage,sdk.meta **Parameters:**

Parameter	Type, Options	Description
handle	uint32	The channel handle running the query
uuid	sring, optional	The device UUID if taking a device ofine insead of canceling the entire query (Broker only). Passing uuid will propagate the reques upsream to fnd the frs matching device to cancel.

content

Description: Returns the packet content for a session

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
session	uint64, optional	The session id to retrieve packet content for
packet	uint64, optional	The packet id of the frs packet of the session, in case the session has rolled out
pinId	sring, optional	The Pin ID of the session to retrieve from long term sorage. If provided, the packet and session parameters mus not be

		provided.
maxSize	size, optional	The max number of bytes to return, zero means no limit
failOnCacheMiss	bool, optional	If true, the content call fails if the NWD is not in the cache. This is used by optimized caching code only.
renderType	sring, optional	The render type (see NwContentTypes in NwSDK.h), pass zero to pick bes option. Not specifying this parameter always results in an NWD and renderFlags and renderOptions are not considered. The following values are also supported: 'pcap', 'fle-lis-json', 'session-meta-fle-lis-json', 'nwd', 'log', 'fles'
renderFlags	uint32, optional	Bitwise mask to control options, (see NwContentFlags in NwSDK.h)
renderOptions	sring, optional	An encoded sring params containing additional options for controlling the operation.

Manual for content

count

Description: Returns the number of child nodes **Security.roles:** everyone

delCache

Description: Removes all cached NWD fles from either the normal NWD cache or the pin cache

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
type	sring, optional, {enum-one:The value mus be one of the following: nwd|pin|all}	Specifes which cache to delete all NWD fles from. The default is 'nwd'. Passing 'all' will delete from all caches.

deviceId

Description: Given a session id, this message retrieves the session id and the device it maps to

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
session	uint64	The session id that will be translated into the device name and session id.

Manual for deviceId

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

hierarch

Description: Report the set of devices attached to this service

Security.roles: sdk.meta Manual for

hierarch **info**

Description: Returns detailed information about the node **Security.roles:** everyone

keyrefs

Description: Retrieves the set of entity keys

Security.roles: sdk.meta Manual for

keyrefs

language

Description: Retrieves the language defnition

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The sarting language id

id2	uint64, optional	The ending language id
time1	date-time, optional	The sarting time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS)
time2	date-time, optional	The ending time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS)
options	sring, optional	Extra options, currently unused
fags	sring, optional	Optional fags to confgure how the results are returned. Can be a number (bitwise mask) or comma separated values like default, no-count, quick-count or full-count.
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query

Manual for language **ls**

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

msearch

Description: Search for pattern matches in many sessions or packets

Security.roles: sdk.content & sdk.meta

Parameters:

Parameter	Type, Options	Description

sessions	sring, optional	The session ID ranges to search
packets	sring, optional	The packet ID ranges to search
search	sring	The search sring to use
where	sring, optional	The where clause used to identify sessions to consider for the search
limit	uint64, optional	The maximum number of sessions to traverse for this search
size	uint64, optional	The maximum number of results to return for this search
fags	sring, {enum-any:The value mus be one or more of the following: regex|sp|sm|ci|pre|pos|ds|precache|si}	Flags to use for search. This is a comma separated lis of one or more of the fag values: regex,sp,sm,ci,pre,pos,ds,precache,si
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query

Manual for msearch

packets

Description: Stream packets back based on the input parameters provided

Security.roles: sdk.content & sdk.packets

Parameters:

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: setup|sart|cancel|processed}	The operation to perform (sart, cancel, processed)
sessions	sring, optional	A comma separated lis of session ids or session id ranges (#-#) whose packets will be sreamed back.
packets	sring, optional	A comma separated lis of packet ids or session/packet ids (#&#) that will be sreamed back.
where	sring, optional	Where clause which will be evaluated to determine which sessions to sream back
time1	date-time, optional	A sarting time range (UTC) where matching packets will be sreamed back ("2010-Apr-20 09:00:00")
time2	date-time, optional	An ending time range (UTC) where matching packets will be sreamed back ("2010-Apr-20 10:00:00")
fags	uint32, optional	Additional fags as defned by the NwPackets SDK function
queryPriority	int32, optional, {signed:The value mus be zero OR	The query priority for this query

	between -20 and 20, inclusive}	
devices	sring, optional	Optional lis of comma separated device names that will be queried. If this parameter is not present, then all devices are queried. The device names mus match the lis defned under /concentrator/devices
excludeDevices	sring, optional	Optional lis of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs mus match the UUID defned under /sys/sats/uuid of each device to be queried.

Manual for packets

pin

Description: Pin a session in the long term cache so it can be retrieved even after the meta or packets have rolled out. To retrieve, you mus make a content call with the Pin ID that is returned from this command. The op={unpin,validate} commands all take one or more comma separated Pin IDs in the pinId parameter and returns the satus for the Pin IDs that were recognized by a service.

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
session	uint64, optional	The session ID to pin. This cannot be used with the 'sessions' parameter.
sessions	sring, optional	A comma separated lis of session ids or session id ranges (#-#) to pin. This cannot be used with the 'session' parameter.
pinId	sring, optional	One or more Pin IDs, separated by commas
op	sring, optional, {enum-one:The value mus be one of the following: ls|unpin|validate}	Various operations that can be performed. Will be submitted to upsream devices if not confgured for long term cache.

Manual for pin

precache

Description: Efciently caches NWD fles for future retrieval via the content message

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
sessions	sring, optional	A comma separated lis of session ids or session id ranges (#-#) that will be saved in the NWD cache directory
packets	sring, optional	A comma separated lis of session and packet ids (#p#) that will be saved in the NWD cache directory
waitOnComplete	bool, optional	If true, will wait for precache to complete before returning a response.

query

Description: Performs a query agains the meta database

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The sarting meta id (to run the query from mos recent to oldes meta, make id1 larger than id2)
id2	uint64, optional	The ending meta id
size	uint32, optional	The max number of entries to return, or jus sream back all results if zero
query	sring, optional	The query sring to use
fags	sring, optional	The fags to use for query. Can be a number (bitwise mask) or comma separated values like query-log.
threshold	uint64, optional	Query optimization to sop processing results after the threshold is reached (useful with select aggregate functions). Zero means no threshold (the default).
partition	uint32, optional	Used to partition the result set across multiple clients. Mus be greater than zero and less than or equal to totalPartitions
totalPartitions	uint32, optional	The total number of clients that will be submitting the same query for partitioning.
search	sring, optional	A text search clause to apply to the where clause of this query
sreamLimit	uint32, optional	Limit the number of sessions returned by a sreaming query
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query

Manual for query

reconfg

Description: Calculates default values for some of the confg nodes.

Security.roles: sdk.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the confg nodes with the defaults
memory	size, optional	If non-zero, will use that setting as insalled physical RAM (e.g., memory="24 GiB"). This is useful for hybrids and/or all-in-one machines where the service will need to share memory.

search

Description: Searches for matches in session/packet content

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
session	uint64	The session id to search
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
search	sring	The search sring to use

Manual for search

session

Description: Retrieves the meta id range for the session range

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
id1	uint64	The sarting session id (inclusive). Pass zero to scope to lowes valid id.
id2	uint64	The ending session id (inclusive). Pass zero to scope to highes valid id.

Manual for session

summary

Description: Retrieves summary information from the databases

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
fags	sring, optional	Optional SDK fags. Can be a number (bitwise mask) or comma separated values like default or ignore-cache.

Manual for summary **timeline**

Description: Returns the count of sessions/size/packets in discrete time intervals

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
time1	date-time, optional	The sarting time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS), if greater than time2, then the whole time range is returned
time2	date-time, optional	The ending time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS)
timezone	int32, optional	The timezone of the local computer to receive the data, hour ofset from GMT [-13 to 13]
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
fags	sring, optional	The fags to use for the timeline. Can be a number (bitwise mask) or comma separated values like sessions, size, packets, order-ascending, order-descending, ignore-cache or clear-cache.
where	sring, optional	Optional where clause for fltering the data
bookendHours	uint32, optional	The number of hours outside the given time1, time2 which will become part of the cache. Default is 24 hours.
feldName	sring, optional	The time feld to retrieve values for, defaults to 'time'
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
search	sring, optional	A text search clause to apply to the where clause of this query

Manual for timeline

validate

Description: Validate a query or values call without execution

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: query|values}	The operation to validate
validateLanguage	bool, optional	If true (default is false), language tokens and syntax will be validated, otherwise jus syntax will be validated
feldName	sring, optional	Required if validating a 'values' operation. The feld to retrieve values for.
where	sring, optional	Required if validating a 'values' operation. The flter criteria to be validated.
query	sring, optional	Required if validating a 'query' operation. The query sring to be validated.
json	bool, optional	Structure response in JSON syntax

Manual for validate

values

Description: Performs a value count query and returns the matching values for a report

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The sarting meta id
id2	uint64, optional	The ending meta id
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
fags	sring, optional	The fags to use for values. Can be a number (bitwise mask) or comma separated values like sessions, size, packets, aggregate, sort-total, sort-value, order-ascending, order-descending, ignore-cache, clear-cache, or database-scan.
threshold	uint64, optional	Query optimization to sop processing large session counts
feldName	sring	The feld to retrieve values for
where	sring, optional	The flter criteria for the values
aggregateFunction	sring, optional, {enum-one:The value mus be one of the following: count|sum}	The aggregate operation to be applied where the aggregate fag is set
aggregateFieldName	sring, optional	The meta feld to aggregate in the aggregateFunction

min	sring, optional	The lower limit of values to return. Only values greater than this value will be returned.
max	sring, optional	The upper limit of values to return. Only values less than this value will be returned.
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
search	sring, optional	A text search clause to apply to the where clause of this query

Manual for values

xforms

Description: Retrieves transforms for the specifed key

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
key	sring, optional	An optional language key to retrieve transform for, all transforms returned if not provided

/sys node

Manual for /sys

API Messages

caCert

Description: Display or delete trused CA certs

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: lis|delete|add}	CA action to perform (lis,delete,add)
id	sring, optional	ID of CA certifcate to remove when op=delete

Manual for caCert

count

Description: Returns the number of child nodes **Security.roles:** everyone

fleEdit

Description: View and edit sysem confguration fles

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: dir|get|getBackup|put|create|delete}	The operation to perform (dir, get, getBackup, put, create, delete)
type	sring, optional, {enum-one:The value mus be one of the following: parser|flter}	The type of fle, if not provided will check fle extension
flename	sring, optional, {length:The value length mus be between 1 and 64, inclusive}	File to create, edit, get or delete

Manual for fileEdit

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

peerCert

Description: Display or modify trused peer certifcates

Security.roles: sys.manage
Parameters:

Parameter	Type, Options	Description
op	sring, optional, {enum-one:The value mus be one of the following: lis|delete|add}	Operation to perform on the lis of trused peers (lis,delete,add)
id	sring, optional	ID of peer certifcate to remove when op=delete

Manual for peerCert

save

Description: Forces a save of sysem settings

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
force	uint32, optional	If 1, forces the confguration fle to saved, regardless if changes are detected, default is 0

Manual for save

servCert

Description: Display the current server certifcate in PEM format.

Security.roles: sys.manage Manual for

servCert

shutdown

Description: Stop the service

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
reason	sring, optional, {length:The value length mus be between 0 and 512, inclusive}	The reason for shutting down the service
cl	sring, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next resart

Manual for shutdown

satHis

Description: Retrieve hisorical sats from the sats db. Don't send time1/time2 to get bounding times about sats db. Supported wildcards are ? to match any single char, * to match zero or more characters, not including slash /, ** to match zero or more characters including slash /. **Security.roles:** sys.manage

Parameters:

Parameter	Type, Options	Description
time1	date-time, optional	The sarting time for retrieving sats
time2	date-time, optional	The ending time for retrieving sats
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	Specify the time format for each sat snapshot, default is posix (seconds since 1970)
include	sring, optional	Comma separated lis of sats to include (wildcards allowed)
exclude	sring, optional	Comma separated lis of sats to exclude (wildcards allowed, has precedence over include)
onChange	sring, optional	Comma separated lis of sats that reduce the result set to only when at leas one of the sats change from their previous value
reduce	bool, optional	If true, reduces data transmission by replacing sat pathnames with a shorthand and provides a lookup table as frs result
showAll	bool, optional	If true, changes each result to show all sats previously seen, even if the value hasn't changed. It has a moderate performance impact and is bes used with include, exclude or onChange parameters to reduce the returned result set size.

Manual for statHist

telemtry

Description: Returns telemetry information about the service

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
level	uint32, optional, {signed:The value mus be between 1 and 3, inclusive}	The level of information to be sent by the telemetry reques. A value of 1 (default) will send all sat nodes, a value of 2 will send sat and confg nodes, and a value of 3 will send sat nodes, confg nodes and usage metrics for rules, parsers and devices (where applicable).
include	sring, optional	Comma separated lis of nodes to include (wildcard usage is ok).
exclude	sring, optional	Comma separated lis of nodes to exclude (wildcard usage is ok). If "include" is also specifed, exclusions are applied to the set of nodes specifed by "include".
url	sring, optional	Endpoint URL to send the JSON response.
op	sring, optional, {enum-one:The value mus be one of the following: get|confgure}	The telemetry operation to execute. Default is 'get'.

/users node

API Messages

addOrMod

Description: Add a new user or update an exising user in the sysem

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#$%&'+-=?^_`{|}~.[]\}	The username to add or update, mus be alphanumeric or @!#$%&'+-=?^_`{|}~.[]\
password	sring, optional	The user's password
pwdIsHashed	bool, optional	Set to true if the password has already been hashed
groups	sring, optional	The groups the user belongs to
authType	sring, optional, {enum-one:The value mus be one of the following: netwitness|pam}	The authentication sysem to use, "netwitness" is the default
queryTimeout	uint32, optional, {unsigned:The value mus be between 0 and 43200, inclusive}	The maximum number of minutes that a query is allowed to execute for this user. Zero means unlimited.
displayName	sring, optional	Display name for this user account
email	sring, optional	Email address

		for this account
description	sring, optional	Description of this user account
queryPrefx	sring, optional	Query filter applied to every query performed by this user account
sessionThreshold	uint32, optional, {unsigned:The value mus be between 0 and 4294967295, inclusive}	Query optimization which will extrapolate the remaining session counts when they exceed this value

auths

Description: Get supported authentication types **Security.roles:** everyone

count

Description: Returns the number of child nodes **Security.roles:** everyone

delete

Description: Delete a user from the sysem

Security.roles: users.manage **Parameters:**

Parameter	Type, Options	Description
name	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#$%&'+-=?^_`{|}~.[]\}	The username to delete

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone
Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

unlock

Description: Unlock a locked out account

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	sring	The username or account to unlock

whoAmI

Description: Returns information about the authenticated user

Security.roles: everyone

Broker service

Introduction

Broker aggregates and sore database ranges from multiple core devices and provides a Concentrator like SDK API to treat multiple devices as a single query-able entity. In common deployments Brokers aggregate from multiple Concentrators and sub Brokers.

Broker does not aggregate data like session and meta.

API Usage

Broker service provides API interface through its tree form as nodes, node messages and child nodes.

The following are few examples of the several API nodes and messages supported by Broker:

- /broker node messages:
  - `start` and `stop` aggregation `repair` to reset the global summary based on the
  - device ranges `add` a new device for aggregation
  - `/broker/stats` node for broker aggregation stats information /sdk node
- messages:
  - `reconfig` sdk for optimal settings `summary` to retrieve summary information from the database
  - ranges `values` to perform a value count query and return the matching values for a report `query`
  - to perform query against meta database of upstream devices `language` of meta keys for key info,
  - format, description etc...
  - `/sdk/stats` node for queries active, pending and other sdk operations /index node
- messages:
  - `ranges` to show sessionid ranges for all devices `gaps` to show the gaps and session
  - virtual range mappings per device `save` to save broker device mappings to disk /sys
  - node messages:
- `statHist` to retrieve historical stats from stats database
  - `/sys/stats` node for service system stats information

Sample Reques

The following is a sample reques to get top 3 values of ip.src through sdk values on the Broker.
Reques path is `/sdk`, message is `msg=values` and parameters are `fieldName=ip.src where=\"time=\'2024-08-16 16:30:00\' - \'2024-08-16 16:34:59\'\" size=3 flags=sessions,sort-total,order-descending threshold=100000`

For queries that are expected to take longer time on execution an additional URL parameter `expiry=0` can be added to disable default REST reques timeout and continue with user profle timeout.

Using URL encoded reques

```
$ curl 'https://broker-hos:50103/sdk?msg=values&feldName=ip.src%20where=\"time=\'2024-08-16%2016:30:00\'%20-%20\'2024-08-16%2016:34:59\'\"&size=3&fags=sessions,sort-total,order-descending&threshold=100000&expiry=0' \
    -H 'Accept: application/json;charset=UTF-8' \
    -H 'Content-Type: application/x-www-form-urlencoded; charset=ISO-8859-1' \
    -u 'username:password'
```

Using special content type

```
Content Type: application/x-netwitness-sring-params

$ curl 'https://broker-hos:50103/sdk?msg=values&expiry=0' \
    -H 'Accept: application/json;charset=UTF-8' \
    -H 'Content-Type: application/x-netwitness-sring-params' \
    -d "feldName=ip.src where=\"time=\'2024-08-16 16:30:00\' - \'2024-08-16 16:34:59\'\"
size=3 fags=sessions,sort-total,order-descending threshold=100000" \
    -u 'username:password'
```

Sample Response

HTTP Headers

```
HTTP/1.1 200 OK
Content-Length: 769
Connection: Keep-Alive
Pragma: no-cache
Expires: -1
Cache-Control: no-cache, no-sore, mus-revalidate
Content-Type: application/json
```

JSON Response

```
{
  "fags": 1074200577,
  "results": {
    "id1": 0,
    "id2": 0,
    "felds": [
      {
        "id1": 79390446,
        "id2": 85993604,
        "count": 104240,
        "format": 128,
        "type": "ip.src",
```

```
        "fags": 1,
        "group": 100000,
        "value": "10.XX.XX.81"
      },
      {
        "id1": 79115795,
        "id2": 85854695,
        "count": 101206,
        "format": 128,
        "type": "ip.src",
        "fags": 1,
        "group": 100000,
        "value": "10.XX.XX.83"
      },
      {
        "id1": 79114441,
        "id2": 85709910,
        "count": 100845,
        "format": 128,
        "type": "ip.src",
        "fags": 1,
        "group": 100000,
        "value": "10.XX.XX.152"
      }
    ]
  },
  "resultsInfo": {
    "devices": "XX.XX.XX.85:50005=on XX.XX.XX.84:56005=on",
    "ofineDeviceCount": "0",
    "onlineDeviceCount": "2",
    "deviceElapsedTime": "XX.XX.XX.85:50005=00:00:00 XX.XX.XX.84:56005=00:00:00",
    "uuidMapping": "XX.XX.XX.85:50005=7bae6dd3-c277-4db8-ba06-edc77298601c
XX.XX.XX.84:56005=58794d2a-e3f-4a23-a931-c9e0fbc2ebc8"
  }
}
```


Sample Reques

The following is a sample reques to select 3 values of ip.src in ssl trafc originating from a country using sdk query on the Broker. Reques path is `/sdk`, message is `msg=query` and parameters are `query="select ip.src where time=\'2024-08-16 18:50:00\' - \'2024-08-16 18:54:59\' && tcp.dstport=443 && country.src='United States'" size=3`

Using special content type

```
$ curl 'https://broker-hos:50103/sdk?msg=values&expiry=0' \
    -H 'Accept: application/json;charset=UTF-8' \
    -H 'Content-Type: application/x-netwitness-sring-params' \
    -d "query=\"select ip.src where time=\'2024-08-16 18:50:00\' - \'2024-08-16 18:54:59\'
&& tcp.dsport=443 && country.src='United States'\" size=3" \
    -u 'username:password'
```

Sample Response

HTTP Headers

```
HTTP/1.1 200 OK
Content-Length: 786
Connection: Keep-Alive
Pragma: no-cache
Expires: -1
Cache-Control: no-cache, no-sore, mus-revalidate
Content-Type: application/json
```

JSON Response


```
{
  "fags": 1074200577,
  "results": {
    "id1": 2009111894,
    "id2": 2370947558,
    "felds": [
      {
        "id1": 2009111743,
        "id2": 2009111743,
        "count": 0,
        "format": 128,
        "type": "ip.src",
        "fags": 0,
        "group": 79178639,
        "value": "11.XX.XX.152"
      },
      {
        "id1": 2009111773,
        "id2": 2009111773,
        "count": 0,
        "format": 128,
        "type": "ip.src",
        "fags": 0,
        "group": 79178640,
        "value": "11.XX.XX.152"
      },
      {
        "id1": 2009111893,
```

```
        "id2": 2009111893,
        "count": 0,
        "format": 128,
        "type": "ip.src",
        "fags": 0,
        "group": 79178644,
        "value": "11.XX.XX.198"
      }
    ]
  },
  "resultsInfo": {
    "devices": "10.XX.XX.85:50005=on 10.XX.XX84:56005=on",
    "ofineDeviceCount": "0",
    "onlineDeviceCount": "2",
    "deviceElapsedTime": "10.XX.XX.85:50005=00:00:00 10.XX.XX.84:56005=00:00:00",
    "uuidMapping": "10.XX.XX.85:50005=7bae6dd3-c277-4db8-ba06-edc77298601c
10.XX.XX.84:56005=58794d2a-e3f-4a23-a931-c9e0fbc2ebc8"
  }
}
```


`Note` : The API Messages section for the path nodes provide more details on API messages available and any associated manual pages.

/broker node

API Messages

add

Description: Add a new device for aggregation

Security.roles: concentrator.manage **Parameters:**

Parameter	Type, Options	Description
device	sring, optional, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to add (ip:port or [ipv6]:port), either this or ip mus be set
ip	sring, optional	The IP address of the device, either this or device mus be set
port	uint32, optional, {unsigned:The value mus be between 0 and 65535, inclusive}	The port number the device is lisening on
username	sring	The username to run as
password	sring, optional	The account password. Assumes service trus aggregation if no password is provided.
options	sring, optional	The device options
ssl	sring, optional	If true, the device requires SSL to connect
backup	sring, optional	If true, the device will be used as a backup
reconnect	sring, optional	If true, will automatically reconnect when not connected
name	sring, optional	Device collection name if device is a collection

count

Description: Returns the number of child nodes **Security.roles:** everyone

delete

Description: Delete an exising aggregation device

Security.roles: concentrator.manage

Parameters:

Parameter	Type, Options	Description
device	sring, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to delete
name	sring, optional	Device collection name if device is a collection

edit

Description: Edit an exising aggregation device

Security.roles: concentrator.manage **Parameters:**

Parameter	Type, Options	Description
device	sring, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to edit
username	sring, optional	The username to run as
password	sring, optional	The account password. For service trus aggregation you mus not provide this parameter
options	sring, optional	The device options
ssl	sring, optional	If true, the device requires SSL to connect
backup	sring, optional	If true, the device will be used as a backup
reconnect	sring, optional	If true, will automatically reconnect when not connected
name	sring, optional	Device collection name if device is a collection
switchToTrusedAuth	bool, optional	If true, removes any password associated with this account and will use trused authentication henceforth. When switching to trused, only the 'device' parameter is allowed.

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

repair

Description: Reset the global summary based on the device ranges **Security.roles:**

index.manage

reset

Description: Reset data, index, manifess, sats, confguration, or logs for this service. Data automatically deletes index and sats, unless flesCreatedAfter is specifed. Service is automatically resarted. Example arguments:d ata=1 confg=1 log=1T his example will reset data, index, logs, and confguration index=1T his example will reset the index only manifes=1T his example will delete everything in the manifes directory flesCreatedAfter="2015-12-01 14:00:00"T his example will delete all session, meta and packet fles created on or after Dec 1s, 2015 2pm (UTC) from the service. All other fles will remain. The index will not be touched, but upon resart will be truncated to match the las session in the session database.

Security.roles: concentrator.manage

Parameters:

Parameter	Type, Options	Description
data	bool, optional	Reset data, automatically resets index (may not be applicable to all services)
index	bool, optional	Reset index, only valid if data is not reset (may not be applicable to all services)
manifes	bool, optional	Reset all manifess sored in the long term manifes directory (may not be applicable to all services)
confg	bool, optional	Reset confguration settings to default
sats	bool, optional	Reset the sats database
log	bool, optional	Reset the log database
flesCreatedAfter	date-time, optional	Delete all database fles (session, meta, packets) created after a certain date. The index will automatically get rolled back on resart. Not valid with option index.
cl	sring, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next resart
parseStats	bool, optional	Reset the parse sats database (PD/LD only)
force	bool, optional	Force reset without confrmation

resetMax

Description: Resets all max sats, including device max sats back to zero. **Security.roles:**

concentrator.manage

sart

Description: Starts aggregation

Security.roles: concentrator.manage

satus

Description: Change the online/ofine satus of an aggregation device

Security.roles: concentrator.manage

Parameters:

Parameter	Type, Options	Description
device	sring, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to change satus
satus	sring, optional, {enum-one:The value mus be one of the following: online|ofine}	Whether to take device ofine or put online
name	sring, optional	Device collection name if device is a collection

sop

Description: Stops aggregation

Security.roles: concentrator.manage

whoAgg

Description: Returns information on who is aggregating from this service

Security.roles: concentrator.manage

Parameters:

Parameter	Type, Options	Description
clean	bool, optional	If true, will delete any aggregation trackers that no longer have a valid channel

/connections node

API Messages

closeAll

Description: Closes all connections

Security.roles: connections.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, optional, {enum-any:The value mus be one or more of the following: native|res|amqp}	Indicates the type of connection to close, either native, res or amqp. Default is all types.
limit	uint32, optional	If non-zero, will sop closing connections after limit is reached. NOTE: It sarts closing the mos idle connections frs.
idleLongerThan	sring, optional	A time duration in HH:MM:SS or ##{d,h,m,s} format. Any connection that has been idle longer than the passed in duration is eligible to be closed. Connections that have seen activity within the duration window will not be closed. Examples: "01:05:10" (1 hour, 5 mins, 10 secs) or "65m" (65 minutes)
exceptMe	bool, optional	If true, will not close the connection that initiated the command.

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

netSpeed

Description: Network Speed Tes **Security.roles:**

connections.manage,aggregate

Parameters:

Parameter	Type, Options	Description
size	size, optional, {byte-range:The value (in bytes, size modifers like MB are accepted) mus be between 1024 and 536870912, inclusive}	The size of each message sent, default is 64k
count	uint32, optional, {unsigned:The value mus be between 1 and 536870912, inclusive}	The number of messages to send, default is 16k

/index node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

gaps

Description: Shows the gaps and other error checking in session virtual range mappings per device

Security.roles: index.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The sarting session id
id2	uint64, optional	The ending session id

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node

Security.roles: everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

ranges

Description: Shows the session id ranges for all devices or a single device

Security.roles: index.manage **Parameters:**

Parameter	Type, Options	Description
device	sring, optional	The device to report ranges for, defaults to 'all' devices
name	sring, optional	The device collection name if the device is a collection
id1	uint64, optional	The sarting session id
id2	uint64, optional	The ending session id
format	sring, optional, {enum-one:The value mus be one of the following: tab|comma}	The format of the returned report

save

Description: Saves the broker device mappings to disk

Security.roles: index.manage

/logs node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

download

Description: Downloads log entries

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The sarting log number to download
id2	uint64, optional	The ending log number to download
time1	date-time, optional	The sarting log date to download, can be posix time (seconds since 1970) or DD-MM-YY HH:MM:SS
time2	date-time, optional	The ending log date to download, can be posix time (seconds since 1970) or DD-MM-YY HH:MM:SS
op	sring, {enum-one:The value mus be one of the following: sart|next|cancel}	The operation to perform (sart, next, cancel)
logTypes	sring, optional	The type(s) of log messages to retrieve (debug,info,audit,warning,failure)
match	sring, optional	Case insensitive sring to match in each log message
regex	sring, optional	Regular expression to match in each log message
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	The time format used in each log message, default is posix time (seconds since 1970)
batchSize	uint32, optional, {unsigned:The value mus be between 2000 and 10000, inclusive}	The number of logs to retrieve per batch, default is 2000

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

pull

Description: Downloads N log entries

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The frs log id number to retrieve, this is mutually exclusive with id2
id2	uint64, optional	The las log id number that will be sent, defaults to mos recent log message when id1 or id2 is not sent
count	uint32, optional, {unsigned:The value mus be between 1 and 10000, inclusive}	The number of logs to pull
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	The time format used in each log message, default is posix time (seconds since 1970)

timeRoll

Description: Delete log entries that exceed a given age

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
timeCalc	sring, optional, {enum-any:The value mus be one or more of the following: current|las-write}	The time calculation to use (current date or las write date), default is current
minutes	uint32, optional, {unsigned:The value mus be between 0 and 60, inclusive}	Remove log entries older than the given number of minutes
hours	uint32, optional, {unsigned:The value mus be between 0 and 24, inclusive}	Remove log entries older than the given number of hours
days	uint32, optional, {unsigned:The value mus be between 0 and 365, inclusive}	Remove log entries older than the given number of days
date	sring, optional	Remove log entries older than the given date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

/res node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated
		values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

/sdk node

Manual for /sdk

API Messages

aliases

Description: Retrieves aliases for the specifed key

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
key	sring	The language key to retrieve aliases for
value	sring, optional	An optional value to return a specifc alias for

cancel

Description: Cancel a running query, regardless of the user

Security.roles: sdk.manage,sdk.meta **Parameters:**

Parameter	Type, Options	Description
handle	uint32	The channel handle running the query
uuid	sring, optional	The device UUID if taking a device ofine insead of canceling the entire query (Broker only). Passing uuid will propagate the reques upsream to fnd the frs matching device to cancel.

content

Description: Returns the packet content for a session

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
session	uint64, optional	The session id to retrieve packet content for
packet	uint64, optional	The packet id of the frs packet of the session, in case the session has rolled out
pinId	sring, optional	The Pin ID of the session to retrieve from long term sorage. If provided, the packet and session parameters mus not be

		provided.
maxSize	size, optional	The max number of bytes to return, zero means no limit
failOnCacheMiss	bool, optional	If true, the content call fails if the NWD is not in the cache. This is used by optimized caching code only.
renderType	sring, optional	The render type (see NwContentTypes in NwSDK.h), pass zero to pick bes option. Not specifying this parameter always results in an NWD and renderFlags and renderOptions are not considered. The following values are also supported: 'pcap', 'fle-lis-json', 'session-meta-fle-lis-json', 'nwd', 'log', 'fles'
renderFlags	uint32, optional	Bitwise mask to control options, (see NwContentFlags in NwSDK.h)
renderOptions	sring, optional	An encoded sring params containing additional options for controlling the operation.

Manual for content

count

Description: Returns the number of child nodes **Security.roles:** everyone

delCache

Description: Removes all cached NWD fles from either the normal NWD cache or the pin cache

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
type	sring, optional, {enum-one:The value mus be one of the following: nwd|pin|all}	Specifes which cache to delete all NWD fles from. The default is 'nwd'. Passing 'all' will delete from all caches.

deviceId

Description: Given a session id, this message retrieves the session id and the device it maps to

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
session	uint64	The session id that will be translated into the device name and session id.

Manual for deviceId

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

hierarch

Description: Report the set of devices attached to this service

Security.roles: sdk.meta Manual for

hierarch **info**

Description: Returns detailed information about the node **Security.roles:** everyone

keyrefs

Description: Retrieves the set of entity keys

Security.roles: sdk.meta Manual for

keyrefs

language

Description: Retrieves the language defnition

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The sarting language id

id2	uint64, optional	The ending language id
time1	date-time, optional	The sarting time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS)
time2	date-time, optional	The ending time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS)
options	sring, optional	Extra options, currently unused
fags	sring, optional	Optional fags to confgure how the results are returned. Can be a number (bitwise mask) or comma separated values like default, no-count, quick-count or full-count.
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
devices	sring, optional	Optional lis of comma separated device names or UUIDs that will be queried. If this parameter is not present, then all devices are queried. The device names mus match the lis defned under /broker/devices or match the /sys/sats/uuid of each device to be queried.
excludeDevices	sring, optional	Optional lis of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs mus match the UUID defned under /sys/sats/uuid of each device to be queried.

Manual for language **ls**

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

msearch

Description: Search for pattern matches in many sessions or packets

Security.roles: sdk.content & sdk.meta **Parameters:**

Parameter	Type, Options	Description
sessions	sring, optional	The session ID ranges to search
packets	sring, optional	The packet ID ranges to search
search	sring	The search sring to use
where	sring, optional	The where clause used to identify sessions to consider for the search
limit	uint64, optional	The maximum number of sessions to traverse for this search
size	uint64, optional	The maximum number of results to return for this search
fags	sring, {enum-any:The value mus be one or more of the following: regex|sp|sm|ci|pre|pos|ds|precache|si}	Flags to use for search. This is a comma separated lis of one or more of the fag values: regex,sp,sm,ci,pre,pos,ds,precache,si
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
devices	sring, optional	Optional lis of comma separated device names or UUIDs that will be queried. If this parameter is not present, then all devices are queried. The device names mus match the lis defned under /broker/devices or match the /sys/sats/uuid of each device to be queried.
excludeDevices	sring, optional	Optional lis of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs mus match the UUID defned under /sys/sats/uuid of each device to be queried.

Manual for msearch

packets

Description: Stream packets back based on the input parameters provided

Security.roles: sdk.content & sdk.packets

Parameters:

Parameter	Type, Options	Description

op	sring, {enum-one:The value mus be one of the following: setup|sart|cancel|processed}	The operation to perform (sart, cancel, processed)
sessions	sring, optional	A comma separated lis of session ids or session id ranges (#-#) whose packets will be sreamed back.
packets	sring, optional	A comma separated lis of packet ids or session/packet ids (#&#) that will be sreamed back.
where	sring, optional	Where clause which will be evaluated to determine which sessions to sream back
time1	date-time, optional	A sarting time range (UTC) where matching packets will be sreamed back ("2010-Apr-20 09:00:00")
time2	date-time, optional	An ending time range (UTC) where matching packets will be sreamed back ("2010-Apr-20 10:00:00")
fags	uint32, optional	Additional fags as defned by the NwPackets SDK function
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
devices	sring, optional	Optional lis of comma separated device names that will be queried. If this parameter is not present, then all devices are queried. The device names mus match the lis defned under /broker/devices
excludeDevices	sring, optional	Optional lis of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs mus match the UUID defned under /sys/sats/uuid of each device to be queried.

Manual for packets

pin

Description: Pin a session in the long term cache so it can be retrieved even after the meta or packets have rolled out. To retrieve, you mus make a content call with the Pin ID that is returned from this command. The op={unpin,validate} commands all take one or more comma separated Pin IDs in the pinId parameter and returns the satus for the Pin IDs that were recognized by a service.

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
session	uint64, optional	The session ID to pin. This cannot be used with the 'sessions' parameter.
sessions	sring, optional	A comma separated lis of session ids or session id ranges (#-#) to pin. This cannot be used with the 'session' parameter.
pinId	sring, optional	One or more Pin IDs, separated by commas
op	sring, optional, {enum-one:The value mus be one of the following: ls|unpin|validate}	Various operations that can be performed. Will be submitted to upsream devices if not confgured for long term cache.

Manual for pin

precache

Description: Efciently caches NWD fles for future retrieval via the content message

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
sessions	sring, optional	A comma separated lis of session ids or session id ranges (#-#) that will be saved in the NWD cache directory
packets	sring, optional	A comma separated lis of session and packet ids (#p#) that will be saved in the NWD cache directory
waitOnComplete	bool, optional	If true, will wait for precache to complete before returning a response.

query

Description: Performs a query agains the meta database

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The sarting meta id (to run the query from mos recent to oldes meta, make id1 larger than id2)
id2	uint64, optional	The ending meta id
size	uint32, optional	The max number of entries to return, or jus sream back all results if zero
query	sring, optional	The query sring to use
fags	sring, optional	The fags to use for query. Can be a number (bitwise mask) or comma separated values like query-log.
threshold	uint64, optional	Query optimization to sop processing results after the threshold is reached (useful with select aggregate functions). Zero means no threshold (the default).
partition	uint32, optional	Used to partition the result set across multiple clients. Mus be

		greater than zero and less than or equal to totalPartitions
totalPartitions	uint32, optional	The total number of clients that will be submitting the same query for partitioning.
search	sring, optional	A text search clause to apply to the where clause of this query
sreamLimit	uint32, optional	Limit the number of sessions returned by a sreaming query
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
devices	sring, optional	Optional lis of comma separated device names or UUIDs that will be queried. If this parameter is not present, then all devices are queried. The device names mus match the lis defned under /broker/devices or match the /sys/sats/uuid of each device to be queried.
excludeDevices	sring, optional	Optional lis of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs mus match the UUID defned under /sys/sats/uuid of each device to be queried.

Manual for query

reconfg

Description: Calculates default values for some of the confg nodes.

Security.roles: sdk.manage **Parameters:**

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the confg nodes with the defaults
memory	size, optional	If non-zero, will use that setting as insalled physical RAM (e.g., memory="24 GiB"). This is useful for hybrids and/or all-in-one machines where the service will need to share memory.

search

Description: Searches for matches in session/packet content

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
session	uint64	The session id to search
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
search	sring	The search sring to use

Manual for search

session

Description: Retrieves the meta id range for the session range

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
id1	uint64	The sarting session id (inclusive). Pass zero to scope to lowes valid id.
id2	uint64	The ending session id (inclusive). Pass zero to scope to highes valid id.

Manual for session

summary

Description: Retrieves summary information from the databases

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
fags	sring, optional	Optional SDK fags. Can be a number (bitwise mask) or comma separated values like default or ignore-cache.

Manual for summary

timeline

Description: Returns the count of sessions/size/packets in discrete time intervals

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
time1	date-time, optional	The sarting time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS), if greater than time2, then the whole time range is returned

time2	date-time, optional	The ending time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS)
timezone	int32, optional	The timezone of the local computer to receive the data, hour ofset from GMT [-13 to 13]
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
fags	sring, optional	The fags to use for the timeline. Can be a number (bitwise mask) or comma separated values like sessions, size, packets, order-ascending, order-descending, ignore-cache or clear-cache.
where	sring, optional	Optional where clause for fltering the data
bookendHours	uint32, optional	The number of hours outside the given time1, time2 which will become part of the cache. Default is 24 hours.
feldName	sring, optional	The time feld to retrieve values for, defaults to 'time'
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
search	sring, optional	A text search clause to apply to the where clause of this query
devices	sring, optional	Optional lis of comma separated device names or UUIDs that will be queried. If this parameter is not present, then all devices are queried. The device names mus match the lis defned under /broker/devices or match the /sys/sats/uuid of each device to be queried.
excludeDevices	sring, optional	Optional lis of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs mus match the UUID defned under /sys/sats/uuid of each device to be queried.

Manual for timeline **validate**

Description: Validate a query or values call without execution

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: query|values}	The operation to validate
validateLanguage	bool, optional	If true (default is false), language tokens and syntax will be

		validated, otherwise jus syntax will be validated
feldName	sring, optional	Required if validating a 'values' operation. The feld to retrieve values for.
where	sring, optional	Required if validating a 'values' operation. The flter criteria to be validated.
query	sring, optional	Required if validating a 'query' operation. The query sring to be validated.
json	bool, optional	Structure response in JSON syntax

Manual for validate

values

Description: Performs a value count query and returns the matching values for a report

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The sarting meta id
id2	uint64, optional	The ending meta id
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
fags	sring, optional	The fags to use for values. Can be a number (bitwise mask) or comma separated values like sessions, size, packets, aggregate, sort-total, sort-value, order-ascending, order-descending, ignore-cache, clear-cache, or database-scan.
threshold	uint64, optional	Query optimization to sop processing large session counts
feldName	sring	The feld to retrieve values for
where	sring, optional	The flter criteria for the values
aggregateFunction	sring, optional, {enum-one:The value mus be one of the following: count|sum}	The aggregate operation to be applied where the aggregate fag is set
aggregateFieldName	sring, optional	The meta feld to aggregate in the aggregateFunction
min	sring, optional	The lower limit of values to return. Only values greater than this value will be returned.
max	sring, optional	The upper limit of values to return. Only values less than this value will be returned.

queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
search	sring, optional	A text search clause to apply to the where clause of this query
devices	sring, optional	Optional lis of comma separated device names or UUIDs that will be queried. If this parameter is not present, then all devices are queried. The device names mus match the lis defned under /broker/devices or match the /sys/sats/uuid of each device to be queried.
excludeDevices	sring, optional	Optional lis of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs mus match the UUID defned under /sys/sats/uuid of each device to be queried.

Manual for values

xforms

Description: Retrieves transforms for the specifed key

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
key	sring, optional	An optional language key to retrieve transform for, all transforms returned if not provided

/sys node

Manual for /sys

API Messages

caCert

Description: Display or delete trused CA certs

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: lis|delete|add}	CA action to perform (lis,delete,add)
id	sring, optional	ID of CA certifcate to remove when op=delete

Manual for caCert

count

Description: Returns the number of child nodes **Security.roles:** everyone

fleEdit

Description: View and edit sysem confguration fles

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: dir|get|getBackup|put|create|delete}	The operation to perform (dir, get, getBackup, put, create, delete)
type	sring, optional, {enum-one:The value mus be one of the following: parser|flter}	The type of fle, if not provided will check fle extension
flename	sring, optional, {length:The value length mus be between 1 and 64, inclusive}	File to create, edit, get or delete

Manual for fileEdit

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

peerCert

Description: Display or modify trused peer certifcates

Security.roles: sys.manage
Parameters:

Parameter	Type, Options	Description
op	sring, optional, {enum-one:The value mus be one of the following: lis|delete|add}	Operation to perform on the lis of trused peers (lis,delete,add)
id	sring, optional	ID of peer certifcate to remove when op=delete

Manual for peerCert

save

Description: Forces a save of sysem settings

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
force	uint32, optional	If 1, forces the confguration fle to saved, regardless if changes are detected, default is 0

Manual for save

servCert

Description: Display the current server certifcate in PEM format.

Security.roles: sys.manage Manual for

servCert

shutdown

Description: Stop the service

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
reason	sring, optional, {length:The value length mus be between 0 and 512, inclusive}	The reason for shutting down the service
cl	sring, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next resart

Manual for shutdown

satHis

Description: Retrieve hisorical sats from the sats db. Don't send time1/time2 to get bounding times about sats db. Supported wildcards are ? to match any single char, * to match zero or more characters, not including slash /, ** to match zero or more characters including slash /. **Security.roles:** sys.manage

Parameters:

Parameter	Type, Options	Description
time1	date-time, optional	The sarting time for retrieving sats
time2	date-time, optional	The ending time for retrieving sats
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	Specify the time format for each sat snapshot, default is posix (seconds since 1970)
include	sring, optional	Comma separated lis of sats to include (wildcards allowed)
exclude	sring, optional	Comma separated lis of sats to exclude (wildcards allowed, has precedence over include)
onChange	sring, optional	Comma separated lis of sats that reduce the result set to only when at leas one of the sats change from their previous value
reduce	bool, optional	If true, reduces data transmission by replacing sat pathnames with a shorthand and provides a lookup table as frs result
showAll	bool, optional	If true, changes each result to show all sats previously seen, even if the value hasn't changed. It has a moderate performance impact and is bes used with include, exclude or onChange parameters to reduce the returned result set size.

Manual for statHist

telemtry

Description: Returns telemetry information about the service

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
level	uint32, optional, {signed:The value mus be between 1 and 3, inclusive}	The level of information to be sent by the telemetry reques. A value of 1 (default) will send all sat nodes, a value of 2 will send sat and confg nodes, and a value of 3 will send sat nodes, confg nodes and usage metrics for rules, parsers and devices (where applicable).
include	sring, optional	Comma separated lis of nodes to include (wildcard usage is ok).
exclude	sring, optional	Comma separated lis of nodes to exclude (wildcard usage is ok). If "include" is also specifed, exclusions are applied to the set of nodes specifed by "include".
url	sring, optional	Endpoint URL to send the JSON response.
op	sring, optional, {enum-one:The value mus be one of the following: get|confgure}	The telemetry operation to execute. Default is 'get'.

/users node

API Messages

addOrMod

Description: Add a new user or update an exising user in the sysem

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#$%&'+-=?^_`{|}~.[]\}	The username to add or update, mus be alphanumeric or @!#$%&'+-=?^_`{|}~.[]\
password	sring, optional	The user's password
pwdIsHashed	bool, optional	Set to true if the password has already been hashed
groups	sring, optional	The groups the user belongs to
authType	sring, optional, {enum-one:The value mus be one of the following: netwitness|pam}	The authentication sysem to use, "netwitness" is the default
queryTimeout	uint32, optional, {unsigned:The value mus be between 0 and 43200, inclusive}	The maximum number of minutes that a query is allowed to execute for this user. Zero means unlimited.
displayName	sring, optional	Display name for this user account
email	sring, optional	Email address

		for this account
description	sring, optional	Description of this user account
queryPrefx	sring, optional	Query filter applied to every query performed by this user account
sessionThreshold	uint32, optional, {unsigned:The value mus be between 0 and 4294967295, inclusive}	Query optimization which will extrapolate the remaining session counts when they exceed this value

auths

Description: Get supported authentication types **Security.roles:** everyone

count

Description: Returns the number of child nodes **Security.roles:** everyone

delete

Description: Delete a user from the sysem

Security.roles: users.manage **Parameters:**

Parameter	Type, Options	Description
name	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#$%&'+-=?^_`{|}~.[]\}	The username to delete

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone
Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

unlock

Description: Unlock a locked out account

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	sring	The username or account to unlock

whoAmI

Description: Returns information about the authenticated user

Security.roles: everyone

Archiver service

Introduction

Archiver is similar to a Concentrator in all but one respect. In addition to aggregating sessions and meta, it would also aggregate the raw logs. It typically has smaller index meant for compliance reports and not invesigations. Because it aggregates logs, it is almos always paired with one or more LogDecoders, not Decoders.
The aggregated data is sored in collections which can be confgured through rules for example: syslog collection for all logs with `device.type=syslog` . By default all the aggregated data would be part of `default` collection.

Requess for the raw data (logs) are not forwarded to LogDecoder since the data is already sored on the Archiver.

API Usage

Archiver service provides API interface through its tree form as nodes, node messages and child nodes.

The following are few examples of the several API nodes and messages supported by Archiver:

- /archiver node messages:
  - `start` and `stop` aggregation `repair` to reset the global summary based on the
  - collection ranges `add` a new device for aggregation
  - `/archiver/stats` node for archiver aggregation stats information /sdk node
- messages:
  - `reconfig` sdk for optimal settings
  - `summary` to retrieve summary information from the database ranges `values` to perform a value
  - count query and return the matching values for a report `query` to perform query against meta
  - database `language` of meta keys for key info, format, description etc...
  - `/sdk/stats` node for queries active, pending and other sdk operations /index node
- messages:
  - `ranges` to show sessionid ranges for all collections
  - `gaps` to show the gaps and session virtual range mappings per collection `save` to save
  - arciver collection mappings to disk /sys node messages:
  - `statHist` to retrieve historical stats from stats database `/sys/stats` node for
  - service and system stats information

Archiver also support per collection API node paths and messages. In order to access API node and messags for a specifc collection the reques path needs to be `/archiver/collections/{collection-name}` e.g for default collection it would be `/archiver/collections/default` .

`Note` : Queries issued on root `/sdk` path would be run agains all available collections and queries issued on collection sdk path would be run agains that collection.

Sample Reques

The following is a sample reques query on Archiver to get 4 values of disinct ip.ds, grouped by ip.src and ordered by disinct ip.ds in descending order.

Reques path is `/sdk` , message is `msg=query` and parameters are `query=\"select distinct(ip.dst) where time=\'2024-08-16 16:50:00\' - \'2024-08-16 16:54:59\' GROUP BY ip.src ORDER BY distinct(ip.dst) desc \" size=4"`

For queries that are expected to take longer time on execution an additional URL parameter `expiry=0` can be added to disable default REST reques timeout and continue with user profle timeout.

Using URL encoded reques

```
$ curl 'https://archiver-hos:50108/sdk?
msg=query&expiry=0&query=select%20disinct(ip.ds)%20where%20time='2024-08-16%2016:50:00'%20-
%20'2024-08-
16%2016:54:59'%20GROUP%20BY%20ip.src%20ORDER%20BY%20disinct(ip.ds)%20desc&size=4'\
    -H 'Accept: application/json;charset=UTF-8' \
    -H 'Content-Type: application/x-www-form-urlencoded; charset=ISO-8859-1' \
    -u 'username:password'
```

Using special content type

```
Content Type: application/x-netwitness-sring-params

$ curl 'https://archiver-hos:50108/sdk?msg=query&expiry=0' \
    -H 'Accept: application/json;charset=UTF-8' \
    -H 'Content-Type: application/x-netwitness-sring-params' \
    -d "query=\"select disinct(ip.ds) where time=\'2024-08-16 16:50:00\' - \'2024-08-16
16:54:59\' GROUP BY ip.src ORDER BY disinct(ip.ds) desc  \" size=4" \
    -u 'username:password'
```

HTTP Headers

```
HTTP/1.1 200 OK
Content-Length: 561
Connection: Keep-Alive
Pragma: no-cache
Expires: -1
Cache-Control: no-cache, no-sore, mus-revalidate
Content-Type: application/json
```

JSON Response

```
{
  "fags": 1074200577,
```

```
  "results": {
    "id1": 1231176344,
    "id2": 1267146902,
    "felds": [
      {
        "id1": 1126481600,
        "id2": 1212151333,
        "count": 142,
        "format": 128,
        "type": "ip.ds",
        "fags": 6,
        "group": 0,
        "value": "192.XX.XX.246"
      },
      {
        "id1": 1126676313,
        "id2": 1231176343,
        "count": 342,
        "format": 128,
        "type": "ip.ds",
        "fags": 6,
        "group": 0,
        "value": "10.XX.XX.22"
      },
      {
        "id1": 1126779641,
        "id2": 1225198229,
        "count": 320,
        "format": 128,
        "type": "ip.ds",
        "fags": 6,
        "group": 0,
        "value": "10.XX.XX.174"
      },
      {
        "id1": 1126936749,
        "id2": 1224923290,
        "count": 660,
        "format": 128,
        "type": "ip.ds",
        "fags": 6,
        "group": 0,
        "value": "10.XX.XX.235"
      }
    ]
  }
}
```


`Note` : The API Messages section for the path nodes provide more details on API messages available and any associated manual pages.

/archiver node

API Messages

add

Description: Add a new device for aggregation

Security.roles: archiver.manage **Parameters:**

Parameter	Type, Options	Description
device	sring, optional, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to add (ip:port or [ipv6]:port), either this or ip mus be set
ip	sring, optional	The IP address of the device, either this or device mus be set
port	uint32, optional, {unsigned:The value mus be between 0 and 65535, inclusive}	The port number the device is lisening on
username	sring	The username to run as
password	sring, optional	The account password. Assumes service trus aggregation if no password is provided.
options	sring, optional	The device options
ssl	sring, optional	If true, the device requires SSL to connect
backup	sring, optional	If true, the device will be used as a backup
reconnect	sring, optional	If true, will automatically reconnect when not connected
name	sring, optional	Device collection name if device is a collection

count

Description: Returns the number of child nodes **Security.roles:** everyone

delete

Description: Delete an exising aggregation device

Security.roles: archiver.manage

Parameters:

Parameter	Type, Options	Description
device	sring, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to delete
name	sring, optional	Device collection name if device is a collection

edit

Description: Edit an exising aggregation device

Security.roles: archiver.manage **Parameters:**

Parameter	Type, Options	Description
device	sring, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to edit
username	sring, optional	The username to run as
password	sring, optional	The account password. For service trus aggregation you mus not provide this parameter
options	sring, optional	The device options
ssl	sring, optional	If true, the device requires SSL to connect
backup	sring, optional	If true, the device will be used as a backup
reconnect	sring, optional	If true, will automatically reconnect when not connected
name	sring, optional	Device collection name if device is a collection
switchToTrusedAuth	bool, optional	If true, removes any password associated with this account and will use trused authentication henceforth. When switching to trused, only the 'device' parameter is allowed.

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

reconfg

Description: Calculates optimal settings for archiver pools and bufers based on the insalled hardware.

Security.roles: archiver.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the respective archiver settings, otherwise it will jus output the calculations
memory	size, optional	If non-zero, will use that setting as insalled physical RAM (e.g., memory="24 GiB"). This is useful for hybrids and/or all-in-one machines where the service will need to share memory.

repair

Description: Reset the global summary based on the collection ranges -or- delete a specifed collection's mapping ranges and regenerate them anew.

Security.roles: archiver.manage **Parameters:**

Parameter	Type, Options	Description
name	sring, optional	The name of the collection if regenerating ranges

reset

Description: Reset data, index, manifess, sats, confguration, or logs for this service. Data automatically deletes index and sats, unless flesCreatedAfter is specifed. Service is automatically resarted. Example arguments:d ata=1 confg=1 log=1T his example will reset data, index, logs, and confguration index=1T his example will reset the index only manifes=1T his example will delete everything in the manifes directory flesCreatedAfter="2015-12-01 14:00:00"T his example will delete all session, meta and packet fles created on or after Dec 1s, 2015 2pm (UTC) from the service. All other fles will remain. The index will not be touched, but upon resart will be truncated to match the las session in the session database. **Security.roles:** archiver.manage

Parameters:

Parameter	Type, Options	Description
data	bool, optional	Reset data, automatically resets index (may not be applicable to all services)
index	bool, optional	Reset index, only valid if data is not reset (may not be applicable to all services)
manifes	bool, optional	Reset all manifess sored in the long term manifes directory (may not be applicable to all services)
confg	bool, optional	Reset confguration settings to default
sats	bool, optional	Reset the sats database
log	bool, optional	Reset the log database
flesCreatedAfter	date-time, optional	Delete all database fles (session, meta, packets) created after a certain date. The index will automatically get rolled back on resart. Not valid with option index.
cl	sring, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next resart
parseStats	bool, optional	Reset the parse sats database (PD/LD only)
force	bool, optional	Force reset without confrmation

resetMax

Description: Resets all max sats, including device max sats back to zero. **Security.roles:**

archiver.manage

sart

Description: Starts aggregation

Security.roles: archiver.manage

satus

Description: Change the online/ofine satus of an aggregation device

Security.roles: archiver.manage **Parameters:**

Parameter	Type, Options	Description
device	sring, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to change satus
satus	sring, optional, {enum-one:The value mus be one of the following: online|ofine}	Whether to take device ofine or put online
name	sring, optional	Device collection name if device is a collection

sop

Description: Stops aggregation

Security.roles: archiver.manage **Parameters:**

Parameter	Type, Options	Description
fush	bool, optional	If true (default), process all sessions and packets currently in memory. Otherwise, sop immediately, discarding unprocessed sessions and packets.

whoAgg

Description: Returns information on who is aggregating from this service

Security.roles: archiver.manage

Parameters:

Parameter	Type, Options	Description
clean	bool, optional	If true, will delete any aggregation trackers that no longer have a valid channel

/connections node

API Messages

closeAll

Description: Closes all connections

Security.roles: connections.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, optional, {enum-any:The value mus be one or more of the following: native|res|amqp}	Indicates the type of connection to close, either native, res or amqp. Default is all types.
limit	uint32, optional	If non-zero, will sop closing connections after limit is reached. NOTE: It sarts closing the mos idle connections frs.
idleLongerThan	sring, optional	A time duration in HH:MM:SS or ##{d,h,m,s} format. Any connection that has been idle longer than the passed in duration is eligible to be closed. Connections that have seen activity within the duration window will not be closed. Examples: "01:05:10" (1 hour, 5 mins, 10 secs) or "65m" (65 minutes)
exceptMe	bool, optional	If true, will not close the connection that initiated the command.

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

netSpeed

Description: Network Speed Tes **Security.roles:**

connections.manage,aggregate

Parameters:

Parameter	Type, Options	Description
size	size, optional, {byte-range:The value (in bytes, size modifers like MB are accepted) mus be between 1024 and 536870912, inclusive}	The size of each message sent, default is 64k
count	uint32, optional, {unsigned:The value mus be between 1 and 536870912, inclusive}	The number of messages to send, default is 16k

/index node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

gaps

Description: Shows the gaps and other error checking in session virtual range mappings per device

Security.roles: index.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The sarting session id
id2	uint64, optional	The ending session id

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node

Security.roles: everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

ranges

Description: Shows the session id ranges for all devices or a single device

Security.roles: index.manage **Parameters:**

Parameter	Type, Options	Description
device	sring, optional	The device to report ranges for, defaults to 'all' devices
name	sring, optional	The device collection name if the device is a collection
id1	uint64, optional	The sarting session id
id2	uint64, optional	The ending session id
format	sring, optional, {enum-one:The value mus be one of the following: tab|comma}	The format of the returned report

save

Description: Saves the broker device mappings to disk

Security.roles: index.manage

/logs node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

download

Description: Downloads log entries

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The sarting log number to download
id2	uint64, optional	The ending log number to download
time1	date-time, optional	The sarting log date to download, can be posix time (seconds since 1970) or DD-MM-YY HH:MM:SS
time2	date-time, optional	The ending log date to download, can be posix time (seconds since 1970) or DD-MM-YY HH:MM:SS
op	sring, {enum-one:The value mus be one of the following: sart|next|cancel}	The operation to perform (sart, next, cancel)
logTypes	sring, optional	The type(s) of log messages to retrieve (debug,info,audit,warning,failure)
match	sring, optional	Case insensitive sring to match in each log message
regex	sring, optional	Regular expression to match in each log message
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	The time format used in each log message, default is posix time (seconds since 1970)
batchSize	uint32, optional, {unsigned:The value mus be between 2000 and 10000, inclusive}	The number of logs to retrieve per batch, default is 2000

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

pull

Description: Downloads N log entries

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The frs log id number to retrieve, this is mutually exclusive with id2
id2	uint64, optional	The las log id number that will be sent, defaults to mos recent log message when id1 or id2 is not sent
count	uint32, optional, {unsigned:The value mus be between 1 and 10000, inclusive}	The number of logs to pull
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	The time format used in each log message, default is posix time (seconds since 1970)

timeRoll

Description: Delete log entries that exceed a given age

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
timeCalc	sring, optional, {enum-any:The value mus be one or more of the following: current|las-write}	The time calculation to use (current date or las write date), default is current
minutes	uint32, optional, {unsigned:The value mus be between 0 and 60, inclusive}	Remove log entries older than the given number of minutes
hours	uint32, optional, {unsigned:The value mus be between 0 and 24, inclusive}	Remove log entries older than the given number of hours
days	uint32, optional, {unsigned:The value mus be between 0 and 365, inclusive}	Remove log entries older than the given number of days
date	sring, optional	Remove log entries older than the given date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

/res node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated
		values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

/sdk node

Manual for /sdk

API Messages

aliases

Description: Retrieves aliases for the specifed key

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
key	sring	The language key to retrieve aliases for
value	sring, optional	An optional value to return a specifc alias for

cancel

Description: Cancel a running query, regardless of the user

Security.roles: sdk.manage,sdk.meta **Parameters:**

Parameter	Type, Options	Description
handle	uint32	The channel handle running the query
uuid	sring, optional	The device UUID if taking a device ofine insead of canceling the entire query (Broker only). Passing uuid will propagate the reques upsream to fnd the frs matching device to cancel.

content

Description: Returns the packet content for a session

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
session	uint64, optional	The session id to retrieve packet content for
packet	uint64, optional	The packet id of the frs packet of the session, in case the session has rolled out
pinId	sring, optional	The Pin ID of the session to retrieve from long term sorage. If provided, the packet and session parameters mus not be

		provided.
maxSize	size, optional	The max number of bytes to return, zero means no limit
failOnCacheMiss	bool, optional	If true, the content call fails if the NWD is not in the cache. This is used by optimized caching code only.
renderType	sring, optional	The render type (see NwContentTypes in NwSDK.h), pass zero to pick bes option. Not specifying this parameter always results in an NWD and renderFlags and renderOptions are not considered. The following values are also supported: 'pcap', 'fle-lis-json', 'session-meta-fle-lis-json', 'nwd', 'log', 'fles'
renderFlags	uint32, optional	Bitwise mask to control options, (see NwContentFlags in NwSDK.h)
renderOptions	sring, optional	An encoded sring params containing additional options for controlling the operation.

Manual for content

count

Description: Returns the number of child nodes **Security.roles:** everyone

delCache

Description: Removes all cached NWD fles from either the normal NWD cache or the pin cache

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
type	sring, optional, {enum-one:The value mus be one of the following: nwd|pin|all}	Specifes which cache to delete all NWD fles from. The default is 'nwd'. Passing 'all' will delete from all caches.

deviceId

Description: Given a session id, this message retrieves the session id and the device it maps to

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
session	uint64	The session id that will be translated into the device name and session id.

Manual for deviceId

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

hierarch

Description: Report the set of devices attached to this service

Security.roles: sdk.meta Manual for

hierarch **info**

Description: Returns detailed information about the node **Security.roles:** everyone

keyrefs

Description: Retrieves the set of entity keys

Security.roles: sdk.meta Manual for

keyrefs

language

Description: Retrieves the language defnition

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The sarting language id

id2	uint64, optional	The ending language id
time1	date-time, optional	The sarting time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS)
time2	date-time, optional	The ending time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS)
options	sring, optional	Extra options, currently unused
fags	sring, optional	Optional fags to confgure how the results are returned. Can be a number (bitwise mask) or comma separated values like default, no-count, quick-count or full-count.
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
devices	sring, optional	Optional lis of comma separated device names or UUIDs that will be queried. If this parameter is not present, then all devices are queried. The device names mus match the lis defned under /archiver/devices or match the /sys/sats/uuid of each device to be queried.
excludeDevices	sring, optional	Optional lis of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs mus match the UUID defned under /sys/sats/uuid of each device to be queried.

Manual for language **ls**

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

msearch

Description: Search for pattern matches in many sessions or packets

Security.roles: sdk.content & sdk.meta **Parameters:**

Parameter	Type, Options	Description
sessions	sring, optional	The session ID ranges to search
packets	sring, optional	The packet ID ranges to search
search	sring	The search sring to use
where	sring, optional	The where clause used to identify sessions to consider for the search
limit	uint64, optional	The maximum number of sessions to traverse for this search
size	uint64, optional	The maximum number of results to return for this search
fags	sring, {enum-any:The value mus be one or more of the following: regex|sp|sm|ci|pre|pos|ds|precache|si}	Flags to use for search. This is a comma separated lis of one or more of the fag values: regex,sp,sm,ci,pre,pos,ds,precache,si
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
devices	sring, optional	Optional lis of comma separated device names or UUIDs that will be queried. If this parameter is not present, then all devices are queried. The device names mus match the lis defned under /archiver/devices or match the /sys/sats/uuid of each device to be queried.
excludeDevices	sring, optional	Optional lis of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs mus match the UUID defned under /sys/sats/uuid of each device to be queried.

Manual for msearch

packets

Description: Stream packets back based on the input parameters provided

Security.roles: sdk.content & sdk.packets

Parameters:

Parameter	Type, Options	Description

op	sring, {enum-one:The value mus be one of the following: setup|sart|cancel|processed}	The operation to perform (sart, cancel, processed)
sessions	sring, optional	A comma separated lis of session ids or session id ranges (#-#) whose packets will be sreamed back.
packets	sring, optional	A comma separated lis of packet ids or session/packet ids (#&#) that will be sreamed back.
where	sring, optional	Where clause which will be evaluated to determine which sessions to sream back
time1	date-time, optional	A sarting time range (UTC) where matching packets will be sreamed back ("2010-Apr-20 09:00:00")
time2	date-time, optional	An ending time range (UTC) where matching packets will be sreamed back ("2010-Apr-20 10:00:00")
fags	uint32, optional	Additional fags as defned by the NwPackets SDK function
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
devices	sring, optional	Optional lis of comma separated device names that will be queried. If this parameter is not present, then all devices are queried. The device names mus match the lis defned under /archiver/devices
excludeDevices	sring, optional	Optional lis of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs mus match the UUID defned under /sys/sats/uuid of each device to be queried.

Manual for packets

pin

Description: Pin a session in the long term cache so it can be retrieved even after the meta or packets have rolled out. To retrieve, you mus make a content call with the Pin ID that is returned from this command. The op={unpin,validate} commands all take one or more comma separated Pin IDs in the pinId parameter and returns the satus for the Pin IDs that were recognized by a service.

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
session	uint64, optional	The session ID to pin. This cannot be used with the 'sessions' parameter.
sessions	sring, optional	A comma separated lis of session ids or session id ranges (#-#) to pin. This cannot be used with the 'session' parameter.
pinId	sring, optional	One or more Pin IDs, separated by commas
op	sring, optional, {enum-one:The value mus be one of the following: ls|unpin|validate}	Various operations that can be performed. Will be submitted to upsream devices if not confgured for long term cache.

Manual for pin

precache

Description: Efciently caches NWD fles for future retrieval via the content message

Security.roles: sdk.content **Parameters:**

Parameter	Type, Options	Description
sessions	sring, optional	A comma separated lis of session ids or session id ranges (#-#) that will be saved in the NWD cache directory
packets	sring, optional	A comma separated lis of session and packet ids (#p#) that will be saved in the NWD cache directory
waitOnComplete	bool, optional	If true, will wait for precache to complete before returning a response.

query

Description: Performs a query agains the meta database

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The sarting meta id (to run the query from mos recent to oldes meta, make id1 larger than id2)
id2	uint64, optional	The ending meta id
size	uint32, optional	The max number of entries to return, or jus sream back all results if zero
query	sring, optional	The query sring to use
fags	sring, optional	The fags to use for query. Can be a number (bitwise mask) or comma separated values like query-log.
threshold	uint64, optional	Query optimization to sop processing results after the threshold is reached (useful with select aggregate functions). Zero means no threshold (the default).
partition	uint32, optional	Used to partition the result set across multiple clients. Mus be

		greater than zero and less than or equal to totalPartitions
totalPartitions	uint32, optional	The total number of clients that will be submitting the same query for partitioning.
search	sring, optional	A text search clause to apply to the where clause of this query
sreamLimit	uint32, optional	Limit the number of sessions returned by a sreaming query
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
devices	sring, optional	Optional lis of comma separated device names or UUIDs that will be queried. If this parameter is not present, then all devices are queried. The device names mus match the lis defned under /archiver/devices or match the /sys/sats/uuid of each device to be queried.
excludeDevices	sring, optional	Optional lis of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs mus match the UUID defned under /sys/sats/uuid of each device to be queried.

Manual for query

reconfg

Description: Calculates default values for some of the confg nodes.

Security.roles: sdk.manage **Parameters:**

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the confg nodes with the defaults
memory	size, optional	If non-zero, will use that setting as insalled physical RAM (e.g., memory="24 GiB"). This is useful for hybrids and/or all-in-one machines where the service will need to share memory.

search

Description: Searches for matches in session/packet content

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
session	uint64	The session id to search
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
search	sring	The search sring to use

Manual for search

session

Description: Retrieves the meta id range for the session range

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
id1	uint64	The sarting session id (inclusive). Pass zero to scope to lowes valid id.
id2	uint64	The ending session id (inclusive). Pass zero to scope to highes valid id.

Manual for session

summary

Description: Retrieves summary information from the databases

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
fags	sring, optional	Optional SDK fags. Can be a number (bitwise mask) or comma separated values like default or ignore-cache.

Manual for summary

timeline

Description: Returns the count of sessions/size/packets in discrete time intervals

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
time1	date-time, optional	The sarting time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS), if greater than time2, then the whole time range is returned

time2	date-time, optional	The ending time in seconds since 1970 or a time sring (YYYY-MMM-DD HH:MM:SS)
timezone	int32, optional	The timezone of the local computer to receive the data, hour ofset from GMT [-13 to 13]
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
fags	sring, optional	The fags to use for the timeline. Can be a number (bitwise mask) or comma separated values like sessions, size, packets, order-ascending, order-descending, ignore-cache or clear-cache.
where	sring, optional	Optional where clause for fltering the data
bookendHours	uint32, optional	The number of hours outside the given time1, time2 which will become part of the cache. Default is 24 hours.
feldName	sring, optional	The time feld to retrieve values for, defaults to 'time'
queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
search	sring, optional	A text search clause to apply to the where clause of this query
devices	sring, optional	Optional lis of comma separated device names or UUIDs that will be queried. If this parameter is not present, then all devices are queried. The device names mus match the lis defned under /archiver/devices or match the /sys/sats/uuid of each device to be queried.
excludeDevices	sring, optional	Optional lis of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs mus match the UUID defned under /sys/sats/uuid of each device to be queried.

Manual for timeline **validate**

Description: Validate a query or values call without execution

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: query|values}	The operation to validate
validateLanguage	bool, optional	If true (default is false), language tokens and syntax will be

		validated, otherwise jus syntax will be validated
feldName	sring, optional	Required if validating a 'values' operation. The feld to retrieve values for.
where	sring, optional	Required if validating a 'values' operation. The flter criteria to be validated.
query	sring, optional	Required if validating a 'query' operation. The query sring to be validated.
json	bool, optional	Structure response in JSON syntax

Manual for validate

values

Description: Performs a value count query and returns the matching values for a report

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The sarting meta id
id2	uint64, optional	The ending meta id
size	uint32, {unsigned:The value mus be between 1 and 1677721, inclusive}	The max number of entries to return
fags	sring, optional	The fags to use for values. Can be a number (bitwise mask) or comma separated values like sessions, size, packets, aggregate, sort-total, sort-value, order-ascending, order-descending, ignore-cache, clear-cache, or database-scan.
threshold	uint64, optional	Query optimization to sop processing large session counts
feldName	sring	The feld to retrieve values for
where	sring, optional	The flter criteria for the values
aggregateFunction	sring, optional, {enum-one:The value mus be one of the following: count|sum}	The aggregate operation to be applied where the aggregate fag is set
aggregateFieldName	sring, optional	The meta feld to aggregate in the aggregateFunction
min	sring, optional	The lower limit of values to return. Only values greater than this value will be returned.
max	sring, optional	The upper limit of values to return. Only values less than this value will be returned.

queryPriority	int32, optional, {signed:The value mus be zero OR between -20 and 20, inclusive}	The query priority for this query
search	sring, optional	A text search clause to apply to the where clause of this query
devices	sring, optional	Optional lis of comma separated device names or UUIDs that will be queried. If this parameter is not present, then all devices are queried. The device names mus match the lis defned under /archiver/devices or match the /sys/sats/uuid of each device to be queried.
excludeDevices	sring, optional	Optional lis of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs mus match the UUID defned under /sys/sats/uuid of each device to be queried.

Manual for values

xforms

Description: Retrieves transforms for the specifed key

Security.roles: sdk.meta **Parameters:**

Parameter	Type, Options	Description
key	sring, optional	An optional language key to retrieve transform for, all transforms returned if not provided

/sys node

Manual for /sys

API Messages

caCert

Description: Display or delete trused CA certs

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: lis|delete|add}	CA action to perform (lis,delete,add)
id	sring, optional	ID of CA certifcate to remove when op=delete

Manual for caCert

count

Description: Returns the number of child nodes **Security.roles:** everyone

fleEdit

Description: View and edit sysem confguration fles

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: dir|get|getBackup|put|create|delete}	The operation to perform (dir, get, getBackup, put, create, delete)
type	sring, optional, {enum-one:The value mus be one of the following: parser|flter}	The type of fle, if not provided will check fle extension
flename	sring, optional, {length:The value length mus be between 1 and 64, inclusive}	File to create, edit, get or delete

Manual for fileEdit

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

peerCert

Description: Display or modify trused peer certifcates

Security.roles: sys.manage
Parameters:

Parameter	Type, Options	Description
op	sring, optional, {enum-one:The value mus be one of the following: lis|delete|add}	Operation to perform on the lis of trused peers (lis,delete,add)
id	sring, optional	ID of peer certifcate to remove when op=delete

Manual for peerCert

save

Description: Forces a save of sysem settings

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
force	uint32, optional	If 1, forces the confguration fle to saved, regardless if changes are detected, default is 0

Manual for save

servCert

Description: Display the current server certifcate in PEM format.

Security.roles: sys.manage Manual for

servCert

shutdown

Description: Stop the service

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
reason	sring, optional, {length:The value length mus be between 0 and 512, inclusive}	The reason for shutting down the service
cl	sring, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next resart

Manual for shutdown

satHis

Description: Retrieve hisorical sats from the sats db. Don't send time1/time2 to get bounding times about sats db. Supported wildcards are ? to match any single char, * to match zero or more characters, not including slash /, ** to match zero or more characters including slash /. **Security.roles:** sys.manage

Parameters:

Parameter	Type, Options	Description
time1	date-time, optional	The sarting time for retrieving sats
time2	date-time, optional	The ending time for retrieving sats
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	Specify the time format for each sat snapshot, default is posix (seconds since 1970)
include	sring, optional	Comma separated lis of sats to include (wildcards allowed)
exclude	sring, optional	Comma separated lis of sats to exclude (wildcards allowed, has precedence over include)
onChange	sring, optional	Comma separated lis of sats that reduce the result set to only when at leas one of the sats change from their previous value
reduce	bool, optional	If true, reduces data transmission by replacing sat pathnames with a shorthand and provides a lookup table as frs result
showAll	bool, optional	If true, changes each result to show all sats previously seen, even if the value hasn't changed. It has a moderate performance impact and is bes used with include, exclude or onChange parameters to reduce the returned result set size.

Manual for statHist

telemtry

Description: Returns telemetry information about the service

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
level	uint32, optional, {signed:The value mus be between 1 and 3, inclusive}	The level of information to be sent by the telemetry reques. A value of 1 (default) will send all sat nodes, a value of 2 will send sat and confg nodes, and a value of 3 will send sat nodes, confg nodes and usage metrics for rules, parsers and devices (where applicable).
include	sring, optional	Comma separated lis of nodes to include (wildcard usage is ok).
exclude	sring, optional	Comma separated lis of nodes to exclude (wildcard usage is ok). If "include" is also specifed, exclusions are applied to the set of nodes specifed by "include".
url	sring, optional	Endpoint URL to send the JSON response.
op	sring, optional, {enum-one:The value mus be one of the following: get|confgure}	The telemetry operation to execute. Default is 'get'.

/users node

API Messages

addOrMod

Description: Add a new user or update an exising user in the sysem

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#$%&'+-=?^_`{|}~.[]\}	The username to add or update, mus be alphanumeric or @!#$%&'+-=?^_`{|}~.[]\
password	sring, optional	The user's password
pwdIsHashed	bool, optional	Set to true if the password has already been hashed
groups	sring, optional	The groups the user belongs to
authType	sring, optional, {enum-one:The value mus be one of the following: netwitness|pam}	The authentication sysem to use, "netwitness" is the default
queryTimeout	uint32, optional, {unsigned:The value mus be between 0 and 43200, inclusive}	The maximum number of minutes that a query is allowed to execute for this user. Zero means unlimited.
displayName	sring, optional	Display name for this user account
email	sring, optional	Email address

		for this account
description	sring, optional	Description of this user account
queryPrefx	sring, optional	Query filter applied to every query performed by this user account
sessionThreshold	uint32, optional, {unsigned:The value mus be between 0 and 4294967295, inclusive}	Query optimization which will extrapolate the remaining session counts when they exceed this value

auths

Description: Get supported authentication types **Security.roles:** everyone

count

Description: Returns the number of child nodes **Security.roles:** everyone

delete

Description: Delete a user from the sysem

Security.roles: users.manage **Parameters:**

Parameter	Type, Options	Description
name	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#$%&'+-=?^_`{|}~.[]\}	The username to delete

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone
Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

unlock

Description: Unlock a locked out account

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	sring	The username or account to unlock

whoAmI

Description: Returns information about the authenticated user

Security.roles: everyone

```
$ curl 'https://appliance-hos:50106/appliance?msg=srvLis'  \
    -H 'Accept: application/json;charset=UTF-8' \
    -H 'Content-Type: application/x-www-form-urlencoded; charset=ISO-8859-1' \
    -u 'username:password'
```

Appliance service

Introduction

Appliance is a service that co-exis along with other core services and provide options to collect appliance sats like disk, cpu, temperature etc..., also manage core services and confgure sorage attached to the machine.

API Usage

Appliance service provides API interface through its tree form as nodes, node messages and child nodes.

The following are few examples of the several API nodes and messages supported by Appliance:

- /appliance node messages:
  - `start` and `stop` core service `srvList` to list core services on the appliance `devlist` list storage devices `dfAlloc` create a default configuration for storage used by a core service `partNew` to make partition on block device `/appliance/stats` node for appliance stats information /sys node messages:
  - `statHist` to retrieve historical stats from stats database
- `/sys/stats` node for service system stats information

Sample Reques

The following is a sample reques to get lis of core services on the appliance.

Reques path is `/appliance` and message is `msg=srvList`

```
$ curl 'https://appliance-hos:50106/appliance?msg=srvLis'  \
    -H 'Accept: application/json;charset=UTF-8' \
    -H 'Content-Type: application/x-www-form-urlencoded; charset=ISO-8859-1' \
    -u 'username:password'
```

Sample Response HTTP Headers

```
HTTP/1.1 200 OK
Content-Length: 657
Connection: Keep-Alive
Pragma: no-cache
Expires: -1
Cache-Control: no-cache, no-sore, mus-revalidate
Content-Type: application/json
```

JSON Response

It is possible for response values to contain text/plain key-value pairs in content for certain requess.

```
{
  "fags": 1073938433,
  "params": {
    "localhos:56004": "type=decoder
    packet.dir=\"/var/netwitness/decoder/packetdb=137.67 GB\"
    meta.dir=\"/var/netwitness/decoder/metadb=137.67 GB\"
    session.dir=\"/var/netwitness/decoder/sessiondb=137.67 GB\"
    index.dir=\"/var/netwitness/decoder/index=130.43 GB\"
    ... "
  }
}
```

`Note`: The API Messages section for the path nodes provide more details on API messages available and any associated manual pages.

/appliance node

Manual for /appliance

API Messages

addFSMon

Description: Add a flesysem monitor Example arguments:p ath=/var/lib/netwitness/decoder/packetdb

Security.roles: appliance.manage **Parameters:**

Parameter	Type, Options	Description
path	sring, {fle:The fle or directory mus exis or be creatable on the sysem}	Filesysem path to monitor

clock

Description: Set the appliance local clock

Security.roles: appliance.manage **Parameters:**

Parameter	Type, Options	Description
set	date-time, optional	Set a new time. Specify the time with format "YYYY-MM-DD HH:MM:SS".

cntrlcnt

Description: Return the number of controllers **Security.roles:**

appliance.manage

count

Description: Returns the number of child nodes **Security.roles:** everyone

delFSMon

Description: Delete a flesysem monitor Example arguments:p ath=/var/lib/netwitness/decoder/packetdb

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
path	sring, {fle-sysem-monitor:Validates the fle sysem exiss}	Monitored flesysem path to remove

devlis

Description: lis sorage devices

Security.roles: appliance.manage

dpToNic

Description: Remove an Ethernet device from DPDK

Security.roles: appliance.manage **Parameters:**

Parameter	Type, Options	Description
eth	sring, optional, {enum-one:The value mus be one of the following: }	Network interface to remove from DPDK

ethtool

Description: Show interface information **Security.roles:**

appliance.manage

fle

Description: Lis, retrieve and delete fles from approved directories on this device

Security.roles: appliance.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: ls|get|delete}	The operation to perform (ls, get, delete)
location	sring, {enum-one:The value mus be one of the following: crashreporter|confg|update|feeds|parsers}	The location on the device where the operation will be performed
flename	sring, optional	The name of the fle to retrieve or delete

fwdLogs

Description: Enable or disable syslog forwarding Example arguments:h os=loghos.localdomain **Security.roles:** appliance.manage **Parameters:**

Parameter	Type, Options	Description
hos	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789=:/.-}	Remote syslog server

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel,
		resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

nicToDp

Description: Attach a NIC to DPDK

Security.roles: appliance.manage **Parameters:**

Parameter	Type, Options	Description
nic	sring, optional, {enum-one:The value mus be one of the following: lo|eth0}	Network interface to send to DPDK

partClr

Description: remove partitions on a block device

Security.roles: appliance.manage **Parameters:**

Parameter	Type, Options	Description
name	sring, {enum-one:The value mus be one of the following: sdb|sdc|sda}	block device name
commit	bool, optional	commit changes

Manual for partClr

partNew

Description: make partitions on a block device

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
name	sring, {enum-one:The value mus be one of the following: }	block device name
service	sring, {enum-one:The value mus be one of the following: archiver|concentrator|decoder|logdecoder}	service that will use sorage
volume	sring, optional, {enum-one:The value mus be one of the following: concentrator|index|decodersmall|decoder|packet|hybrid-decoder-meta|logdecodersmall|logdecoder|logpacket|hybrid-logdecoder-meta|archiver|hybrid-concentrator|endpoint-log-hybrid|log-indexed-decoder|logindex}	volume to create
commit	bool, optional	commit changes

Manual for partNew

powerOf

Description: Shut down this appliance No arguments required **Security.roles:**

appliance.manage

raidLis

Description: lis drive shelves attached to this appliance **Security.roles:**

appliance.manage

reboot

Description: Reboot this appliance No arguments required **Security.roles:**

appliance.manage

serial

Description: Retrieve the appliance serial number **Security.roles:** everyone

setNTP

Description: Set the clock source for this appliance Example arguments:s ource=tictoc.localdomain

Security.roles: appliance.manage **Parameters:**

Parameter	Type, Options	Description
source	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789=:/.-}	The hosname or address of an NTP server that will serve as the hosname for this appliance. Specify "local" to use the built-in clock as the time source.

setSNMP

Description: Enabled or Disable the SNMP service Example arguments:e nable=1

Security.roles: appliance.manage **Parameters:**

Parameter	Type, Options	Description
enable	uint32, {unsigned:The value mus be between 0 and 1, inclusive}	Enable the SNMP service. Set to 1 to enable the service, and 0 to disable the service

srvLis

Description: lis of core services on this appliance **Security.roles:**

appliance.manage

sart

Description: Start a Netwitness service on this appliance Example arguments:s ervice=decoder

Security.roles: appliance.manage **Parameters:**

Parameter	Type, Options	Description
service	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789=:/.-}	The name of the service to sart (decoder, concentrator, etc.)

sop

Description: Stop a Netwitness service on this appliance Example arguments:s ervice=decoder

Security.roles: appliance.manage **Parameters:**

Parameter	Type, Options	Description
service	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789=:/.-}	The name of the service to sop (decoder, concentrator, etc.)

upgrade

Description: Upgrade the server process

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: sart|check|cancel|fnished}	The operation to perform (sart, cancel, fnished)
size	uint64, optional	The size of the incoming upgrade package
flename	sring, optional	The name of the incoming upgrade package
saveFile	bool, optional	If one, the upgrade fle will be saved, otherwise it is deleted
crcFile	bool, optional	If true, will perform a CRC of the fle and compare to the client's CRC
force	bool, optional	If true, forces the upgrade to take place even if server is up-to-date
rpmforce	bool, optional	If true, forces the rpm to use the force option
shutdown	bool, optional	If true, all RSA NetWitness NextGen services will be shutdown and resarted after upgrade

vgs

Description: lis volume groups

Security.roles: appliance.manage Manual for vgs

vmsat

Description: Get CPU and VM satisics

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
count	uint32, optional, {unsigned:The value mus be between 1 and 30, inclusive}	The number of times to gather satisics and print the report. Default is 5

/connections node

API Messages

closeAll

Description: Closes all connections

Security.roles: connections.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, optional, {enum-any:The value mus be one or more of the following: native|res|amqp}	Indicates the type of connection to close, either native, res or amqp. Default is all types.
limit	uint32, optional	If non-zero, will sop closing connections after limit is reached. NOTE: It sarts closing the mos idle connections frs.
idleLongerThan	sring, optional	A time duration in HH:MM:SS or ##{d,h,m,s} format. Any connection that has been idle longer than the passed in duration is eligible to be closed. Connections that have seen activity within the duration window will not be closed. Examples: "01:05:10" (1 hour, 5 mins, 10 secs) or "65m" (65 minutes)
exceptMe	bool, optional	If true, will not close the connection that initiated the command.

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

netSpeed

Description: Network Speed Tes **Security.roles:**

connections.manage

Parameters:

Parameter	Type, Options	Description
size	size, optional, {byte-range:The value (in bytes, size modifers like MB are accepted) mus be between 1024 and 536870912, inclusive}	The size of each message sent, default is 64k
count	uint32, optional, {unsigned:The value mus be between 1 and 536870912, inclusive}	The number of messages to send, default is 16k

/logs node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

download

Description: Downloads log entries

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The sarting log number to download
id2	uint64, optional	The ending log number to download
time1	date-time, optional	The sarting log date to download, can be posix time (seconds since 1970) or DD-MM-YY HH:MM:SS
time2	date-time, optional	The ending log date to download, can be posix time (seconds since 1970) or DD-MM-YY HH:MM:SS
op	sring, {enum-one:The value mus be one of the following: sart|next|cancel}	The operation to perform (sart, next, cancel)
logTypes	sring, optional	The type(s) of log messages to retrieve (debug,info,audit,warning,failure)
match	sring, optional	Case insensitive sring to match in each log message
regex	sring, optional	Regular expression to match in each log message
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	The time format used in each log message, default is posix time (seconds since 1970)
batchSize	uint32, optional, {unsigned:The value mus be between 2000 and 10000, inclusive}	The number of logs to retrieve per batch, default is 2000

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

pull

Description: Downloads N log entries

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The frs log id number to retrieve, this is mutually exclusive with id2
id2	uint64, optional	The las log id number that will be sent, defaults to mos recent log message when id1 or id2 is not sent
count	uint32, optional, {unsigned:The value mus be between 1 and 10000, inclusive}	The number of logs to pull
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	The time format used in each log message, default is posix time (seconds since 1970)

timeRoll

Description: Delete log entries that exceed a given age

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
timeCalc	sring, optional, {enum-any:The value mus be one or more of the following: current|las-write}	The time calculation to use (current date or las write date), default is current
minutes	uint32, optional, {unsigned:The value mus be between 0 and 60, inclusive}	Remove log entries older than the given number of minutes
hours	uint32, optional, {unsigned:The value mus be between 0 and 24, inclusive}	Remove log entries older than the given number of hours
days	uint32, optional, {unsigned:The value mus be between 0 and 365, inclusive}	Remove log entries older than the given number of days
date	sring, optional	Remove log entries older than the given date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

/res node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated
		values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

/sys node

Manual for /sys

API Messages

caCert

Description: Display or delete trused CA certs

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: lis|delete|add}	CA action to perform (lis,delete,add)
id	sring, optional	ID of CA certifcate to remove when op=delete

Manual for caCert

count

Description: Returns the number of child nodes **Security.roles:** everyone

fleEdit

Description: View and edit sysem confguration fles

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: dir|get|getBackup|put|create|delete}	The operation to perform (dir, get, getBackup, put, create, delete)
type	sring, optional, {enum-one:The value mus be one of the following: parser|flter}	The type of fle, if not provided will check fle extension
flename	sring, optional, {length:The value length mus be between 1 and 64, inclusive}	File to create, edit, get or delete

Manual for fileEdit

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

peerCert

Description: Display or modify trused peer certifcates

Security.roles: sys.manage
Parameters:

Parameter	Type, Options	Description
op	sring, optional, {enum-one:The value mus be one of the following: lis|delete|add}	Operation to perform on the lis of trused peers (lis,delete,add)
id	sring, optional	ID of peer certifcate to remove when op=delete

Manual for peerCert

save

Description: Forces a save of sysem settings

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
force	uint32, optional	If 1, forces the confguration fle to saved, regardless if changes are detected, default is 0

Manual for save

servCert

Description: Display the current server certifcate in PEM format.

Security.roles: sys.manage Manual for

servCert

shutdown

Description: Stop the service

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
reason	sring, optional, {length:The value length mus be between 0 and 512, inclusive}	The reason for shutting down the service
cl	sring, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next resart

Manual for shutdown

satHis

Description: Retrieve hisorical sats from the sats db. Don't send time1/time2 to get bounding times about sats db. Supported wildcards are ? to match any single char, * to match zero or more characters, not including slash /, ** to match zero or more characters including slash /. **Security.roles:** sys.manage

Parameters:

Parameter	Type, Options	Description
time1	date-time, optional	The sarting time for retrieving sats
time2	date-time, optional	The ending time for retrieving sats
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	Specify the time format for each sat snapshot, default is posix (seconds since 1970)
include	sring, optional	Comma separated lis of sats to include (wildcards allowed)
exclude	sring, optional	Comma separated lis of sats to exclude (wildcards allowed, has precedence over include)
onChange	sring, optional	Comma separated lis of sats that reduce the result set to only when at leas one of the sats change from their previous value
reduce	bool, optional	If true, reduces data transmission by replacing sat pathnames with a shorthand and provides a lookup table as frs result
showAll	bool, optional	If true, changes each result to show all sats previously seen, even if the value hasn't changed. It has a moderate performance impact and is bes used with include, exclude or onChange parameters to reduce the returned result set size.

Manual for statHist

telemtry

Description: Returns telemetry information about the service

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
level	uint32, optional, {signed:The value mus be between 1 and 3, inclusive}	The level of information to be sent by the telemetry reques. A value of 1 (default) will send all sat nodes, a value of 2 will send sat and confg nodes, and a value of 3 will send sat nodes, confg nodes and usage metrics for rules, parsers and devices (where applicable).
include	sring, optional	Comma separated lis of nodes to include (wildcard usage is ok).
exclude	sring, optional	Comma separated lis of nodes to exclude (wildcard usage is ok). If "include" is also specifed, exclusions are applied to the set of nodes specifed by "include".
url	sring, optional	Endpoint URL to send the JSON response.
op	sring, optional, {enum-one:The value mus be one of the following: get|confgure}	The telemetry operation to execute. Default is 'get'.

/users node

API Messages

addOrMod

Description: Add a new user or update an exising user in the sysem

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#$%&'+-=?^_`{|}~.[]\}	The username to add or update, mus be alphanumeric or @!#$%&'+-=?^_`{|}~.[]\
password	sring, optional	The user's password
pwdIsHashed	bool, optional	Set to true if the password has already been hashed
groups	sring, optional	The groups the user belongs to
authType	sring, optional, {enum-one:The value mus be one of the following: netwitness|pam}	The authentication sysem to use, "netwitness" is the default
queryTimeout	uint32, optional, {unsigned:The value mus be between 0 and 43200, inclusive}	The maximum number of minutes that a query is allowed to execute for this user. Zero means unlimited.
displayName	sring, optional	Display name for this user account
email	sring, optional	Email address

		for this account
description	sring, optional	Description of this user account
queryPrefx	sring, optional	Query filter applied to every query performed by this user account
sessionThreshold	uint32, optional, {unsigned:The value mus be between 0 and 4294967295, inclusive}	Query optimization which will extrapolate the remaining session counts when they exceed this value

auths

Description: Get supported authentication types **Security.roles:** everyone

count

Description: Returns the number of child nodes **Security.roles:** everyone

delete

Description: Delete a user from the sysem

Security.roles: users.manage **Parameters:**

Parameter	Type, Options	Description
name	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#$%&'+-=?^_`{|}~.[]\}	The username to delete

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone
Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

unlock

Description: Unlock a locked out account

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	sring	The username or account to unlock

whoAmI

Description: Returns information about the authenticated user

Security.roles: everyone

Cloud service

Introduction

The NwCloud core service is part of the XDR Sensor package that is part of our SaaS ofering. NwCloud runs alongside NwDecoder and acts as a "go between" between NwDecoder and AWS. In other words, it acts as an aggregator that downloads session/meta information from NwDecoder, converts that information to JSON (for now), and then pushes that data to Kinesis.

Device Regisration

In order to connect to AWS, the NwCloud service mus be regisered using a Device Activation Package (or DAP). The DAP is a fle named `device-actvation-package.json` , and can be downloaded from the tenant's portal. The fle can then be either uploaded to the NwCloud service using the activate message, or it can manually be placed in NwCloud's confguration folder.

The activate command can be used through the REST service at the endpoint `/cloud?msg=activate` This requires a program like Posman that will allow you include the DAP's JSON in the body of the reques. In this case the Content-Type of the reques mus be `application/octet-stream` . Once the reques is sent, the DAP fle will simply be written to NwCloud's confguration folder.
Alternatively, users can SCP or copy the DAP into NwCloud's confguration folder manually.

API Usage

Cloud service provides API interface through its tree form as nodes, node messages and child nodes.

- /cloud node messages:
  - `activate` the NwCloud sensor `refresh` the AWS
  - Credentials

`Note` : The API Messages section for the path nodes provide more details on API messages available and any associated manual pages.

/cloud node

API Messages

activate

Description: Activate the NwCloud sensor **Security.roles:**

cloud.manage

add

Description: Add a new device for aggregation

Security.roles: cloud.manage **Parameters:**

Parameter	Type, Options	Description
device	sring, optional, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to add (ip:port or [ipv6]:port), either this or ip mus be set
ip	sring, optional	The IP address of the device, either this or device mus be set
port	uint32, optional, {unsigned:The value mus be between 0 and 65535, inclusive}	The port number the device is lisening on
username	sring	The username to run as
password	sring, optional	The account password. Assumes service trus aggregation if no password is provided.
options	sring, optional	The device options
ssl	sring, optional	If true, the device requires SSL to connect
backup	sring, optional	If true, the device will be used as a backup
reconnect	sring, optional	If true, will automatically reconnect when not connected
name	sring, optional	Device collection name if device is a collection

count

Description: Returns the number of child nodes **Security.roles:** everyone

delete

Description: Delete an exising aggregation device

Security.roles: cloud.manage **Parameters:**

Parameter	Type, Options	Description
device	sring, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to delete
name	sring, optional	Device collection name if device is a collection

edit

Description: Edit an exising aggregation device

Security.roles: cloud.manage **Parameters:**

Parameter	Type, Options	Description
device	sring, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to edit
username	sring, optional	The username to run as
password	sring, optional	The account password. For service trus aggregation you mus not provide this parameter
options	sring, optional	The device options
ssl	sring, optional	If true, the device requires SSL to connect
backup	sring, optional	If true, the device will be used as a backup
reconnect	sring, optional	If true, will automatically reconnect when not connected
name	sring, optional	Device collection name if device is a collection
switchToTrusedAuth	bool, optional	If true, removes any password associated with this account and will use trused authentication henceforth. When switching to trused, only the 'device' parameter is allowed.

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

reconfg

Description: Calculates optimal settings for cloud pools and bufers based on the insalled hardware.

Security.roles: cloud.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the respective cloud settings, otherwise it will jus output the calculations
memory	size, optional	If non-zero, will use that setting as insalled physical RAM (e.g., memory="24 GiB"). This is useful for hybrids and/or all-in-one machines where the service will need to share memory.

refresh

Description: Refresh the AWS Credentials

Security.roles: cloud.manage **Parameters:**

Parameter	Type, Options	Description
details	bool, optional	If not specifed (default) then AWS credentials will be refreshed. Otherwise show the time of the next expiration.

resetMax

Description: Resets all max sats, including device max sats back to zero. **Security.roles:**

cloud.manage

sart

Description: Starts aggregation **Security.roles:**

cloud.manage

satus

Description: Change the online/ofine satus of an aggregation device

Security.roles: cloud.manage **Parameters:**

Parameter	Type, Options	Description
device	sring, {ip-address:The value mus be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to change satus
satus	sring, optional, {enum-one:The value mus be one of the following: online|ofine}	Whether to take device ofine or put online
name	sring, optional	Device collection name if device is a collection

sop

Description: Stops aggregation

Security.roles: cloud.manage **Parameters:**

Parameter	Type, Options	Description
fush	bool, optional	If true (default), process all sessions and packets currently in memory. Otherwise, sop immediately, discarding unprocessed sessions and packets.

/connections node

API Messages

closeAll

Description: Closes all connections

Security.roles: connections.manage **Parameters:**

Parameter	Type, Options	Description
type	sring, optional, {enum-any:The value mus be one or more of the following: native|res|amqp}	Indicates the type of connection to close, either native, res or amqp. Default is all types.
limit	uint32, optional	If non-zero, will sop closing connections after limit is reached. NOTE: It sarts closing the mos idle connections frs.
idleLongerThan	sring, optional	A time duration in HH:MM:SS or ##{d,h,m,s} format. Any connection that has been idle longer than the passed in duration is eligible to be closed. Connections that have seen activity within the duration window will not be closed. Examples: "01:05:10" (1 hour, 5 mins, 10 secs) or "65m" (65 minutes)
exceptMe	bool, optional	If true, will not close the connection that initiated the command.

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

netSpeed

Description: Network Speed Tes **Security.roles:**

connections.manage,aggregate

Parameters:

Parameter	Type, Options	Description
size	size, optional, {byte-range:The value (in bytes, size modifers like MB are accepted) mus be between 1024 and 536870912, inclusive}	The size of each message sent, default is 64k
count	uint32, optional, {unsigned:The value mus be between 1 and 536870912, inclusive}	The number of messages to send, default is 16k

/logs node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

download

Description: Downloads log entries

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The sarting log number to download
id2	uint64, optional	The ending log number to download
time1	date-time, optional	The sarting log date to download, can be posix time (seconds since 1970) or DD-MM-YY HH:MM:SS
time2	date-time, optional	The ending log date to download, can be posix time (seconds since 1970) or DD-MM-YY HH:MM:SS
op	sring, {enum-one:The value mus be one of the following: sart|next|cancel}	The operation to perform (sart, next, cancel)
logTypes	sring, optional	The type(s) of log messages to retrieve (debug,info,audit,warning,failure)
match	sring, optional	Case insensitive sring to match in each log message
regex	sring, optional	Regular expression to match in each log message
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	The time format used in each log message, default is posix time (seconds since 1970)
batchSize	uint32, optional, {unsigned:The value mus be between 2000 and 10000, inclusive}	The number of logs to retrieve per batch, default is 2000

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

pull

Description: Downloads N log entries

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
id1	uint64, optional	The frs log id number to retrieve, this is mutually exclusive with id2
id2	uint64, optional	The las log id number that will be sent, defaults to mos recent log message when id1 or id2 is not sent
count	uint32, optional, {unsigned:The value mus be between 1 and 10000, inclusive}	The number of logs to pull
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	The time format used in each log message, default is posix time (seconds since 1970)

timeRoll

Description: Delete log entries that exceed a given age

Security.roles: logs.manage **Parameters:**

Parameter	Type, Options	Description
timeCalc	sring, optional, {enum-any:The value mus be one or more of the following: current|las-write}	The time calculation to use (current date or las write date), default is current
minutes	uint32, optional, {unsigned:The value mus be between 0 and 60, inclusive}	Remove log entries older than the given number of minutes
hours	uint32, optional, {unsigned:The value mus be between 0 and 24, inclusive}	Remove log entries older than the given number of hours
days	uint32, optional, {unsigned:The value mus be between 0 and 365, inclusive}	Remove log entries older than the given number of days
date	sring, optional	Remove log entries older than the given date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

/res node

API Messages

count

Description: Returns the number of child nodes **Security.roles:** everyone

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated
		values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

/sys node

Manual for /sys

API Messages

caCert

Description: Display or delete trused CA certs

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: lis|delete|add}	CA action to perform (lis,delete,add)
id	sring, optional	ID of CA certifcate to remove when op=delete

Manual for caCert

count

Description: Returns the number of child nodes **Security.roles:** everyone

fleEdit

Description: View and edit sysem confguration fles

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
op	sring, {enum-one:The value mus be one of the following: dir|get|getBackup|put|create|delete}	The operation to perform (dir, get, getBackup, put, create, delete)
type	sring, optional, {enum-one:The value mus be one of the following: parser|flter}	The type of fle, if not provided will check fle extension
flename	sring, optional, {length:The value length mus be between 1 and 64, inclusive}	File to create, edit, get or delete

Manual for fileEdit

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

peerCert

Description: Display or modify trused peer certifcates

Security.roles: sys.manage
Parameters:

Parameter	Type, Options	Description
op	sring, optional, {enum-one:The value mus be one of the following: lis|delete|add}	Operation to perform on the lis of trused peers (lis,delete,add)
id	sring, optional	ID of peer certifcate to remove when op=delete

Manual for peerCert

save

Description: Forces a save of sysem settings

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
force	uint32, optional	If 1, forces the confguration fle to saved, regardless if changes are detected, default is 0

Manual for save

servCert

Description: Display the current server certifcate in PEM format.

Security.roles: sys.manage Manual for servCert

shutdown

Description: Stop the service

Security.roles: sys.manage **Parameters:**

Parameter	Type, Options	Description
reason	sring, optional, {length:The value length mus be between 0 and 512, inclusive}	The reason for shutting down the service
cl	sring, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next resart

Manual for shutdown

satHis

Description: Retrieve hisorical sats from the sats db. Don't send time1/time2 to get bounding times about sats db. Supported wildcards are ? to match any single char, * to match zero or more characters, not including slash /, ** to match zero or more characters including slash /. **Security.roles:** sys.manage **Parameters:**

Parameter	Type, Options	Description
time1	date-time, optional	The sarting time for retrieving sats
time2	date-time, optional	The ending time for retrieving sats
timeFormat	sring, optional, {enum-one:The value mus be one of the following: posix|simple}	Specify the time format for each sat snapshot, default is posix (seconds since 1970)
include	sring, optional	Comma separated lis of sats to include (wildcards allowed)
exclude	sring, optional	Comma separated lis of sats to exclude (wildcards allowed, has precedence over include)
onChange	sring, optional	Comma separated lis of sats that reduce the result set to only when at leas one of the sats change from their previous value
reduce	bool, optional	If true, reduces data transmission by replacing sat pathnames with a shorthand and provides a lookup table as frs result
showAll	bool, optional	If true, changes each result to show all sats previously seen, even if the value hasn't changed. It has a moderate performance impact and is bes used with include, exclude or onChange parameters to reduce the returned result set size.

Manual for statHist

telemtry

Description: Returns telemetry information about the service

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
level	uint32, optional, {signed:The value mus be between 1 and 3, inclusive}	The level of information to be sent by the telemetry reques. A value of 1 (default) will send all sat nodes, a value of 2 will send sat and confg nodes, and a value of 3 will send sat nodes, confg nodes and usage metrics for rules, parsers and devices (where applicable).
include	sring, optional	Comma separated lis of nodes to include (wildcard usage is ok).
exclude	sring, optional	Comma separated lis of nodes to exclude (wildcard usage is ok). If "include" is also specifed, exclusions are applied to the set of nodes specifed by "include".
url	sring, optional	Endpoint URL to send the JSON response.
op	sring, optional, {enum-one:The value mus be one of the following: get|confgure}	The telemetry operation to execute. Default is 'get'.
arn	sring, optional	SNS ARN to send the JSON response.
decoderInclude	sring, optional	Nodes to include on Decoder

/users node

API Messages

addOrMod

Description: Add a new user or update an exising user in the sysem

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#$%&'+-=?^_`{|}~.[]\}	The username to add or update, mus be alphanumeric or @!#$%&'+-=?^_`{|}~.[]\
password	sring, optional	The user's password
pwdIsHashed	bool, optional	Set to true if the password has already been hashed
groups	sring, optional	The groups the user belongs to
authType	sring, optional, {enum-one:The value mus be one of the following: netwitness|pam}	The authentication sysem to use, "netwitness" is the default
queryTimeout	uint32, optional, {unsigned:The value mus be between 0 and 43200, inclusive}	The maximum number of minutes that a query is allowed to execute for this user. Zero means unlimited.
displayName	sring, optional	Display name for this user account
email	sring, optional	Email address

		for this account
description	sring, optional	Description of this user account
queryPrefx	sring, optional	Query filter applied to every query performed by this user account
sessionThreshold	uint32, optional, {unsigned:The value mus be between 0 and 4294967295, inclusive}	Query optimization which will extrapolate the remaining session counts when they exceed this value

auths

Description: Get supported authentication types **Security.roles:** everyone

count

Description: Returns the number of child nodes **Security.roles:** everyone

delete

Description: Delete a user from the sysem

Security.roles: users.manage **Parameters:**

Parameter	Type, Options	Description
name	sring, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#$%&'+-=?^_`{|}~.[]\}	The username to delete

help

Description: Describes this node and its supported messages. NOTE: Command parameters are passed as ="" ="" etc., where mus be in double quotes if there is whitespace. To pass a quote in the value, you mus escape it by preceding it with a backslash \.

Security.roles: everyone
Parameters:

Parameter	Type, Options	Description
msg	sring, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrsuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	sring, optional, {enum-one:The value mus be one of the following: messages|parameters|description|values|roles|extra|manual}	The specifc help operation to perform (e.g., op=manual would return a man page on this node or the specifed message)
format	sring, optional, {enum-one:The value mus be one of the following: default|xml|html}	The format of the response, default returns in a human friendly format

info

Description: Returns detailed information about the node **Security.roles:** everyone

ls

Description: Returns the lis of child nodes

Security.roles: everyone **Parameters:**

Parameter	Type, Options	Description
depth	uint32, optional	How many levels deep to return node info, default is 1
options	sring, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like confg, sat, folder, session, connection, channel, resart-needed or pretty-print.
exclude	sring, optional	Comma separated lis of nodes or node pathnames to exclude (wildcards allowed: * and ?)

unlock

Description: Unlock a locked out account

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	sring	The username or account to unlock

whoAmI

Description: Returns information about the authenticated user

Security.roles: everyone

Manuals

/database

The Database tree holds the main functionality related to the session, meta and/or packet/log databases.

The following is a brief overview of the Database commands and what they do:

- **hashInfo** - Retrieves hash information for database files that containing session/meta/packet objects for a set of sessions or date range. **dbState** -
- Returns comprehensive information about the current database state or persists the current state to disk for fast reload. **dump** - Provides
- comprehensive information about a single session in the databases.
- **manifest** - If a manifest directory is defined, it will allow operations on the manifest files (such as a time based query) for database files in cold storage. **optimize** - Runs a series of tests to determine the optimal **.write.block.size** based on configured drives and using data from existing databases. **reconfig** -
- Reconfigure database settings based on hardware changes or passed in memory sizes. **resetMax** - Resets all max database stats or just the ones listed.
- **sizeRoll** - Delete database files based on the total size of all databases (passed with 'type' parameter) or space remaining on shared volume(s). This command
- should not be used on databases that don't share storage. **stagger** - Staggers database files to optimize read/write performance across multiple volumes. Typically
- used after adding an empty mount point. **timeRoll** - Delete database files that exceed a given age. **wipe** - Overwrites all packets and/or meta for a session with a pattern (for eliminating sensitive information).
-
-
-

/database dbState

This command is used internally to persis the sate of the databases so they can be reloaded in seconds upon resart. There is an automatically scheduled command to write out the sate every so often and it will always be written out on a normal shutdown. There are also options to view key information about a specifc database that can be useful when troubleshooting.

Supported parameters:

- **op** - The operation to perform. *save* will persist the database state. *summary* returns a short summary of key information about a specific database (see *type* parameter). *filelist* gives detailed information about a database.
- **type** - The database(s) to perform the operation on. One or more of: *session*, *meta* or *packet*. Separate multiple databases with a comma. **options** - Two
- options that affect returned file sizes: *bytes* or *pretty-print*. *pretty-print* returns sizes in human readable formats like GBs.

/database hashInfo

This command can be used to retrieve the hash information for database fles. This is only useful if hashing has been enabled (see confguration for *hash.algorithm*, *hash.databases* and *hash.dir*).

- **sessions** - A comma delimited list of sessions and session ranges to retrieve file hashes for. Use this or *beginDate*, *endDate* but not both. **beginDate** - The
- beginning of a date range (YYYY-MM-DD HH:MM:SS) of database file creation dates to retrieve file hashes for.
- **endDate** - The end of a date range (YYYY-MM-DD HH:MM:SS) of database file creation dates to retrieve file hashes for. **directories** - A semi-colon
- delimited list of additional directories to search for hash files. This may be used to account for changes to the hash.dir configuration.

/database manifes

Manifes fles are created with every session, meta, and packet (log) DB fle and index slice directory. A manifes fle is a fle that describes several key pieces of information about the data to which it refers. Manifes fles are written as a JSON record. Manifes fles travel with the data they represent from tier to tier.

If the data they represent is deleted, the manifes fle is also deleted, except in the following special case. If the service has `/database/config/manifest.dir` confgured to a valid directory, at the point when the manifes data is deleted, a copy of the manifes fle is placed into the directory pointed at by `manifest.dir` (the directory is created if it does not exis). This enables a NetWitness Platform feature called hisorical manifes searching.

The intention of this process is to keep hisorical manifes fles for years, in one location for ofine querying. As you might imagine from a service running for many years, this can potentially generate hundreds of thousands of fles. This should not be a concern however, as the service automatically compresses fles into a single archive in order to save space when they grow too numerous. Manifes fles are very small and compress well.

- **op** - The operation to perform (defaults to query). Other option is 'compress', which will compress all stored manifest files.
- **time1** - The beginning time (UTC) for matching offline database files (YYYY-MM-DD HH:MM:SS) **time2** - The
- ending time (UTC) for matching offline database files (YYYY-MM-DD HH:MM:SS) **timeFormat** - Specify the
- time format that is returned (posix, simple), default is posix Parameters:

Example: manifes time1="2014-04-20 11:00:00" time2="2014-04-11 11:20:00" timeFormat=simple More information

on manifess can be found in the Core Database Tuning Guide.

/database reconfg

Calculates new drive sizes and free space for the session, meta and/or packet directories. No directories are removed and the assumption is each directory is mounted on a separate flesysem and will only be used for sorage of that database.

Parameters:

- **type** - The database types to reconfigure (session, meta or packet), default is all.
- **update** - If true (default is false), will automatically update the respective settings, otherwise it will just output the calculations for viewing.
- **percent** - The drive percentage to use for the calculated size per directory, default is 95. **op** - (PACKET
- DECODER ONLY) The operating mode, 'normal' or '10g'. Default is normal.

If you want to apply the recommended settings, then make sure to pass "update=true".

/database sizeRoll

The sizeRoll command is used to keep two or more databases synchronized wrt data retention. Under normal conditions, each database (session, meta, packet or index) is responsible for managing it's own size and when data should be aged out. However, while this is good for preventing disks from flling up, it's not ideal when one database has too much retention and the oldes data could be orphaned because the other database is aging out faser. This is where the *sizeRoll* command comes into play.

Parameters:

- **type** - The databases (session, meta or packet) to consider for removing the oldest data based on total size or space remaining. **log** - If true, will
- log all actions taken, otherwise all activity will happen silently. Default is true.
- **maxSize** - The maximum size of all databases passed in 'type' parameter combined. When exceeded, oldest data is deleted first until total size is less than maxSize. This applies only to the Hot tier.
- **maxSizeWarm** - The maximum size of all databases passed in 'type' parameter combined. When exceeded, oldest data is deleted first until total size is less than maxSize. This applies only to the Warm tier.
- **maxPercent** - The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When exceeded, oldest data is deleted first until total size is less than maxPercent of total volumes. This applies only to the Hot tier.
- **maxPercentWarm** - The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When exceeded, oldest data is deleted first until total size is less than maxPercent of total volumes. This applies only to the Warm tier.
- **minFree** - The minimum allowed free space on all volumes for all databases in 'type' parameter before oldest data is deleted. When minimum free space drops below target, the oldest data is deleted first until free space once again exceeds target. This only applies to the Hot tier.
- **minFreeWarm** - The minimum allowed free space on all volumes for all databases in 'type' parameter before oldest data is deleted. When minimum free space drops below target, the oldest data is deleted first until free space once again exceeds target. This only applies to the Warm tier.

sizeRoll is used to keep the specifed databases aging out at roughly the same time period. In order for this to work, the databases being managed by *sizeRoll* mus all shared the same sorage space(s). Since they share the same sorage, there is a tradeof in I/O performance. Typically, Archiver uses the *sizeRoll* command to manage it's space, as maximum retention is more important that I/O performance.

There are two *sizeRoll* commands, one on the *database* node and one on the *index* node. The only diference is the *index* node command implicitly includes managing the index size. If you use the one on the *database* node, it will not manage the index size.

The *sizeRoll* command is per use. The typical use case is to set it up as a scheduled command so it runs periodically, aging out fles based on your criteria. For more information on running scheduled commands, see the Core Database Tuning Guide.

/database sagger

The sagger command is typically only useful for a 10G Decoder and usually jus for the packet database. Maximum performance is achieved for soring and retrieving packets when multiple volumes are present. In this scenario, the Decoder always flls the volume with the mos free space. When the volumes are roughly the same size, this results in a saggered write pattern, which allows maximum throughput for reading and writing across all volumes. However, this only naturally occurs when multiple packet sorage volumes are present at the time the Decoder is frs deployed.

A typical use case is adding more sorage to an exising Decoder to increase retention. However, when adding sorage to an deployment that has already flled the exising volumes with sored packets, the Decoder will naturally fll the new sorage with packets before rolling out any packets on the exising sorage. This results in a suboptimal read/write pattern because mos reads will occur on the same volume that is currently being written to. In a 10G deployment, reads are blocked from the volume when writes are occurring. This doesn't sop ALL reads on that volume, because the fle is bufered in memory before being written, but it does result in suboptimal read performance.

With the **sagger** command, you can add more sorage and then have the service naturally sagger the fles across ALL volumes (exising and new) so that read performance is optimized. This command should only be performed AFTER the sorage is mounted and the Decoder confgured to use it (e.g., after adding the mount point(s) to *packet.dir*). The downside to this command is it can take some time to sagger and the Decoder should **not** be capturing during the sagger operation.

Recommended workfow:

1. Add all storage and configure mount points
2. Add new storage mount points to packet.dir (or session.dir/meta.dir) and restart service (very important!)
3. Ensure capture is stopped
4. Run stagger operation but make sure the connection that initiated the stagger operation is never terminated until the operation is complete! If the connection is terminated, then the stagger operation will be canceled. If the operation is canceled, the files that were already staggered will remain in place. The operation can be resumed by rerunning the same command (the work already done will not need to be done again). If running **stagger** from NwConsole, run the "timeout 0" command before sending the stagger command. This will prevent the normal 30 second command timeout.
5. Start capture after stagger command finishes.

The following are the parameters for the command:

- **type** - The database that will be staggered (session, meta or packet). Typically only the packet database is useful for staggering, but it is possible to do the session or meta database when multiple volumes are present for those databases. Since the session and meta databases write far less data than the packet db, typically staggering those databases results in less noticeable performance gains.
- **dryRun** - If true (the default), will only return a description of the operations that would be performed. If false, then the files will actually be moved to an optimal read/write pattern. You MUST pass false to actually stagger the files!

Example usage from NwConsole:

```
login <decoder>:50004 <username> <password> timeout 0
send /database stagger type=packet dryRun=false
```

If you run this command via the RESTful API, please pass the additional parameter **expiry=0** to prevent a timeout from the service. You will also need to ensure the HTTP client does not disconnect before the operation completes.

/database timeRoll

The *timeRoll* command is used to limit the amount of data based on the age of the data when it was written. For insance, if you have a requirement to only sore packet data for 7 days, but no longer than that, then you can use the timeRoll command to "age out" packets that were written more than 7 days ago.

Parameters:

- **type** - The database(s) to age out. Can be one or more of session, meta or packet.
- **timeCalc** - The time calculation to use ("current" date or "last-write" date). Default is current. Unless capture or aggregation is stopped, both settings usually amount to pretty much the same thing.
- **minutes** - Remove database files older than the given number of minutes. Minutes are additive if *hours* or *days* is also supplied. **hours** -
- Remove database files older than the given number of hours. Hours are additive if *minutes* or *days* is also supplied. **days** - Remove database
- files older than the given number of days. Days are additive if *minutes* or *hours* is also supplied.
- **date** - Remove database files older than the given UTC date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters.

Example:

```
timeRoll type=session,meta,packet timeCalc=current hours=12 days=3
```

The above command will age out all session, meta and packet fles that are older than 3.5 days from the current time. If you sop capture, then it will continue to age out the fles until only the mos recent fle is left. It is not permitted to remove the mos recent fle being written, so it's not possible to completely remove all data using the *timeRoll* command. To do that, you would need to run the "reset" command on the /decoder (or /<service> node for other services). If you wanted to always keep 3.5 days worth of data, even if capture is sopped, then change the timeCalc parameter to be "las-write".

The *timeRoll* command is per use. The typical use case is to set it up as a scheduled command so it runs periodically, aging out fles based on your criteria. For more information on running scheduled commands, see the Core Database Tuning Guide.

/decoder

The Decoder tree holds the main functionality related to capture, parsing and confguration of those processes.

The following is a brief overview of the Decoder commands and what they do:

- **sslKeys** - Provides an interface to upload premaster keys, TLS 1.3 keys or private keys so that the captured encrypted packets that match the keys can be decrypted before the parsing step. **start** - Starts capture of the selected device **stop** - Stops capture and flushes any remaining data in the pipeline. This can take
- several minutes.
- **reset** - Used to wipe the device of all data, rebuild the index, or start from a fresh configuration. Use with caution! **whoAgg** -
- Returns information on all the services that are currently aggregating from this Decoder. **select** - Used to select the capture
- device to use for capture. **import** - Import raw data directly thru this interface. Cannot be performed while normal capture is
- running. **reconfig** - Used to perform automatic ("easy") configuration of the service based on the input parameters. **resetMax** -
- Returns all statistics to zero.
-
-

/decoder reset

Used to wipe the device of all data, rebuild the index, or sart from a fresh confguration. Use with caution!

/decoder select

Used to select the capture device to use for capture.

/decoder sslKeys

This command is all about decrypting incoming packets, before the parsing sep, so that the enabled parsers will see the unencrypted packet payload and create meta accordingly. Otherwise, mos parsers will only see encrypted garbage and will fail to create meaningful metadata. Decrypting packets in real-time requires a non-trivial amount of extra work in the parsing sage. You should plan accordingly by making sure the incoming trafc bandwidth does not overwhelm the available compute power. In other words, you may need more Decoders to decrypt trafc than you would if not decrypting.

Packets captured on a Decoder normally have a timeout of around 60 seconds (in the assembly sage) before they are sent to the parse sep. If the Decoder is under memory pressure (very high bandwidth), then the lifetime of the packets in Assembler can be shortened. This timeout can be adjused through confguration and by increasing the amount of memory available to hold packets in Assembly. Regardless, in order to perform decryption of the packets, the decryption key mus be received by the Decoder before the parsing sage.

```
NOTE: Currently with default builds TLS 1.2 and earlier protocols can be decrypted and you must have the native
HTTPS (not Lua) parser enabled. This parser is port agnostic and will attempt to decrypt any SSL/TLS traffic
when it has the corresponding decryption key. If FIPS is enabled, this will restrict the list of ciphers for
decryption to only those that are FIPS approved.
```

```
 TLS 1.3 protocol decryption is ONLY supported in nonfips builds.
```

The following types of encryption keys can be used:

- **premaster** - This is the ephemeral symmetric key actually used in the TLS payload stream for encryption and decryption. **private** - The
- asymmetric private key used during the TLS handshake that encrypts the premaster.
- **tls 1.3 keys** - The TLS 1.3 ephemeral keys which are used in TLS handshake and payload stream encryption and decryption. In the decoder TLS 1.3 decryption is only supported with nonfips builds.

Premaser

The *premaser* key is generated randomly and is ephemeral for the life of one specifc TLS session. Normally, there is not an easy way to get *premaser* keys to a Decoder in time for the parsing sep. However, both Chrome and Firefox can write the premaser keys they generate to a fle. This is useful for tesing purposes. To confgure your browser to do this, all you have to do is create an environment variable called SSLKEYLOGFILE and assign it the pathname

of a text fle to write the keys to. Decoder will accept the fle exactly as it is written and will use all the decryption keys in the fle for any encrypted trafc it captures. The following is a sample NwConsole script that uploads the fle to a Decoder:

```
login <decoder>:50004 <username> <password> send /decoder
sslKeys --file-data=SSLKeys.txt
```

or you could use the following curl command (with the RESTful port):

```
curl -u "<username>:<password>" -H "Content-Type: application/octet-stream" --data-binary
@"/path/SSLKeys.txt" -X POST "http://<hostname>:50104/decoder?msg=sslKeys"
```

Once the symmetric keys are uploaded, they will immediately be used for any necessary decryption. Symmetric keys are sored in memory and there is a limit to how many can be sored at any point in time. As more are added, the earlies keys will be aged out. You can also add premaser keys by jus passing the *random* and *premaser* parameters to **sslKeys**.

TLS 1.3 Keys

TLS 1.3 protocol uses a set of ephemeral symmetric keys also called as key secrets for the life of one specifc TLS 1.3 session. The TLS 1.3 session also encrypts handshake, all the TLS records after Server Hello are encrypted , and at each sage a corresponding key is used for the encryption e.g: client handshake, server handshake, client application and server application etc.. . So in order for decrypting a TLS 1.3 session all the required session keys are needed to be available.

```
NOTE: The asymmetric private key decryption is not supported by TLS 1.3 protocol and only perfect forward
secrecy through ephemeral keys is supported.
```

Similar to *premaser* key, capturing TLS 1.3 keys and uploading to Decoder in time for parsing is not an easy way. However, well known browsers like Chrome , Firefox and IE can write the TLS 1.3 keys they generate to a fle and this is useful for tesing purposes. To confgure your browser to do this, create an environment variable called SSLKEYLOGFILE and set with a pathname to text fle to write the keys.
Decoder will accept the fle exactly as it is written and will use all the decryption keys in the fle for any encrypted trafc it captures.

The following are keys used in TLS 1.3 session:

- **CETS** : The CLIENT_EARLY_TRAFFIC_SECRET is used by TLS client to encrypt and send early payload there by notifying server that client would like to use TLS 1.3 and also saves a round trip. Hence it is also termed as 0-RTT.
- **CHTS** : The CLIENT_HANDSHAKE_TRAFFIC_SECRET is used by TLS client to encrypt handshake records.
- **SHTS** : The SERVER_HANDSHAKE_TRAFFIC_SECRET is used by TLS server to encrypt subsequent handshake.
- **CTS0** : The CLIENT_TRAFFIC_SECRET_0 is the first application traffic key used by TLS client to encrypt application payload of the session. Subsequent application keys are recomputed based on previous application keys on Key Update request.
    - **STS0** : The SERVER_TRAFFIC_SECRET_0 is the first application traffic key used by TLS server to encrypt application payload of the session. Subsequent application keys are recomputed based on previous application keys on Key Update request.

The following is a sample NwConsole script that uploads the fle to a Decoder:

```
login <decoder>:50004 <username> <password> send
/decoder sslKeys --file-data=SSLKeys.txt
```

```
Optionally the parameters can be passed in as list in the upload file, ex: send
/decoder sslKeys --file-data=SSLKeys.txt --file-format=params-list
```

or you could use the following curl command (with the RESTful port):

```
curl -u "<username>:<password>" -H "Content-Type: application/octet-stream" --data-binary
@"/path/SSLKeys.txt" -X POST "http://<hostname>:50104/decoder?msg=sslKeys"
```

Once the symmetric keys are uploaded, they will immediately be used for any necessary decryption. Symmetric keys are sored in memory and there is a limit to how many can be sored at any point in time. As more are added, the earlies keys will be aged out. You can also add available TLS 1.3 keys by jus passing the *random*, *CETS* , *CHTS* , *SHTS* , *CTS0* and *STS0* parameters to **sslKeys**.

```
NOTE: For most of the sessions the CLIENT_EARLY_TRAFFIC_SECRET key may not be generated or used by TLS clients
and it is normal. However, if handshake and application keys are not generated and not available then TLS 1.3
decryption wouldn't be successful.
```

Private Keys or PEM fles

Private keys are normally sored in PEM fles and are the asymmetric keys generated by services that accept TLS trafc. These keys are used during the TLS handshake to encrypt the premaser symmetric key that will be used for the res of the payload encryption. For example, if you have a web server whose trafc you want visibility into, then you would upload the private key it uses to encrypt trafc. You only need to do this once, as it is sored permanently (or until a delete command). Once insalled, all TLS handshakes that use that private key will be able to be decrypted by the Decoder. After upload, a parser reload command needs to be issued so the newly insalled key becomes visible to the HTTPS parser. The following are some sample commands that will upload a PEM fle for decryption.

```
NOTE: Not all ciphers suites compute the symmetric key using the RSA private key (e.g., Ephemeral Diffie
Hellman, which is the E in DHE and ECDHE, has forward secrecy).  Encrypted traffic with those ciphers cannot be
decrypted unless the premaster key is uploaded to Decoder before the session is parsed.  For those ciphers, the
RSA key is only used to sign the ephemeral keys to protect the key exchange against MITM attacks.
```

For NwConsole:

```
send /decoder sslKeys pemFilename=MyKey.pem --file-data=/path/MyKey.pem
```

Using the RESTful interface (you mus provide the pemFilename parameter in the URL):

```
curl -u "<username>:<password>" -H "Content-Type: application/octet-stream" --data-binary
@"/path/MyKey.pem" -X POST "http://<hostname>:50104/decoder?msg=sslKeys&pemFilename=MyKey.pem"
```

> Private keys are automatically encrypted before storing to protect them.

Key Management

The following is the full lis of parameters that can be used with the **sslKeys** command:

- **clear** - Removes all premaster keys from memory. Will not delete any PEM files installed on the system. **maxKeys** - Changes the maximum
- number of premaster keys that will be stored in memory. **listPems** - Returns a list of all installed private key PEM files. **deletePem** - Deletes the
- named PEM file from the file system. You can pass this parameter more than once to remove multiple files. **random** - The random hash used to
- identify the premaster key or TLS 1.3 keys.
- **premaster** - The premaster key that will be installed for the previous random parameter. They must show up in pairs and random must be first. **CETS** - The TLS
- 1.3 client early traffic secret key that will be installed for the previous random parameter. It must show up with random parameter.
- **CHTS** - The TLS 1.3 client handshake traffic secret key that will be installed for the previous random parameter. It must show up with random parameter.
  SHTS - The TLS 1.3 server handshake traffic secret key that will be installed for the previous random parameter. It must show up with random parameter.
- **CTS0** - The TLS 1.3 client traffic secret 0 key that will be installed for the previous random parameter. It must show up with random parameter.
  STS0 - The TLS 1.3 server traffic secret 0 key that will be installed for the previous random parameter. It must show up with random parameter.
-
  Example: The random and TLS 1.3 keys can be passed as space separated parameters to sskKeys, ex: *random*
-
- =<value> *CETS* =<value> *CHTS* =<value> *SHTS* =<value> *CTS0* =<value> *STS0* =<value>

Return Values

Mos commands return name/value pairs of sats about the symmetric keys in memory. The following explains what they mean:

- **added** - The number of premaster and TLS 1.3 keys just added during this command **total** - The total number of premaster and TLS 1.3 keys
- loaded in memory **agedOut** - The total number of premaster and TLS 1.3 keys that were removed during this command (this is not a lifetime
- stat) **maxKeys** - The current maximum allowed premaster and TLS 1.3 keys **premaster** - The total number of premaster keys loaded in
- memory. **tls1.3** - The total number of TLS 1.3 keys loaded in memory.
-
-

Viewing Unencrypted Trafc

Even though the packets are decrypted during the parse sage, only the encrypted packets are written to disk. However, the matching premaser key used for decrypting is written to the *tls.premaser* meta key. This key can then be used to subsequently decrypt the packets on the fy.

Incase of TLS 1.3 session the matching keys are written to the following meta and these keys can then be used to subsequently decrypt the packets on the fy.

- *tls.client.early* : Hex encoded client early traffic key *tls.client.hdshk* : Hex
- encoded client handshake traffic key *tls.server.hdshk* : Hex encoded server
- handshake traffic key *tls.client.app* : Hex encoded client application traffic key
- *tls.server.app* : Hex encoded server application traffic key
-

One Decoder API that can be used to see this is the /sdk/content RESTful service. All you need to know is the Session ID of the encrypted packets and the *fags* parameter masked to the value 128 (or 0x80 in hex). Point your browser to the Decoder's RESTful interface and type in something like this:

```
http://<decoder>:50104/sdk/content?session=<id>&flags=128&render=text
```

You should get back a simple web page showing the packets after they are decrypted. Remove the *fags* parameter to see what it looks like encrypted (or at leas remove the 0x80 mask). For more information on the /sdk/content service, see the manual page for /sdk content.

/decoder sart

Starts capture on the selected interface.

/decoder sop

Stops capture and fushes all data currently in the pipeline.

/decoder whoAgg

Returns information on all the services that are currently aggregating from this Decoder.

/sdk

The SDK is the primary means to access parsed metadata and the raw data that generated it. There are three main mechanisms for performing queries in the database, the *query*, *values*, and *msearch* commands. Mos SDK commands over the RESTful interface override the default expiry of 30 seconds to be unlimited. Why? Because there are already mechanisms in place to cancel long running queries based on confgured settings and having a small expiry value only causes confusion.

The following is a brief overview of the commands and what they do:

- **query** - Selects meta from the meta database based on the query that is passed in, possibly using the index for fast retrieval.
- **values** - Returns groups of unique meta values sorted by some criteria. It is optimized to return a subset of the unique values sorted by an aggregate function such as count.
- **msearch** - Takes text search terms as it's input, and returns matching sessions that match the search terms. It can search within indexes, meta, raw packets, or raw logs.
- **packets** - Returns raw packets or log data based on a time range, session ID list or where clause. **summary** - Returns
- name=value pairs of information regarding the active databases of the service. **timeline** - Returns session counts for a
- time period. Usually used for charting sessions over time.
- **deviceId** - Converts a session ID to a equivalent session on a remote service (e.g., which device and session ID was this aggregated from). **content** - Convert
- raw packets data to some other consumable format. Typically used to extract files out of well known protocols like HTTP or SMTP/POP. **aliases** - Returns the textual representations of values that are normally integer based (e.g., service 80 is HTTP).
- **language** - Returns the language schema of this service's index.
- **keyrefs** - Returns list of meta entities and renamed keys. **validate** - Validate a query or values call without execution. **pin** - Pins
- one or more sessions for long term storage outside of the normal session/meta/packet rollover mechanics.
-
-

/sdk content

The content call is used to retrieve a Session sored on disk and return it as one of many diferent representations. Or it may jus return the Session as the raw data, which would include all the metadata and the raw packets or log. This command supports a wide range of representations (hex, packets, web, VOIP, email, meta, text, etc.), but many of those representations are only useful for the Invesigator thick client. Nowadays, mos third party API access revolves around extracting one or more fles out of a specifc protocol like HTTP.

Parameters:

- **session** - The session ID that will be retrieved **packet** - The packet ID of the first packet of the session. This should never be used by the caller
- as it's for internal API use only. **pinId** - The pinId of the session in long term cache. If provided, session cannot be used.
- **maxSize** - The max number of bytes to return, zero means no limit. This parameter is used to control the maximum bytes that a large network session should
- return and is mainly meant to prevent an extraordinary large network session from consuming a large number of resources during the transfer. Be careful setting this parameter to zero.
  failOnCacheMiss - If true, the content call fails if the NWD is not in the cache. This is used by optimized caching code only and is not normally used by the
- caller.
  renderType - The render type (see below). Not specifying this parameter always results in an NWD and renderFlags and renderOptions are not considered.
- **renderFlags** - Bitwise mask to control option (see below). **renderOptions** - An encoded string params containing additional options. For internal use only.

- Render Types
-

C Enum	Value	String Equivalent	Description
NW_CONTENT_TYPE_AUTO	0	apptype	Auto Select HTML Content View
NW_CONTENT_TYPE_DETAILS	1	meta	HTML Meta Details View
NW_CONTENT_TYPE_TEXT	2	text	HTML Text View
NW_CONTENT_TYPE_HEX	3	hex	HTML Hex View
NW_CONTENT_TYPE_PACKETS	4	packets	HTML Packet View
NW_CONTENT_TYPE_MAIL	5	mail	HTML Mail View
NW_CONTENT_TYPE_WEB	6	web	HTML Web Page View
NW_CONTENT_TYPE_VOIP	7	voip	HTML VOIP View
NW_CONTENT_TYPE_IM	8	im	HTML IM View
NW_CONTENT_TYPE_FILES	9	filelist	HTML Listing of all files found in session
NW_CONTENT_TYPE_PCAP	100	pcap	Pcap Packet File

NW_CONTENT_TYPE_RAW	102	raw	Raw Content
NW_CONTENT_TYPE_XML	103		Meta XML File
NW_CONTENT_TYPE_CSV	104		Meta Comma Separated File
NW_CONTENT_TYPE_TXT	105		Meta Tab Separated File
NW_CONTENT_TYPE_NWD	106	nwd	NetWitness Data File
NW_CONTENT_TYPE_FILE_EXTRACTOR	107	files	Extract files from common protocols
NW_CONTENT_TYPE_LOGS	108	logs	Log extract (captured logs, LF delimited)
NW_CONTENT_TYPE_PROTOBUF	109		Content of the NWD as a Google Protocol Buffer object

NetWitness Content Render Flags

C Enum	Value	Comment
NW_CONTENT_FLAG_STREAM1	0x00001	Return only request stream content.
NW_CONTENT_FLAG_STREAM2	0x00002	Return only response stream content.
NW_CONTENT_FLAG_SINGLE_COLUMN	0x00004	Format generated web page as a single column with requests and responses interleaved.
NW_CONTENT_FLAG_PACKET_PAYLOAD	0x00008	Include only session payload.
NW_CONTENT_FLAG_DECODEAS_EBCDIC	0x00010	Convert session payload from EBCDIC to ASCII.
NW_CONTENT_FLAG_DO_NOT_EMBED	0x00020	Do not embed application/audio/video traffic into the generated web page.
NW_CONTENT_FLAG_UNCOMPRESS_TEXT	0x00040	Unzip web content in text view.
NW_CONTENT_FLAG_DECODE_SSL	0x00080	Attempt to decrypt SSL sessions if the encryption key is provided.
NW_CONTENT_FLAG_STRIP_STYLE_TAGS	0x00100	Removes all html style tags from the original html document.
NW_CONTENT_FLAG_IGNORE_CACHE	0x01000	Ignore any content in cache and requery, affects only request.
NW_CONTENT_FLAG_NO_EMBEDDED_EXE	0x02000	Do not look for or extract hidden/embedded PE files when performing file extraction.
Protobuf only flags		
NW_CONTENT_FLAG_INCLUDE_DUPS	0x04000	Include packets otherwise removed by assembly
NW_CONTENT_FLAG_INCLUDE_HEADERS	0x08000	Include packet header meta information
NW_CONTENT_FLAG_CAPTURE_ORDER	0x10000	Do not assemble packets, return in capture order

RESTful API

The content command is exposed over the RESTful interface by appending /content to the path of the /sdk node (normally jus /sdk/content, but would be diferent for Workbench collections). The RESTful API converts the native content messages that are normally sent over the wire into one of several supported output formats that will be returned over HTTP. The primary use case of /sdk/content involves extracting fles from well known protocols. For insance, it is possible to extract all fles transfered over HTTP, SMTP, POP3 or SMB, assuming the full session was captured. When requesing fles over the RESTful interface, the fles will be returned as multipart/mixed.

Example usage for extracting fles from a session with a supported protocol:

```
/sdk/content?session=12345&maxSize=0&render=files&expiry=0
```

> The *expiry* parameter is the server side RESTful timeout (in seconds). Setting it to zero means do not timeout.

/sdk hierarch

The `hierarch` call returns information about the hierarchy of devices attached to the collection represented in this database.

A hierarchy consiss of this device, plus any devices that this device is connected to. For each device, the contents of the `/sys/stats` folder is returned. This information includes the device name, it's UUID, and it's version information.

The hierarchy command returns it's information as a MessagePack object, which may be translated into diferent representations depending on what API you are using to access the Core service. For example, using the REST API it is translated to a JSON object.

For devices that connect to upsream devices, such as a Broker or Concentrator, the hierarchy message will contain a `devices` member. The `devices` member is an array that holds the contents of the `hierarch` message as executed on each upsream device. In this way, the `hierarch` message forms a hierarchical directory of all services that the device connects to, both directly and indirectly.

/sdk keyrefs

Returns lis of meta entities and renamed keys. This messages does not accept any parameters.

1. Meta Entities. These are defined as entity elements in the index configuration.
2. Renamed Keys. These are defined as keys that contain a rename element.

The returned lis has two types of entries:

Renamed keys can be disinguished from entities in the result set because renamed keys also include the key itself in the lis of referenced keys.

/sdk language

Returns the current meta database schema as well as the level of indexing for each key as defned by the current confguration on the service. Each result returned describes a single meta key.

Parameters:

- **id1** - The starting language id. This is optional and is usually left off or set to zero. The only time this parameter should be used is if the full language is not retrieved in a single call and the next batch must be retrieved. In this case, you would set *id1* to the next valid ID to be retrieved based on the last valid ID retrieved from the previous call.
- **id2** - The ending language id. Normally not used, but see remarks for *id1*.
- **time1** - An optional starting date and time for retrieving language counts. Only valid when *flags* is set to *quick-count* or *full-count*. **time2** - An
- optional ending date/time for retrieving language counts.
- **options** - Extra optional arguments, currently unused.
- **flags** - Optional flags to configure how the results are returned. Can be a number (bitwise mask as defined in the C API) or comma separated values like *default*, *no-count*, *quick-count* or *full-count*.
- **size** - The max number of entries to return. In most cases, you will want to set this number high (e.g., 1000) to retrieve the full language with one call.

C API

Each NwField result will describe a single language element as follows:

- NwField::type will contain the name (accessible through the @p name member)
- NwField::variant will contain an nwText value with a description
- NwField::flags will specify the index level and other information as defined by @ref NwLanguageResultFlags. Please use the provided masks to extract the required information from the uint32.

```
After calling this function, you can optionally call NwResultsInfo to check for additional information about
the query results.
```

Parameters:

- **collectionHandle** - Handle of the collection. **lang1** - The first language id,
- usually pass zero. **lang2** - The second language id, usually pass zero. **time1** -
- The starting time in seconds since 1970. **time2** - The ending time in seconds
- since 1970. **reserved** - Reserved for future use, pass NULL. **flags** - Additional
- NwLanguageFlags to control the data returned.
- **result** - NwField array, large enough to hold size results. **resultSize** - The size of
- the results array, returns size of results.
-
- Language Flags

C Enum Flag	Value	Comment
NW_LANGUAGE_FLAG_DEFAULT	0	No value counting is done. Time1 and time2 parameters are ignored.
NW_LANGUAGE_FLAG_NO_COUNT	1	Same as default.
NW_LANGUAGE_FLAG_QUICK_COUNT	2	Returns the count of values from the most recent non-empty time slice only. Slices of index between time1 and time2 are considered.
NW_LANGUAGE_FLAG_FULL_COUNT	4	Scan the time slices between time1 and time2 and return the unique value counts between those times in the results.

NetWitness Language Result Flags

```
These flags and masks are used to decipher the results returned from the NwLanguage function.
Specifically, the nwuint32 flags member in NwField.
```

C Enum Flag	Value	Comment
NW_LANGUAGE_KEY_INDEX_MASK	0x000F	Mask for the index level.
NW_LANGUAGE_KEY_INDEX_FILTER	0	Indicates the key is filtered, so data related to the key should be ignored.

NW_LANGUAGE_KEY_INDEX_NONE	1	Indicates the data is kept but not indexed
NW_LANGUAGE_KEY_INDEX_KEYS	2	Indicates the data is indexed only at the key level.
NW_LANGUAGE_KEY_INDEX_VALUES	3	Indicates the data is indexed at key and value levels.
NW_LANGUAGE_KEY_SPECIAL_MASK	0x00F0	Special flags about each language key
NW_LANGUAGE_KEY_SPECIAL_SINGLETON	0x0010	Token appears only once per session
NW_LANGUAGE_KEY_SPECIAL_VALUE_ALIASES	0x0040	Indicates this key has value aliases
NW_LANGUAGE_KEY_SPECIAL_TRANSFORM	0x0080	Indicates this key has a transform
NW_LANGUAGE_KEY_ACTION_MASK	0x0F00	Mask indicating default behavior in Investigator
NW_LANGUAGE_KEY_ACTION_HIDDEN	0x0100	Indicates the default is to hide the key, user configurable from index-*.xml file.
NW_LANGUAGE_KEY_ACTION_OPEN	0x0200	Indicates the default is to open the key, user configurable from index-*.xml file.
NW_LANGUAGE_KEY_ACTION_CLOSE	0x0300	Indicates the default is to close the key, user configurable from index-*.xml file.
NW_LANGUAGE_KEY_ACTION_AUTO	0x0400	Indicates the default is to auto open the key, user configurable from index-*.xml file.
NW_LANGUAGE_KEY_PROTECTED_VALUE	0x10000	Values of this type should be treated as protected.
NW_LANGUAGE_KEY_PROTECTED_TOKEN	0x20000	Indicates that the key is intended to contain tokenized values.
NW_LANGUAGE_KEY_TRANSIENT	0x800000	Indicates that values for this key are not persisted and only exist during parsing on a decoder.
NW_LANGUAGE_KEY_RESULT	0x80000000	Indicates this result is a response to a language call

Returns:

- 1 on success, 0 on failure

/sdk msearch

The index provides a low-level `msearch` function to perform text searches agains all meta types. This type of search does not require users to defne their queries in terms of known meta types. Insead, it searches all parts of the database for matches. `msearch` is used by the Events view text search. See the "Filter and Search Results in the Events View" topic in the *Invesigation and Malware Analysis Guide* for detail on the accepted search forms and examples.

`msearch` parameters:

```
msearch-params  = search-param, {space, where-param}, {space, limit-param}, {space, size-param},
{space, flags-param}; search-param    = "search=", ? free-form search
string ? ; where-param     = "where=", ? optional where clause ? ;
limit-param     = "limit=", ? optional session scan limit ? ; size-
param     = "size=", ? optional result count limit ? ; flags
= "flags=", {msearch-flag, {"," msearch-flag} }; msearch-flag    =
"sp" | "sm" | "si" | "ci" | "regex" ;
```

The `msearch` algorithm works as follows:

1. A set of sessions is identified from the index by finding the intersection of three sets:
   - (Set 1) All sessions in the database
   - (Set 2) Sessions that match the `where` clause parameter
   - (Set 3) If the `si` flag is specified, sessions that indexed values that match the search string parameter.

2. If the search specifies the `sm` parameter, all meta items from the set of sessions identified in step 1 are read and scanned to see if they match the search string parameter. The meta items will be read from the service nearest to the point where the search was executed. For example, if the search is performed on a Broker, the meta items may be read from the Concentrator nearest to the broker, but if the search is performed on an Archiver the meta items will be read from the Archiver itself.
3. If the search specifies the `sp` parameter, all raw packet or log entries from the set of sessions identified in step 1 are read and scanned to see if they match the search string parameter. The packets will be read from the service nearest to the point where the search was executed. For example, if the search is performed on a Concentrator, the packet data will be read from the Decoder, but if the search is performed on an Archiver, the packet data will be read from the Archiver itself.
4. Matches from step 2 and step 3 are returned as they are found, up to the point where the `limit` parameter is reached or the `size` count is reached, whichever occurs first. The `limit` parameter specifies the maximum number of sessions for which meta and packet data will be scanned. If `limit` is not specified, the entire set of sessions determined in step 1 is scanned. The `size` parameter specifies the maximum number of results that will be returned. In practice, the `size` parameter acts more as a suggestion. It is possible that slightly more results than specified will be returned, but fewer results will never be returned. If the `size` parameter is not specified, all results matching the search will be returned.

`msearch` Flags

Flag	Description
`sp`	Scans raw packet data
`sm`	Scans all meta data
`si`	Does index lookups for all search parameters before scanning meta
`ci`	Performs a case insensitive search. Returned results are case-preserving.
`regex`	Treats the search parameter as a regular expression. Only a single regular expression can be specified, but the regular expression may be arbitrarily complex.

`msearch` Index Search Mode

Using the index search mode, specifed by using the `si` fag, causes results to be returned signifcantly faser than any other mode. The main limitation of this mode is that it only returns matches on text terms that match value-indexed meta values.

- The `si` parameter must be combined with the `sm` flag. The `si` parameter implies the search only matches indexed meta.
- The `si` parameter can be used with regex searches, however only text indexed values will match. IP addresses and numbers will not match the regex.

Text Search Syntax

The search parameter given to `msearch` is composed of 1 or more words, separated by whitespace. For example, searching for `foo` returns sessions that contain the word foo.

If multiple terms are provided for the search, they are implicitly considered to be an AND operation. For example, searching for `foo bar` returns sessions that contain both foo AND bar. Sessions that contain only foo or only bar are fltered out. If you want to search for sessions containing any of two or more terms, you mus explicitly separate the terms with the word OR. For example, searching for `foo OR bar` returns sessions that contain either foo or bar.

Search Syntax And Index Modes

The searches given to the `msearch` command are interpreted according to the index level on all the indexes. `msearch` works on the valueindexed keys in the index. Search terms provided to `msearch` will fnd values that are an exact match to values that were indexed.

As of version 11.1, there are new index modes available that allow `msearch` to locate text that is not an exact match to the search input. `msearch` supports wildcard searches on the word meta index, if the word meta index has the ngram option enabled. For details on the ngram option, see the topic Index Cusomization.

The wildcard search allows the use of the `*` and `?` characters as wildcards in search terms. The `*` can sand for 0 or more characters, while the `?` may sand for any single character. To search for those characters in an N-gram enabled index, you may escape them with a backslash character.

If the word index has the 'edge' N-gram option enabled, then it can be used to locate searches that end in a wildcard. This means it is only useful for fnding text that begins with a known prefx.

If the word index has the 'all' N-gram option enabled, then wildcards may appear anywhere in the search term.

This table summarizes the relationship between word index level, and the types of searches that `msearch` will locate.

Search input	Non-indexed	IndexValues	IndexValues with Edge N-grams	IndexValues with All N-grams
"foo"	no match	"foo"	words starting with "foo"	words containing "foo"
"foo*"	no match	literal "foo*"	words starting with "foo"	words starting with "foo"
"*foo"	no match	literal "*foo"	no match	words ending with "foo"
"*foo*"	no match	literal "*foo*"	words staring with foo	words containing "foo"
"foo*"	no match	literal "foo*"	literal "foo*"	literal "foo*"

`msearch` Tips

- Always use the `where` clause to specify a time range for the search.
- To search for IP address ranges, specify them in the `where` clause.
- Use the `limit` parameter when not using the index search mode. Without it, there will be an extremely large amount of data read by the meta and packet databases.

/sdk packets

The packets call returns raw packets or logs based on the user's selection criteria. This command corresponds to the C SDK command NwPackets() and the RESTful URL at /sdk/packets. Both APIs operate by calling the "/sdk packets" command on the server to do all the heavy lifting.

NwConsole also supports saving pcap or log fles via the internal packets command. For more information, type "help packets" while running NwConsole. Detailed help on the command can be found by looking at the manual page ("man packets").

RESTful API

The packets command is exposed over the RESTful interface by appending /packets to the path of the /sdk node (normally jus /sdk/packets, but would be diferent for Workbench collections). The RESTful API converts the native packet messages that are normally sent over the wire into one of several supported output formats that will be returned over HTTP.

The following are the arguments used to select the session's packets/logs to be returned:

- **sessions** - A CSV list of Session IDs or Session ID ranges. Example: sessions=1,3,5-10,12-15,17 **where** - An optional where clause that will be used to match
- which session's packets or logs will be returned **time1** - The starting time (either POSIX time or "YYYY-MM-DD HH:MM:SS" format). Cannot be used with
- the parameters *sessions* and *where*. **time2** - The ending time (either POSIX time or "YYYY-MM-DD HH:MM:SS" format). Cannot be used with the parameters
- *sessions* and *where*.

> The *time1* and *time2* parameters are extremely fast but they do not filter any raw data within the specified time range.

The following are the supported output formats, which are selected by passing the *render* parameter with one of the following values:

- **pcap** - Returns the packets in a pcap file (log sessions are ignored). Content-Type: application/octet-stream **logs** - Returns raw logs in a plain text file
- (network sessions are ignored). Content-Type: text/plain **text/csv** - Return a line delimited CSV file containing only logs (network sessions are ignored). Each
- entry/line consists of 5 fields: timestamp, source, forwarder, lccid, log. The last field (log) is the actual log text and is not escaped, any comma's that occur after the 4th delimiter are part of the log text. Content-Type: text/plain **text/xml** - Return an XML file containing only logs (network sessions are ignored). Each log entry will include timestamp and optionally source, forwarder and lccid if known. The log text will be contained in a CDATA, and split across consecutive
- CDATA if the logs contains a CDATA terminator (]]>). Content-Type: text/xml **application/json** - Return a JSON file containing only logs (network sessions are ignored). Each log entry will include timestamp, source, forwarder, lccid and log text. Content-Type: application/json Example:

```
Grab 5 minutes worth of packets: time1="2015-09-12 15:00:00" time2="2015-09-12 15:05:00"
/sdk/packets?expiry=0&time1=2015-09-15%2016%3A00%3A00&time2=2015-09-15%2016%3A05%3A00
```

> The *expiry* parameter is the server side RESTful timeout (in seconds). Setting it to zero means do not timeout.

C API

```
nwuint32 NwPackets(  nwuint32
collectionHandle,  const char*
parameters,  nwuint32 flags,
 NwPacketsHandler packetsHandler,  void*
userData);
```

Streams packets from a local or remote collection based on a lis of session IDs or a time range. The packets are retrieved from the passed in NwPacketsHandler (which cannot be NULL). The handler is called once for every packet. userData can be anything, usually a pointer to a sructure, and is passed verbatim to the packets handler.

Examples of the parameters sring:

```
Session Range w/Where Clause
 sessions=1,5,8,11-15,21-30,45-23221 where="service=80 && time='2015-03-19 00:00:00'-'2015-03-20
00:00:00'"

Time Range
 time1="2010-2-13 13:00:00" time2="2010-2-13 13:05:00"
```

This call can be canceled using NwCancel() from another thread or from the packets handler.

> NOTE: The time range is only guaranteed to work correctly if the collection has not imported any packets out of order.

Parameters:

- **collectionHandle** - Handle of the collection.
- **parameters** - A string containing either a comma separated list of session IDs and session ranges, a where clause or a beginning and ending time range. The time range (*time1*/*time2*) is mutually exclusive with *sessions* and/or *where*. **flags** - Additional flag specifiers. Requesting session ids is only possible when
- passing the "sessions" parameter. **packetsHandler** - The callback function, called once for each packet returned. **userData** - User defined pointer returned in
- the progress handler.
-

Returns 1 on success, 0 on failure

/sdk pin

The idea behind the `pin` command is to sore sessions (with meta and packets) for long term retention, outside of the normal rollover mechanics. For insance, if several sessions indicate there might be a attack on a cusomer's network, those sessions could be pinned (perhaps by NetWitness Respond) so they can be viewed at a later time without worry that the meta or packets might be rolled out before the threat is accessed.

There are two parameters that control how a session is pinned:

- **session** - The session ID that should be pinned
- **sessions** - One or more comma separated session IDs or session ranges that should be pinned. You can use either parameter interchangeably, but they cannot both be submitted at the same time. Example: "sessions=1,5,8-11", this will pin session IDs 1, 5, 8, 9, 10 and 11.

The response from pinning one or more sessions is called the Pin ID. For every session pinned, you will receive back a response that contains the original session ID, the Pin ID and the total size (bytes) of the pinned session in long term sorage. When pinning multiple sessions, you will get back multiple responses, each response will be for a single session.

The Pin ID should be sored for later retrieval. For insance, Respond will sore the Pin IDs for each incident in which the cusomer requesed the sessions to be pinned. Later on, you can view those pinned sessions via a content call. See the /sdk content call for more details. Pinned sessions can only be viewed via a content call. They will not show up in a query or other SDK command once the sessions have rolled out.

Miscellaneous Operations

There is one additional parameter `op` than can be used for some additional operations around pinned sessions. The following are the valid values for the `op` parameter:

- **ls** - Returns a listing of all pinned sessions for all online services. The returned listing will contain the Pin ID, the creation date of the pinned session and the storage size of the pinned session (in bytes).
- **validate** - Validates that one or more Pin IDs are still valid and in Pin storage. The response to each Pin ID (see the `pinId` parameter below) passed in will either indicate that the Pin ID is valid, not-found or if the no Core service recognizes the Pin ID, then no response will be given for those Pin IDs. If no response is received for a specific Pin ID, you can only assume that the service that stores those Pin IDs is offline, is no longer operational or that the original service for the Pin ID is no longer configured to store pinned sessions.
- **unpin** - This parameter is used to delete previously pinned sessions. Like `validate` above, it requires one or more Pin IDs to be passed in. The responses received are also nearly identical to `validate` , with the exception that instead of indicating the Pin ID is valid, it indicates the Pin ID was deleted.

pinId

This parameter accepts one or more comma separated Pin IDs. It mus be provided when `op=validate` or `op=unpin` is used.

/sdk query

The `query` message has the following syntax:

```
query-params     = size-param, space, query-param, {space, start-meta-param}, {space, end-metaparam},
{space, search-param}, {space, streamlimit-param}; size-param       = "size=", ? integer between 0 and
1,677,721 ? ; query-param      = "query=", query-string ; start-meta-param = "id1=", metaid ; end-meta-
```

```
param   = "id2=", metaid ; search-param     = "search=", search-string ; streamlimit-param =
"streamLimit=", streamSize ; metaid           = ? any meta ID from the meta database ? ;
```

The `id1` , `id2` , and `size` parameters form a paging mechanism for returning a large number of results from the database. Their usage mosly benefts developers who are writing applications directly agains the NetWitness Core database. Normally, results are returned in the order of oldes to newes data (higher meta IDs are always more recent). In order to return results from mos recent to oldes, reverse the IDs such that `id1` is larger than `id2` . This has a slight performance penalty, because the where clause mus be completely evaluated before processing in reverse order can begin.

When size is left of or set to zero, the sysem sreams back all results without paging. For the RESTful interface, this results in the full response to be returned with chunked-encoding. The native protocol returns the results over multiple messages.

The `streamLimit` parameter to the query specifes how many groups of results will be returned when the query is operating in sreaming mode. For example, the query may be in sreaming mode by virtue of not having a 'size' parameter, but the sreamLimit can be used to limit how many results are actually returned. Specifying the `streamLimit` implies that sreaming mode is requesed. If the client requess a 'sreamLimit' but the query cannot be processed in sreaming mode then an error is generated.

The `query` parameter is a `query` command sring with its own NetWitness-specifc syntax:

```
query-string      = select-clause {, where-clause} {, group-by-clause {, order-by-clause } } ; select-
clause     = "select ",  ( "*" | meta-or-aggregate {, meta-or-aggregate} ) ; where-clause      = " where
",  { where-criteria } ; meta-or-entity    = (meta_key | entity) ;
meta-or-aggregate = meta-or-entity | aggregate_func, "(", meta-or-entity, ")" ;
aggregate_func    = "sum" | "count" | "min" | "max" | "avg" | "distinct" | "first" | "last" | "len" |
"avglen" | "countdistinct" ; group-by-clause   = " group by
", meta-key-list meta-key-list     = meta-or-entity {, meta-
key-list} order-by-clause   = " order by ", order-by-column
order-by-column   = meta-or-aggregate { "asc" | "desc" } {, order-by-column}
```

The `select` clause allows you to specify either `*` to return all the meta in all the sessions that match the where clause, or a set of meta feld names and aggregate functions to select a subset of the meta with each session.

The `select` clause may contain entity names in the place of meta key names. If an entity name is in the select clause, meta items returned by the query will have their key name set to the entity name, rather than their actual meta key name sored in the session. Thus, the names of the meta items returned in the query will match the names of the metas in the select clause. For example, if there is an entity `ip` that consiss of `ip.dst` and `ip.src` , then a query containing `select ip` will only return `ip` felds, with nothing to disiguish `ip.dst` meta items from `ip.src` meta items in the result set.

The `select` clause may contain renamed meta key names. Any felds appearing in the result set as a result of a renamed key in the `select` clause will be returned with the meta key name matching the name used in the `select` clause. For example, if the key `port_src` is used to rename `tcp.srcport` , then a query containing `select port_src` will only return `port_src` felds, even if the underlying meta had type `tcp.srcport` .

Note: Usage of renamed meta key pairs in the `select` clause cannot be combined with fxed-size result paging for a query. Doing so causes discrepancies in the results returned to Brokers. The reason for the discrepancies is that Concentrators cannot return only one of the key values of a renamed meta key pair and sill preserve the correctness of the result set for the requesed size. Hence, the Concentrator omits renamed meta key pair results to preserve the correctness of the result set, which causes the Broker to pull the result from the next Concentrator and advance the IDs that are returned.

Example: `select ip.proto,ipv6.proto` cannot be combined with `size=10` (a paging query) `size=10 flags=0 threshold=0 query="select time,ip.src,ip.dst, ip.proto,ipv6.proto,eth.type,size,payload,lifetime,client,did`

The aggregate functions have the following efect on the query result set.

Function	Result
`sum`	Add all meta values together; only works on numbers
`count`	The total number of meta fields that would have been returned
`min`	The minimum value seen

`max`	The maximum value seen
`avg`	The average value for the number
`distinct`	Returns a list of all unique values seen
`countdistinct`	Returns the number of unique values seen. `countdistinct` is equivalent to the number of metas that would have been returned by the distinct function.
`first`	Returns the first value seen
`last`	Returns the last value seen
`len`	Converts all field values to a UInt32 length instead of returning the actual value. This length is the number of bytes to store the actual value, not the length of the structure stored in the meta database. For example, the word "NetWitness" returns a length of 10. All IPv4 fields, like `ip.src`, return 4 bytes.
`avglen`	Returns a single value which is the average value returned from the `len` function. The result is always a float64 value.

`where` Clauses

The `where` clause is a flter specifcation that allows you to select sessions out of the collection by using the index.

Syntax:

```
where-criteria    = criteria-or-group, { space, logical-op, space, criteria-or-group } ; criteria-or-group =
criteria | group ;
criteria          = (meta-key | entity), ( unary-op | binary-op meta-value-ranges ) ; group             =
["~"], "(" where-clause ")" ; logical-op        = "&&" | "||" ; unary-op          = "exists" | "!exists" ;
binary-op         = "=" | "!=" | "<" | ">" | ">=" | "<=" | "begins" | "contains" | "ends" | "regex" ; meta-
value-ranges = meta-value-range, { ",", meta-value-range } ; meta-value-range  = (meta-value | "l" ), [ "-", (
meta-value | "u" ) ] ; meta-value        = number | quoted-value | ip-address | mac-address | relative-time ;
number            = ? any numeric value ? | ( '"' text '"' ) quoted-value      = ( '"' text '"' ) | ( '"' date-
time '"' ) ; relative-time     = "rtp(" , time-boundary , "," ,  positive-integer , time-unit, ")" ; time-
boundary     = "earliest" | "latest" | "now" ; positive-integer  = ? any non-negative integral number ?
time-unit         = "s" | "m" | "h" ;
```

When specifying rule criteria, the `meta-value` part of the clause is expected to match the type of the meta specifed by the `meta-key` . For example, if the key is `ip.src` the `meta-value` should be an IPv4 address. Entity names are allowed in any location where a meta-key name is required.

Queries using a `meta-key` name will match meta items corresponding both to the `meta-key` name as well as to the names of any "renames" specifed for the key. See "Key Renaming" under the Index Cusomization topic for details on key renaming.

Query Operators

The following table describes the function of each operator.

Operator	Function
`=`	Match sessions containing the meta value exactly. If a range of values is specified, any of the values is considered a match.
`!=`	Matches all sessions that would not match the same clause as if it were written with the `=` operator.
`<`	For numeric values, matches sessions containing meta with the numeric value less than the right side. If the right side is a range, the first value in the range is considered. If multiple ranges are specified, the behavior is undefined. For text metas, a lexicographical comparison is performed.

<=	Same behavior as <, but sessions containing meta that equals the value exactly are also considered matches.
>	Similar to the < operator, but matches sessions where the numeric value is greater than the right side. If the right side is a range, the last value in the range is considered for the comparison.
>=	Same behavior as >, but sessions containing meta that equals the value exactly are also considered matches.
begins	Matches sessions that contain text meta value that starts with the same characters as the right side.
ends	Matches sessions that contain text meta that ends with the same characters as the right side.
contains	Matches sessions that contain text meta that contains the substring given on the right side.
regex	Matches sessions that contain text meta that matches the regex given on the right side. The regex parsing is handled by boost::regex.
exists	Matches sessions that contain any meta value with the given meta key.
!exists	Matches sessions that do not contain any meta value with the given meta key.
length	Matches sessions that contain text meta values of a certain length. The expression on the right side must be a nonnegative number.

Text Values

The sysem expects quoted text values. Unless it can be parsed as a time (see below), a quoted value is interpreted as text.

> It is also important to quote any text value that may contain - so that it is not interpreted as a range.

For text values, the backslash character `\` is used as an escape value. This character is used when you need to search for a value containing quote characters. If you need to search for a backslash character, then the backslash itself mus be escaped, as `\\` . Note that if you are wrapping the query parameters within another language, such as the parameter felds of the REST interface, you may need to add additional escape levels as required by whatever API or interface you are using to interact with the core service.

IP Addresses

IP addresses can be expressed using sandard text representations for IPv4 and IPv6 addresses. In addition, the query can use CIDR notation to express a range of addresses. If CIDR notation is used, it is expanded to the equivalent value range.

MAC Addresses

A MAC address can be specifed using sandard MAC address notation: `aa:bb:cc:dd:ee:ff`

Numeric Values

In a where clause, you can specify numeric search values. Numbers should not be surrounded by quotes.

Bucketed Numeric Indexes

Meta keys indexed with bucketing can be used like any other numeric search value. Under mos situations such searches will return sessions that have a meta value that exactly matches the requesed search criteria.

Special behavior is invoked for queries that select only `sessionid` , for example a query of the form `select sessionid where size = 2048` . Selecting `sessionid` explicitly bypasses all meta database read operations, and only returns index information. If selecting `sessionid` only, and if the numeric value specifed is exactly equal to one of the bucket values, then the sysem will return all sessions that match somewhere in the bucket, rather than an exact match. For example, the search term `size = 2048` will match all sessions in the 2 KB bucket, which is the range from 2048 to 3171 bytes. However, if the search values does not match a bucket values, then the sysem will return only matches for the exact byte value. For example, the search term `size = 2049` will only match sessions with a size meta value exactly 2049. In this mode of operation, specifying a non-bucket value in a where clause is slower than searching within a bucket value. The 'where' clause parameter to the `values` API also invokes this optimization.

Using bucketed values in other forms of `query` does not invoke special behavior. The same is true for the `msearch` API. For those APIs, the use of a bucketed index in the where clause is evaluated accurately, without special meaning applied to bucket values. To search within an entire bucket using these APIs, specify the bucket range explicitly. For example `size=2048-3171` .

More information on how to tell if an index is bucketing is in the topic Index Cusomization.

Numeric Value Aliases

For numeric values, aliases specifed in the index can be used in a query as a quoted sring in place of where a literal numeric value would be used; e.g.,

```
select * where service = "NFS"
```

Numeric value aliases can be used anywhere a numeric literal might be used: as a single value, as the beginning or end of a range, or in a comma-delimited lis of values (and/or ranges).

Refer to the topic Index Cusomization for details of how value aliases can be specifed in the index.

Date and Time Expressions

In NetWitness Platform, dates are represented using Unix epoch time, which is the number of seconds since Jan 1, 1970 UTC. In queries, you can express the time as this number of seconds, or you can use the sring representation. The sring representation for the date and time is `"YYYY-mmm-DD HH:MM:SS"` . A three-letter abbreviation represents the month. You can also express the Month as a two-digit number, 0112.

Time values mus be quoted.

All times specifed in queries are expected to be in UTC.

Relative Time Points

Relative time points allow a where clause to reference a value at some fxed ofset relative to the earlies or lates time metas seen in the collection. It can also be used to reference a point in time relative the the current time.

A relative time point expression has the syntax `rtp(boundary, duration)` .

The boundary is either `earliest` , `latest` , or `now` .

The duration is an expression of hours, minutes, or seconds. For example, `24h` , `60m` , or `60s` . When the boundary is `earliest` , the duration represents the amount of time **after** the earlies time present in the collection. If the boundary is `latest` , the duration represents the amount of time **before** the earlies time present in the collection. If the boundary is `now` , the duration represents the amount **before** the current time.

When the boundary is `now` , the sysem clock of the Core service hos is used to determine what time it is.

Boundary can be represented as 0 seconds if you wish to specify the relative time point with no duration ofset. This is mos useful in the case of the `now` boundary, since it is possible that the highes, lates, time observed in the collection may be much earlier than the current time.

Relative time points can only be used in SDK operations, where there is a collection from which to get the boundaries for earlies and lates time metas.

Relative time points only work on indexed meta types. The default indexed meta types are `time` and `event.time` .

Examples:

```
Last 90m of collection time: time =
rtp(latest, 90m) - u

First 2 days of event time:
event.time = l - rtp(earliest, 48h)
Events added in the last hour: time =
rtp(now, 60m) - rtp(now,0s)
```

Special Range Values

Ranges are normally expressed with the syntax *`smallest` * - * `largest` *, but there are some special placeholder values you can use in range expressions. You can use the letter `l` to represent the lower-bound of the all meta values as the sart of the range, and `u` to represent the upper bound. The bounds are determined by looking at the smalles or larges meta value found in the index out of all the meta values that have already entered the index.

If you use the `l` or `u` tag, it should be unquoted.

For example, the expression `time = "2014-may-20 11:57:00" - u` would match all time from that 2014-may-20 11:57:00 to the mos recent time found in the collection.

Notice that it is easy to confuse a range expression with a text sring. Make sure that text values that contain - are quoted, and that hyphens within range expressions are not within quoted text.

`group by` Clause (since 10.5)

The query API has the ability to generate aggregate groups from the results of a query call. This is done using a `group by` clause on the query. When `group by` is specifed, the result set for the query is subdivided into groups. Each group of results is uniquely identifed by the meta values indicated in the group by clause.

For example, consider the query `select count(ip.dst)` . This query returns a count of all ip.ds metas in the database. However, if you add a `group by` clause, like this: `select count(ip.dst) group by ip.src` , the query returns a count of the ip.ds metas found for each unique ip.src.

As of version 10.5, you can utilize up to 6 meta felds in a `group by` clause.

The `group by` clause shares some of the same functionality as the `values` call, but it ofers signifcantly more advanced groups at the expense of longer query times. Producing the results of a grouped query involves reading the meta from the meta database for all sessions that match the `where` clause, while a values call can produce its aggregates by reading the index only.

The contents of each group returned by the query are defned by the `select` clause. The `select` clause can contain any of the aggregate functions or meta felds selected. If multiple aggregates are selected, the result of the aggregate function is defned for each group. If nonaggregate felds are selected, the meta felds are returned in batches for each group.

The result set of a `group by` query is encoded with the following rules:

1. All meta items associated with a group are delivered with the same group number.
2. The first meta items returned to the group identify the group key. For example, if the `group by` clause specifies `group by ip.src` , then the first meta item of each group will be an `ip.src` .
3. The normal, nonaggregate meta items are returned after the `group key` , but they all will have the same group number as the group key metas.
4. The aggregate result meta fields for each group are returned next.
5. All fields within a group are returned together. Different group results will not be interleaved.

If one of the `group by` meta items is missing from one of the sessions matched by the `where` clause, that meta feld is treated as a NULL for the purposes of that group. When the results for that group are returned, the NULL-valued parts of the group key will be omitted from the group's results, since the database has no concept of NULL.

The semantics of a `group by` query difer from a SQL-like database in terms of what meta felds are returned. SQL databases require you to

select the `group by` columns explicitly in the `select` clause if you want them to be returned in the result set. The NetWitness Core database always implicitly returns the group columns frs.

A query with a `group by` clause honors the result set `size` parameter if one is provided. However, due to the nature of the grouping, it puts an additional burden on the caller to page and reform groups if a fxed-size result set is requesed. For this reason, you should not specify an explicit result size when making a `group by` call. By not specifying an explicit size, the entire result set will be delivered as partial results. `group by` clauses allow results to be grouped by an entity defnition.

The following table describes the database honors confguration parameters that limit I/O or memory impact of a group by query.

Parameter	Function
/sdk/config/max.query.groups	This is the limit on how many groups can be held in memory to calculate aggregates. This parameter allows you to limit the overall memory usage of the query.
/sdk/config/max.where.clause.sessions	This is the limit on how many sessions from the where clause can be processed in a query. This parameter allows you to set a limit on the number of sessions that have to be read from the meta and session databases to resolve a query.

order by Clause (since 10.5)

An order by clause can be added to a query that contains a group by clause. The order by clause causes the set of grouped results to be returned in sorted order.

An order by consiss of a set of items to sort by in ascending or descending order. Sorting can be performed on any data feld that will be returned in the result set. This includes meta specifed by the select clause, aggregate function results specifed by the select clause, or group by meta felds.

The order by clause can sort over many columns. There is no limit on the number of order by columns allowed in the query; but a practical limit exiss in that each of the order by columns mus refer to something returned by the select clause or group by clause. The multiple column sort is imposed lexicographically, meaning that if two groups have equal values for the frs column, then they are sorted by the second columns. If they are equal in the second column, they are sorted by the third column, and so on for however many order by columns are provided. Groups that do not contain any of the metas referenced by the order by clause are sorted frs in the result set in the case of an ascending sort, and las in the case of a descending sort.

The NetWitness Core database is unique in that the groups of results returned by a query may each have many values for a selection. For example, it is possible to select all meta items that match a meta type and organize them into groups, and it is possible to use the distinct() function to return groups of disinct meta values.

If an order by clause references one of the felds in the group that has multiple values, the behavior of the returned result set depends on whether the query is processed as a **sreaming** query or a **non-sreaming query**.

Streaming queries are those queries where the results are returned to the client as soon as they are found in the database. This is the common use case for queries where the order by cause refers to an indexed meta key. The index for the meta key is used to determine which sessions to sream back to the client frs. Streaming queries allow for fas server-side sorting of results. They can be used to allow order by clauses on much larger data sets than non-sreaming queries, and they return results much faser than non-sreaming queries. When a sreaming query is sorted, the behavior is as follows:

1. The set of meta values matching the order by clause is fetched from the index, and sorted according to the order by direction (ascending or descending).
2. For each sorted meta value, the sessions containing that value are fetched from the database and returned to the client one value at a time. As those sessions groups are found, they are sent to the client and then discarded to free memory.

Note that the efect this has on sessions that might get fetched for more than one meta disinct value in sep 2. When that happens the session might appear more than once in the result set sream. This is logically valid because the session might have more than one 'correct' position in the sorted result set.

Non-sreaming queries are those queries that cannot be sreamed back to the client and require the Core database to hold the entire result set in memory before determining the fnal sort order. The mos common case for this is if the ordering criteria is an aggregate function. The value generated by the aggregate function has to be calculated before the sort order can be determined, so the results cannot be sreamed back to the client as they are found.

1. Within each group, the fields with multiple matching values are ordered by the ordering clause
2. All the groups are sorted by comparing the first occurrence of the ordered field found within each group

The order by clause is only available in queries that have a group by clause, since groups are required to organize the meta felds into disinct records. If you wish to sort an arbitrary query as if there were no grouping applied, use group by sessionid . This ensures that results are returned in groups of disinct sessions or events.

group by clauses are naturally returned in ascending group key order; but, an order by clause can be used to return groups in a diferent

order.

If an `order by` column does not specify `asc` or `desc` , the default ordering is ascending.

Examples:

```
select countdistinct(ip.dst) GROUP BY ip.src ORDER BY countdistinct(ip.dst) select countdistinct(ip.dst) GROUP BY ip.src ORDER BY countdistinct(ip.dst) desc select countdistinct(ip.dst),sum(size) GROUP BY ip.src ORDER BY sum(size) desc, countdistinct(ip.dst) select sum(size) GROUP BY ip.src, ip.dst ORDER BY ip.dst desc select user.dst,time GROUP BY sessionid ORDER BY user.dst select * GROUP BY sessionid ORDER BY time
```

`search` parameter

The `query` API supports a `search` parameter to perform free text searching. The syntax of the search parameter is identical to the search parameter utilized by the `msearch` API call. Refer to the `msearch` documentation for a description of the search feld syntax.

The `search` parameter acts as an extension of the `where` clause in the `query` parameter. This means that the `query` and `search` parameters work together. Use the `query` parameter to specify the `select` clause, the `group by` clause, or the `order by` clause. Any `where` clause criteria specifed in the `query` parameter are combined with the search flter as if they were joined with an `AND` operation.

Searches through the `query` API are always done agains indexed meta, in case-insensitive mode. It has the same behavior as specifying fags `si,sm,ci` to the `msearch` API.

/sdk search

The *search* command is deprecated in favor of the more powerful *msearch* command.

/sdk session

The *session* command maps a session range on the device to it's corresponding meta range.

Parameters:

- **id1** - The starting session ID. If zero, will scope up to the first valid Session ID on device. **id2** - The
- ending session ID. If zero, will scope up to the last valid Session ID on device.

Returns:

- **id1** - The starting session ID. **id2** - The ending session ID. **field1** -
- The starting meta ID that corresponds to *id1*. **field2** - The ending
- meta ID that corresponds to *id2*.
-

/sdk summary

The summary call returns a lot of encoded information that provides global satisics on the databases. The values are encoded as <name>=" <value>" parameters separated by whitespace. The quotes are optional if no whitespace is in the value.

Defnitions:

- **mid1** = The starting valid meta id **mid2** = The ending valid meta id
- **msize** = The current size of the meta database (bytes) **mmax** = The
- maximum size of the meta database (bytes) **pid1** = The starting valid
- packet id **pid2** = The ending valid packet id
- **psize** = The current size of the packet database (bytes) **pmax** = The maximum size of the
- packet database (bytes) **ptime1** = The time of the first packet in the packet database
- (POSIX time) **ptime2** = The time of the last packet in the packet database (POSIX time)
- **time1** = The first valid time from the index (POSIX time, seconds since 1970) **time2** = The
- last valid time from the index (POSIX time, seconds since 1970) **sid1** = The first valid
- session id **sid2** = The last valid session id
- **ssize** = The current size of the session database (bytes) **smax** = The maximum
- size of the session database (bytes) **stotalsize** = The total size of the summary
- index database (bytes) **isize** = The total size of index in memory (bytes)
-
-
-
-
-

/sdk timeline

Retrieve counts for session/size/packets across a time range, results are given in discrete time intervals of minutes, hours or days. This command only works with meta felds that are of type *TimeT*.

Internally, the *timeline* call performs a *values* call on the passed in feldName (usually "time"). The *values* call returns the counts for the where clause in minute intervals (you can literally get the same data jus using *values*). However, what the *timeline* does do is perform what's known as a "roll up". For reques intervals less than or equal to 24 hours, it jus returns the counts for the minutes within the time range. For intervals more than a day but less than a month, it will roll up the counts into "hours" and return the aggregate counts for each hour within the time range (taking into account the passed in timezone). If the time range is larger than a month, then it will perform the roll up as "days", again taking into account the passed in timezone of the client.

```
Roll Up Reasoning: The main function of this API is for graphing session counts over a time period.  The "roll
up" helps provide a readable graph over large time periods.  Most monitors can't display a graph at the minute
level over more than a couple of days, so the roll up keeps the data transfer reasonable and provides an easy
to read graph.  Should the user zoom in further (less than a day), the graph will then provide greater detail
at the minute level.  If for some reason you don't want the "roll up", it's always possible to make a _values_
call for the same data at the minute level only.

Another feature of the _timeline_ call is an in memory cache of time periods greater than a day.  The cache
really improves performance over large time ranges by taking the load off the index for often requested data
that changes very little.  The cache stores results in hour increments and ages it out after an hour.  It is
possible for the cache to grow stale for data received within the last hour, so it's possible to pass the
"ignore-cache" flag so it hits the index for the latest data.  Just keep in mind that this will result in
slower queries and is definitely *not recommended* when the time range is very large (weeks or months).
```

Parameters:

- **time1** - The starting time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS), if greater than time2, then the whole time range is returned, which is the default behavior is neither **time1** or **time2** is supplied.
- **time2** - The ending time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS). If not provided, the default value is zero (EPOCH).
  timezone - The timezone of the local computer to receive the data, hour offset from GMT [-13 to 13]. If zero, results are returned in UTC.
- **size** - The max number of entries to return **flags** - The flags to use for the timeline. Can be a number (bitwise mask) or comma separated values like sessions,
- size, packets, order-ascending, order-descending, ignore-cache or clear-cache.
- **where** - Optional where clause for filtering the data. Only counts of sessions that match the where clause will be returned.
  bookendHours - The number of hours outside the given time1, time2 which will become part of the cache. Default is 24 hours. A larger number of hours can
- make the current call slower, but can make subsequent calls much faster that fall within this timerange + bookend hours.
- **fieldName** - The meta field to return time values on. The default value is *time*. NOTE: *time* is highly optimized in the index and will typically perform much better than other *TimeT* fields like *event.time*.
-

The time intervals returned are in minutes, hours or days. The interval cannot be specifed by the caller, but is controlled by the amount of data requesed. Any time1/time2 interval of 24 hours or less will always be returned in minute intervals. There isn't an option to return the values in seconds. If you need exact values, then you mus use the *values* or *query* command. Keep in mind that *time* is always rounded down to the minute in the index (for maximum performance), whereas *event.time* is maintained at the second level. However, if you need to retrieve exact *time* values, you can use the *query* command to retrieve them from the Meta DB, which sores them accurately.

Each result returned is made up of the following felds:

- **id1** - The minimum session ID for this time period **id2** - The
- maximum session ID for this time period
- **count** - The count of sessions/size/packets as requested in the *flags* parameter for this time period **format** - Always 32 which is
- date-time **value** - The time period for this result. Will always be set to the start of the minute, hour or day as defined by *type*.
- **type** - Indicates whether the result is a *minute*, *hour* or *day* interval.
- **flags** - Always zero **group** - Always
- zero
-

/sdk validate

Validate a query or values call without execution.

This API is used to do syntax validation on the `query` and `values` call. The two modes of operation are selected using the `op` paramter.

If validating a `values` call, provide the where clause and feldName using their respective parameters.

If validating a `query` call, provide the `query` syntax using the query parameter.

/sdk values

The index provides a low-level `values` function to access the unique meta values that have been sored in the index. This function allows developers to perform more advanced operations on groups of unique meta values.

The `values` call parameter syntax:

```
values-params         = field-name-param, space, where-param, space, size-param, {space, flags-param} {space,
start-meta-param}, {space, end-meta-param}, {space, threshold-param}, {space, aggregate-funcparam}, {space,
aggregate-field-param}, {space, min-param}, {space, max-param}, {space, search-param}
; field-name-param      = "fieldName=", (meta-key | entity) ; where-param
= "where=", where-clause ; size-param            = "size=", ? integer between 1
and 1,677,721 ? ; start-meta-param      = ? same as query message ? end-meta-
param        = ? same as query message ?
flags-param           = "flags=", {values-flag, {"," values-flag} } ;
values-flag           = "sessions" | "size" | "packets" | "sort-total" | "sort-value" | "orderascending"
| "order-descending" ; threshold-flag        = "threshold=", ? non-negative integer ? ; aggregate-func-
param  = "aggregateFunction=", { aggregate-func-flag } ; aggregate-func-flag   = "count" | "sum" ;
aggregate-field-param = "aggregateFieldName=", ( meta-key | entity ) ; min-param             = "min=",
meta-value ; max-param             = "max=", meta-value ; search-param          = "search=", search-
string ;
```

The `values` call provides the function of returning a set of unique meta values for a given meta key. For each unique value, the `values` call can provide an aggregate total count. The function used to generate the total is controlled by the fags parameter.

Parameters

The following table describes the function of each parameter.

Parameter	Function
`fieldName`	This is the meta key name for which you retrieve unique values. For example, if `fieldName` is `ip.src`, this function returns the unique source IP values in the collection. Entities can be used for the field name, in which case the result is defined as the combined set of field values for all the referenced meta keys. If the `fieldName` refers to a key with rename references, the result is defined as the combined set of field values for the given meta key name plus all of the references' meta keys.

`where`	This is a `where` clause which filters the set of sessions for which the unique values are returned. For example, if the `fieldName` is `ip.src`, and the `where` clause is `ip.src = 192.168.0.0/16`, only values in the range of `192.168.0.0` to `192.168.255.255` are returned. For information on the `where` clause syntax, see *Where Clauses*.
`size`	The size of the set of unique values to return. This function is optimized to return a small subset of the possible unique values in the database.
`id1`, `id2`	These optional parameters limit the scope of the search for unique values to a specific region of the meta database and the index. Setting the `id1` and `id2` parameters to a limited range of the meta database is very important to running searches quickly on large collections.
`flags`	Flags control how the values are sorted and totaled. Flags are described in the following Values Flags section.
`threshold`	Setting the `threshold` parameter allows the `values` call to short-cut collection of the total associated with each value once the threshold is reached. By providing a threshold, the caller can reduce the amount of index and meta items that must be retrieved from the database. If the `threshold` parameter is omitted or set to 0, this optimization is not used.
`aggregateFunction`	Optional parameter used to change the default behavior from counting sessions, packets, or size to counting or summing the numeric field defined by `aggregateFieldName`. Both parameters must be specified when either is defined. Pass either `sum` or `count` to specify which behavior to perform.
`aggregateFieldName`	The meta field on which to perform the `aggregateFunction`. Both `aggregateFunction` and `aggregateFieldName` parameters must be specified when the `aggregate` flag is set. Performing a `values` call using one of the aggregate functions can be significantly slower than a `values` call that collects totals of sessions, packets, or size. The reason for this is that each session that matches the `where` clause must be retrieved from the meta database. This scan causes a large portion of the query to be I/O bound on the meta DB volumes. The time taken to run an aggregate `values` call is linearly proportional to the number of sessions that match the `where` clause.
`min`, `max`	The minimum and maximum value that should be returned from the call. These parameters are used to iterate (or page) over an extremely large number of values, typically more values than could be returned from a single call. Primarily used in conjuction with the flags `sort-value`,`sort-ascending` such that the highest value returned would be used in a subsequent call as the `min` parameter value. The values are exclusive. If `min="rsa"` was specified and `rsa` was a valid value, `rsa` would not be returned; instead, the next highest value would be returned.
`search`	Text search pattern to be used to further refine the `where` parameter

`values` Flags

The `flags` parameter controls how the values call operates. There are three groups of fags that correspond to the diferent modes of operation as shown in the following table.

Flag	Description
`sessions`, `size`, `packets`	The `values` call allows you to specify one of these flags to determine how the total for each value is calculated. If the flag is `sessions`, the `values` call returns a count of sessions that contain each value. If the flag is `size`, the `values` call totals the size of all sessions that contain each unique value, and reports the total size for each unique value. If the flag is `packets`, the values call totals the number of packets in all sessions that contain each unique value, and then reports that total for each unique value.
`sort-total`, `sortvalue`	These flags control how results are sorted. If the flag is `sort-total`, the result set is sorted in order of the totals collected. If the flag is `sort-value`, the results are returned in order of the sorting order of the values.

order-ascending, orderdescending	These flags control the sort order of the result set. For example, if sorting by total in descending order, the values with the greatest total are returned first.
suggest	Enables suggestion mode for the values API. All other flags are ignored if this flag is set
databasescan	Make the values call bypass the index and instead collect unique values as if where traversing the meta database. This mode is slow on most cases, but it can be fast if the where clause matches very few sessions.
ignore-cache, clearcache	These flags control result set caching on the set of values returned by this call. Normally these should not be used.

values Call Example

The values call is used extensively by the Navigation view in NetWitness Platform. The default view generates calls that look like this:

```
/sdk/values id1=198564099173 id2=1542925695937 size=20 flags=sessions,sort-total,order-descending
threshold=100000 fieldName=ip.src where="time=\"2014-May-20 13:12:00\"-\"2014-May-21 13:11:59\""
```

In this example, the Navigation view requess unique values for ip.src . It requess unique values of ip.src in the time range given. It asks for the count of sessions that match each ip.src , and the results are the top 20 ip.src values when sorted by the number total count of sessions in descending order. In addition, the Navigation view has a meta ID range in order to provide an optimization hint to the query engine.

Values call and bucketing mode

When a values call is executed with a fieldName parameter that specifes a bucketed indexed meta, the sysem will only return the bucket values present within the res of the criteria. This has the side efect of producing counts and totals that represent all sessions within each returned bucket. This is useful because it summarizes size meta into groups that represent human-readable ranges like 1 MB, 2 MB, and so one.

When the number of sessions scanned by the values call drops below 1000 sessions, the values call operates in meta-scanning mode, and at that point it returns exact values for numeric value indexes, regardless of the bucketing setting on the index.

Suggesion Mode

The values call has an additional execution mode that is used to provide suggesed search values. In this mode of operation, the values call only identifes unique values sored with the given meta key name. It provides these results within milliseconds. To achieve this it does not provide any session counts, it will not utilize any other sort fags. The suggesion mode does utilize the where parameter to refne suggesions, but it only utilizes the time range clause if provided. Other portions of the where clause are not utilized to refne suggesion.

Suggesion mode is enabled by setting the suggest fag in the flags parameter.

Suggesion mode gives special meaning to the min parameter. The min parameter can contain the sarting point for the suggesed values.
The return values of suggest mode will only include values that sart with the text provided in the min parameter.

search parameter

The values API supports a search parameter to perform free text searching. The syntax of the search parameter is identical to the search parameter utilized by the msearch API call. Refer to the msearch documentation for a description of the search feld syntax.

The search parameter acts as an extension of the where parameter. This means that the where and search parameters work together.
Any where parameter specifed is combined with the search flter as if they were joined with an AND operation.

Searches through the values API are always done agains indexed meta, in case-insensitive mode.

The `values` API only operates on index entries for mos requess, in order to provide fas totals over a large numbers of events. When the search parameter operates over in-exact indexes, such as N-Gram indexes, it may include sessions with near matches insead of narrowing the search to exact matches.

/sys

The sysem tree holds the main functionality related to overall service confguration, scheduler and access to the Stats Database.

The following is a brief overview of the sysem commands and what they do:

- **save** - Saves the current configuration to disk. Any configuration change will be written to disk within a couple of minutes or shutdown, so this is typically redundant. **caCert** - List, add or delete trusted certs on the system. **shutdown** - Saves all data and then performs a clean shutdown of the service. Typically the
- service will restart based on systemd settings. **fileEdit** - Used to view and edit configuration files, index files, parsers and even the scheduler jobs.
- **peerCert** - List, add or delete trusted peer certs on the system. This enables clients to connect using a trusted SSL connection.
- **servCert** - Displays the current server certificate in PEM format **statHist** - Query
- historical stats stored in the Stats Database.
-
-

/sys caCert

Parameter	Description
op	Operation to perform on the CA certificate list (list, delete, add)
id	ID of CA certificate to remove (only valid with op=delete)

Manipulate the certifcate authority (CA) lis used by this service. The Certifcate Authority lis is used to verify the authenticity of peer certifcates.

You mus specify what operation to perform on the certifcate authority lis: lis, delete, or add The **op=lis**

command will output a lis of the known certifcate authority certifcates.

The **op=delete** command will remove a certifcate from the certifcate authority lis. The delete command mus be accompanied by an id parameter. The id refers to the id of the certifcate authority certifcate to remove from the certifcate authority lis. Use the lis command to see a lis of certifcate authority id's.

The **op=add** command will upload an additional certifcate to the service, and add it to the certfciate authority lis. This command can only be invoked programmatically. The REST interface provides a simple HTML interface to upload a certifcate authority certifcate using the add operation. This is available at /sys/caupload on the REST interface.

/sys fleEdit

Parameter	Description
op	The operation to perform (dir, get, getBackup, put, create, delete)
type	The type of file (parser, filter). If not provided, it will use the file extension of the filename parameter.
filename	The filename to create, edit, retrieve or delete.

The purpose of this command is to provide an API to retrieve the contents of fles that can be edited by the user. This includes Lua parsers, cusom index fles, scheduler jobs and a few other miscellaneous fles. The workfow around this API is thus:

1. Retrieve the list of files that can be edited by sending the parameter **op=dir**. This will return a file list pair containing the filename and it's size in bytes.

2. Using the **op=get filename=<filename>** command, retrieve the contents of the file to edit.
3. After editing, you can then save the file back on the service by using **op=put filename=<filename>**.

When using the put operation, the actual contents of the fle mus come after the parameters using the message type called PARAMS_BINARY. This message type can be sent by NwConsole using the following example command:

```
send /sys fileEdit op=put filename=index-concentrator-custom.xml --filedata="/Users/user/Documents/index-concentrator-custom.xml"
```

The **op=create type=parser flename=MyLuaParser.lua** command can be used to add new Lua parsers to the sysem. This will not automatically reload the parser.

To delete fles, you can use **op=delete flename=<flename to delete>**.

/sys peerCert

Parameter	Description
op	Operation to perform on the peer certificate list (list, delete, add)
id	ID of CA certificate to remove (only valid with op=delete)

Manipulate the peer certifcate lis used by this service. The peer certifcate lis is used to grant super-user, or "trused" access to a connection. The if the client provides a certifcate that matches one of the certifcates in the peer certifcate lis, the client is trused. The peer certifcate is *not* trused based on certifcate chain trus. Trus is only granted if the client's certifcate is explicitly added to the peer lis.

You mus specify what operation to perform on the peer certifciate lis: lis, delete, or add

The **op=lis** command will output a lis of the known peer certifcates.

The **op=delete** command will remove a certifcate from the peer certifcate lis. The delete command mus be accompanied by an id parameter. The id refers to the id of the peer certifcate to remove from the peer lis. Use the lis command to see a lis of peer id's.

The **op=add** command will upload an additional certifcate to the service, and add it to the peer certifcate lis. This command can only be invoked programmatically. The REST interface provides a simple HTML interface to upload a peer certifcate using the add operation. This is available at /sys/truspeer on the REST interface.

/sys save

Parameter	Description
force	Forces the configuration to be written to disk, regardless if there are any changes that require a save

Saves the current confguration to disk. This command is redundant because the sysem automatically saves changes within a couple of minutes and always on shutdown.

/sys servCert

Display the public certifcate for this service, in PEM format.

/sys shutdown

Parameter	Description
reason	The reason for the shutdown. This is saved to the log.
cl	Short for command line. This is an advanced parameter used to set options on the command line of the service upon restart.

This will perform a normal shutdown of the service, saving all fles and then exit cleanly. Typically, the service will resart based on the sysemd confguration.

/sys satHis

Parameter	Description
time1	The starting time for retrieving stats.
time2	The ending time for retrieving stats.
timeFormat	Specify the time format for each stat snapshot (posix, simple), default is posix (seconds since 1970).
include	Comma separated list of stats to include (wildcards allowed)
exclude	Comma separated list of stats to exclude (wildcards allowed, has precedence over include)
reduce	If true, reduces data transmission by replacing stat pathnames with a shorthand and provides a lookup table as first result.
onChange	Comma separated list of stats for tracking changes. Only results where at least one stat in the list has changed will be returned.
showAll	If true, changes each result to show all stats previously seen, even if the value hasn't changed. It has a moderate performance impact and is best used with include, exclude or onChange parameters to reduce the returned result set size.

Retrieve hisorical sats from the sats db. Don't send time1/time2 to get bounding times about sats db. Supported wildcards are ? to match any single char and * to match zero or more characters, not including slash /, ** to match zero or more characters including slash /.

With no params, you get information about the how many records are in the database and the bounding time range:

```
/sys statHist
[  total: 1188  time1:
1336497862  time2: 1336499092
]
```

By default, it returns it in posix time, which is the number of seconds since 1970. You can ask for a readable date:

```
/sys statHist timeFormat=simple
```

```
[  total: 1260  time1: 2012-May-08
17:24:22  time2: 2012-May-08 17:46:08
]
```

You can craft a query to return sats within a time range like so (some results snipped for readability):

```
/sys statHist time1="2017-01-19 15:00:00" time2="2017-01-19 15:00:02" timeFormat=simple
[
 /logs/stats/first.id=202028111
 /logs/stats/last.failure.id=205165789
 /logs/stats/last.id=207753128
 /logs/stats/last.warning.id=205849692
 /logs/stats/total=5725018
 /rest/stats/service.status=Ready
 /sdk/stats/queries.active=0
 /sdk/stats/queries.pending=0
 /sdk/stats/values.calls=10369
 /sdk/stats/values.calls.cached=1
 /storedproc/stats/storedproc.active=0
 /storedproc/stats/storedproc.pending=0
 /sys/stats/config.filename=NwConcentrator.cfg
 /sys/stats/cpu=0%
 /sys/stats/hostname=LOKI
 /sys/stats/memory.process=31838142464
 /sys/stats/memory.process.max=135460544512
 /sys/stats/memory.system=116689436672
 /sys/stats/memory.total=135460544512
 /sys/stats/module=concentrator
 /sys/stats/release=0.0.0  /sys/stats/version=11.0.0.0
time="2017-Jan-19 15:00:00"
]
[
 /logs/stats/last.id=207753130
/logs/stats/total=5725020  time="2017-Jan-19 15:00:01"
]
[
 /sys/stats/memory.system=116689952768  time="2017-Jan-19 15:00:02"
]
```

Notice how the frs second has a bunch of sats, but the next 2 do not? What happens is the sat db only sores values that have changed since the las update. However, that would be a problem for sats that rarely change if you wanted to plot them in a chart, so every 5 mins on the 5 min mark, *every* sat is sored again. When the service frs sarts, it also sores every sat in the db, then follows with only sat changes until the next 5 min mark. Bottom line: when charting, it's bes to make sure time1 is on a 5 minute mark.

You can also limit the sats you are interesed in by using include or exclude:

```
/sys statHist time1="2017-01-19 15:00:00" time2="2017-01-19 15:00:05" timeFormat=simple
include=/sys/**,/logs/stats/*
```

This command would show the results above, but only for sats under the whole /sys tree and the immediate sats under /logs/sats.

If you would like to see every sat value with every time period, you can use the parameter **showAll=true**. This parameter will carry forward every sat even if the value hasn't changed. However, performance is noticeably impacted when doing this, especially when querying large time ranges. It will also greatly expand the size of the result set and is bes used with **include**, **exclude** or **onChange** to reduce the returned size.

For example, the sample above would look like this with **showAll=true** :

```
/sys statHist time1="2017-01-19 15:00:00" time2="2017-01-19 15:00:02" timeFormat=simple showAll=true
[
 /logs/stats/first.id=202028111
 /logs/stats/last.failure.id=205165789
 /logs/stats/last.id=207753128
 /logs/stats/last.warning.id=205849692
 /logs/stats/total=5725018
```

```
 /rest/stats/service.status=Ready
 /sdk/stats/queries.active=0
 /sdk/stats/queries.pending=0
 /sdk/stats/values.calls=10369
 /sdk/stats/values.calls.cached=1
 /storedproc/stats/storedproc.active=0
 /storedproc/stats/storedproc.pending=0
 /sys/stats/config.filename=NwConcentrator.cfg
 /sys/stats/cpu=0%
 /sys/stats/hostname=LOKI
 /sys/stats/memory.process=31838142464
 /sys/stats/memory.process.max=135460544512
 /sys/stats/memory.system=116689436672
 /sys/stats/memory.total=135460544512
 /sys/stats/module=concentrator
 /sys/stats/release=0.0.0  /sys/stats/version=11.0.0.0
time="2017-Jan-19 15:00:00"
]
[
 /logs/stats/first.id=202028111
 /logs/stats/last.failure.id=205165789
 /logs/stats/last.id=207753130
 /logs/stats/last.warning.id=205849692
 /logs/stats/total=5725020
 /rest/stats/service.status=Ready
 /sdk/stats/queries.active=0
 /sdk/stats/queries.pending=0
 /sdk/stats/values.calls=10369
 /sdk/stats/values.calls.cached=1
 /storedproc/stats/storedproc.active=0
 /storedproc/stats/storedproc.pending=0$
 /sys/stats/config.filename=NwConcentrator.cfg
 /sys/stats/cpu=0%
 /sys/stats/hostname=LOKI
 /sys/stats/memory.process=31838142464
 /sys/stats/memory.process.max=135460544512
 /sys/stats/memory.system=116689436672
 /sys/stats/memory.total=135460544512
 /sys/stats/module=concentrator
 /sys/stats/release=0.0.0  /sys/stats/version=11.0.0.0
time="2017-Jan-19 15:00:01"
]
[
 /logs/stats/first.id=202028111
 /logs/stats/last.failure.id=205165789
 /logs/stats/last.id=207753130
 /logs/stats/last.warning.id=205849692
 /logs/stats/total=5725020
 /rest/stats/service.status=Ready
 /sdk/stats/queries.active=0
 /sdk/stats/queries.pending=0
 /sdk/stats/values.calls=10369
 /sdk/stats/values.calls.cached=1
 /storedproc/stats/storedproc.active=0
 /storedproc/stats/storedproc.pending=0
 /sys/stats/config.filename=NwConcentrator.cfg
 /sys/stats/cpu=0%
 /sys/stats/hostname=LOKI
 /sys/stats/memory.process=31838142464
 /sys/stats/memory.process.max=135460544512
 /sys/stats/memory.system=116689952768
 /sys/stats/memory.total=135460544512
 /sys/stats/module=concentrator
 /sys/stats/release=0.0.0  /sys/stats/version=11.0.0.0
time="2017-Jan-19 15:00:02" ]
```

As you can see, the result size grew and even the sat values that didn't change are included.

A very useful parameter is **onChange=<sat pathname>[,<sat pathname>]**. This will only return results when the lised sats change. Only one sat has to change, not all of them, to be returned.

For example, suppose you want to see when Decoder had packet drops and the capture rate at that time:

```
time1="2017-01-19 15:00:00" time2="2017-01-19 15:10:00" timeFormat=simple
include=/decoder/stats/capture.rate,/decoder/stats/capture.dropped,time
onChange=/decoder/stats/capture.dropped [ /decoder/stats/capture.rate=3876 /decoder/stats/capture.dropped=102
time="2017-Jan-19 15:05:57" ]
[ /decoder/stats/capture.dropped=210 time="2017-Jan-19 15:08:02" ]
```

So, in the result set above, the sat **capture.dropped** changed twice in the 10 minute period requesed. But for the second result, we don't know what the capture rate was because in that time period, it didn't change from a previous time period and wasn't recorded. This is where **showAll=true** comes in handy:

```
time1="2017-01-19 15:00:00" time2="2017-01-19 15:10:00" timeFormat=simple
include=/decoder/stats/capture.rate,/decoder/stats/capture.dropped,time
onChange=/decoder/stats/capture.dropped showAll=true [ /decoder/stats/capture.rate=3876
/decoder/stats/capture.dropped=102 time="2017-Jan-19 15:05:57" ]
[ /decoder/stats/capture.rate=4205 /decoder/stats/capture.dropped=210 time="2017-Jan-19 15:08:02" ]
```

Now we know what the capture rate was at the time of the second drop. The query kept track of previous values for **capture.rate** and placed the value in the second result. This can be very useful when doing adhoc queries, but keep in mind it will be somewhat slower to return results.

Confguration of the Stats Database

There are 3 confguration nodes that afect how the sat database works. They are all under /sys/confg.

- **stat.compression** - Determines whether or not the records in the stat db are compressed or not. The values can be none, gzip or zstd. If you enable `gzip` or `zstd` compression, the retention of the historical stats will be greatly increased. However, there will be a moderate performance penalty when doing a query, as it will have to uncompress each record on the fly to return results.
- **stat.dir** - The directory on the device where the stats db will be stored. The default location is /var/lib/netwitness/<service>/statdb. More importantly, this value also indicates how large the stat db can grow before it goes into rollover. Each directory can be assigned a maximum size (e.g., /var/lib/netwitness/<service>/statdb=1 GB). If you would like to store more stats, you can increase this size, as long as you have enough disk space.
- **stat.exclude** - This is a comma separated list of stat paths that will be excluded from getting stored in the stats database. The more you exclude, the longer the retention for the stats that are included. A sample value is something like this: /users/roles/*,/connections/**,/services/**,/sdk/stats/queries/**,/decoder/devices/*. As you can see, wildcards are supported, so you can exclude whole subtrees. A single asterisk means only exclude those stats that match at that subtree level, but it is not recursive. Two asterisks means recursively exclude stats in subtrees as well.

/decoder/parsers

Confguration

Dynamic Header Thresholds

The Dynamic Header Thresholds allow headers to identify devices without a message match.

Across devices (parsers), a priority is assigned to each header refecting the complexity of the `content` attribute (pattern) of the header. The more generic a header is the lower the match priority. The more disinct and complex a header is the higher the match priority.

As example, the following header has a very high match priority (**4503**):

```
%CACHEFLOWELFF_syslog:<hfld1>,cs-method="<messageid>",<!payload:hfld1>
```

An example of a header with a very low match priority (**8**):

```
<msgIdPart1> <msgIdPart2>, <!payload:msgIdPart1>
```

The explicit match priority value for each header can be viewed in the logs when debug logging is enabled.

As the header matches are tracked, high and low thresholds are dynamically applied proportionately agains the header with this highes observed match priority. Header matches above the high threshold can be considered even without a known message group ID. Matches below the low threshold which have a hardcoded message group ID or fail to match a message are removed from consideration. This logic helps increases the likelihood of a single device match for srong headers.

- Threshold Percent (**header.threshold.percent**) - Percentage used to establish the high and low match priority thresholds. Setting of '0' disables this feature. Changes take effect on parser reload.
- Minimum Score (**header.threshold.min.score**) - Minimum observed match priority required to engage the high threshold. Setting of '0' disables the high threshold. Changes take effect on parser reload.

Header Prioritization

Message matches resulting from a Message Group ID extracted from message payload can be prioritized over matches using a hardcoded ID. Though principally content driven, default treatment is confgurable to minimize the number of headers which will require edits.

- Shortcut Score (**header.prioritize.shortcut.score**) - Priority score, at and below which, matches resulting from a Message Group ID extracted from message payload are prioritized over those with a hardcoded ID. Content specified treatment takes precedence over configuration. Changes to this value could require changes to content to compensate for changes in header treatment.

Dynamic Mapping

Message matches when fagged by the content elements `<HEADER>` or `<MESSAGE>` with the attribute `discoverable="map"` trigger a temporary event source mapping which directs future parsing to the matched device type. A mapped source is valid for a confgured amount of time. Each time a subsequent match is fagged by content, the expiration time is extended for the source.

If directed parsing to the mapped device does not provide a match, parsing will fall back to discovery. This allows the source to be, if required, dynamically mapped to multiple device types. It also allows for handling repurposed source addresses. Each mapped device type maintains its own expiration time.

- Mapping Minutes (**dynamic.mapping.minutes**) - Delta time in minutes for which a content defined mapping should be retained.

Receiving Protobuf Encoded Logs

Logdecoder can receive 4 byte length prefxed `com.rsa.netwitness.carlos.nextgen.Event` protobuf objects on port 50202 (default). The connection mus be initiated with byte sring:

```
0x4E 0x47 0x43 0x50 0x01 0x00
```

- `raw_message` - [required] if not present or present with a length of 0, the log will not be re-written. Otherwise, this value, along with the log header will be the serialized packet.
- `collection_meta`
- `lc.msgtype` - [required] only protobuf events that have this field and its value is Unstructured will be re-written as raw logs `lc.srcid` - [optional] this will be
- added to the log header in the source field and typed appropriately (text/ipv4/ipv6), text values will be parsed to determine if they are ipv4/ipv6 addresses.
- `lc.cid` - [optional] this is expected as text and will be added to the log header if present
- `content_meta`
- `syslog.pri` - [optional] Will be added to the log header if present (0 otherwise), expects either a text or uint32 value (text must be parseable as a uint32).

LD checks the following felds in the `Event` insances:

All felds of collection and content meta will be checked for entries in the table map to determine if meta should be created.

Messages

- **tzinfo** - Retrieve and update timezone settings used for log event processing. **dyndvmap** - Display,
- purge and force persistence of dynamically mapped event sources.

/decoder/parsers dyndvmap

The `dyndvmap` command provides an interface to display, purge and force persisence of dynamically mapped event sources.

Parameters and Results

- **op** - Operation to perform: `describe` , `clear` or `save` . Defaults to `describe` **deviceFilter**
- - A regex that will only return matching devices. **sourceFilter** - A regex that will only return
- matching sources. **forwarderFilter** - A regex that will only return matching forwarders.
-

Display Mapped Sources

Expired mappings are only purged on synchronization between parse threads. The reported mappings from this command will only include those which have **not** expired. To view the size of the mappings container query the `parsers` sat `mappings.dynamic.source.count` .

View all **valid** dynamically mapped event sources.

```
dyndvmap op=describe
```

View only mappings for a particular device type.

```
dyndvmap op=describe deviceFilter=rhlinux
```

View only mappings for a particular source.

```
dyndvmap op=describe sourceFilter=1.1.1.1
```

Purge Mappings

Flush **all** mapped sources.

```
dyndvmap op=clear
```

Force Persisence

Force the mappings to be saved to disk.

```
dyndvmap op=save
```

/decoder/parsers tzinfo

The **tzinfo** command provides an interface to view timezone information used for converting dates parsed from a captured log to normalized UTC timesamps. This information is derived from the IANA Time Zone Database insalled on the service hos. As timezone abbreviations do not necessarily represent a single GMT ofset, **tzinfo** also provides a means to override the defnition of an abbreviation as well as defne entire new abbreviations.

Parameters and Results

- **op** - The operation to perform: `version` , `names` , `abbreviations` , `duplicates` , `overrides` , `add` or `remove` **abbr** - a
- timezone abbreviation (e.g. `EST`) **gmtoffset** - an offset from GMT in the format [+/-]HH[MM]
- **tzname** - A timezone name (e.g. `America/New_York`) which must be a known timezone (see **Timezone Names**)
-

Version

View the version of the IANA Time Zone Database currently in use.

```
tzinfo op=version
```

Timezone Names

View the name, abbreviations and GMT ofsets for all known timezones.

```
tzinfo op=tznames
```

For each known timezone, the following will be returned:

- **tzname** - The timezone name
- **stdAbbr** - The abbreviation the timezone currently uses for standard time **stdOffset** - The offset
- from GMT the timezone currently uses for standard time **dstAbbr** - The abbreviation the timezone
- currently uses for daylight savings time **dstOffset** - The offset from GMT the timezone currently
- uses for daylight savings time

An empty value for **dsAbbr** indicates that a timezone does not currently observe daylight savings time. Also note that the ofsets returned here refect the entries in the underlying Time Zone Database and are not impacted by **Overridden Abbreviations**.

Timezone Abbreviations

View all current timezone abbreviations, the default GMT ofset and the **Abbreviation Override** ofset if one exiss.

```
tzinfo op=abbrs
```

For each abbreviation defnition, the following will be returned:

- **abbr** - The abbreviation **gmtoffset** - The offset from GMT as defined by the timezone
- database **override** - A specified GMT offset used (see **Overridden Abbreviations**)
-

Duplicates Abbreviations

View all abbreviations that have multiple values for GMT ofset in disinct timezones.

```
tzinfo op=duplicates
```

For each abbreviation defnition that has multiple ofset defnitions in the timezone database, the following will be returned:

- **abbr** - The abbreviation **gmtoffset** - The offset
- from GMT
- **tzname** - The timezone that defines this abbreviation and GMT offset

Abbreviations that are currently overridden and have a GMT ofset or timezone explicitly set will not be displayed as a duplicate.

Overridden Abbreviations

View all abbreviations that have had GMT ofset or timezone explicitly set.

```
tzinfo op=overrides
```

For each overridden abbreviation, the following will be returned:

- **abbr** - The abbreviation **gmtoffset** - If the override is a specific offset, offset
- from GMT **tzname** - If the override specifies a timezone
- **valid** - Indicates wether or not the setting was successfully loaded and is currently active
-

Add/Edit an Abbreviation Override

Add a new abbreviation or override the timezone database defnition of an exising abbreviation. Adding an exising abbreviation override will update that abbreviation.

An abbreviation can be defned as a specifc GMT ofset. The following creates the abbreviation `EQT` with a GMT ofset of -5 hours:

```
tzinfo op=add abbr=EQT gmtoffset=-0500
```

A duplicate abbreviation can be overridden from the default defnition. The following changes the default ofset of `CST` from -6 hours to -5 hours as defned by the timezone `Cuba` :

```
tzinfo op=add abbr=CST tzname=Cuba
```

The timezone mus be known (see **Timezone Names**) and defne the provided abbreviations as sandard or daylight.

Remove an Abbreviation Override

Remove an abbreviation that was added to **Overridden Abbreviations**.

```
tzinfo op=remove abbr=EQT
```

/sdk deviceId

The *deviceId* command takes a session ID on the current device and returns the originating device name and remote session ID on the originating device. This command is only useful for sessions that have been aggregated from another service.

```
NOTE: This command does not track a session down to the originating device (e.g., Broker -> Concentrator ->
Decoder), only one step removed.  So if you run this on a Broker, it will give you the session on the
Concentrator, not the Decoder.
```

Parameters: • **session** - The session ID to look up

Returns:

- **device** - The originating device name that *session* was aggregated from **session** - The
- session ID on the originating device

/appliance

Storage Array Confguration Utilities

The seps for attaching sorage to your NetWitness Platform are:

1. Physically attach the storage and make it available to this host. For direct-attach storage, the RAID manipulation commands will perform the task of constructing hardware RAID volumes and making these volumes appear as drives. For SAN storage, you must allocate the storage through the SAN's management tools and present them to this host. For either type of storage, the result will be that the storage appears as block devices, such as /dev/sdc in the case of direct-attach drives, or /dev/emcpowera in the case of an EMC SAN drive.
2. Allocate the block devices as usable storage. You can use the `partNew` command to take block storage and organize it into filesystems to be used by storage. The result of this is that the block device is put into a volume group, such as `decoder` , `concentrator` , or `index`

3. Allocate the storage to a service. You can use the `srvAlloc` command to assign the filesystems that are created to services that are running on this host.

Summary of Array Confguration Commands

Each of the commands lised below has built-in help that describes their function and usage. If you are using the REST interface, select the command from the drop-down menu to see the help text.

Commands that deal with direct-attach RAID volumes

- **raidList** - List the RAID controllers and direct-attach enclosures that are present on this host **raidNew** -
- Allocate direct-attach enclosures into block devices **raidClr** - De-allocate direct-attach enclosures used as
- block devices

Commands that deal with allocating block devices as sorage

- **devlist** - List the block devices that are present on this host **partNew** - Allocate
- partitions on a block device and create volume groups **partClr** - Remove volume
- groups from a block device **vgs** - Summarize how block devices are organized into
- volume groups

Commands that deal with allocating sorage to services

- **srvList** - List services on the host and their allocated storage paths **srvAlloc** -
- Allocate a volume group to a service **srvFree** - Remove a volume gorup from a
- service

/appliance partClr

Uninitialize a block sorage device

This command clears data from a block sorage device. It performs these tasks:

1. Unmount any filesystems within volume groups on the device.
2. Remove any volume groups the device is a part of.
3. Wipe the physical volume labels from the device.
4. Create a new, empty partition table on the device.

Notes:

- You must remove the block device from all service configurations.
- You must stop the service if you are removing the last block device from that service.
- All `decoder` volumes must be removed before `decodersmall` volumes
- All `logdecoder` volumes must be removed before `logdecodersmall` volumes
- All `index` volumes must be removed before `concentrator` volumes

/appliance partNew

Initialize a block sorage device

This command prepares a sorage device to use in NetWitness. It performs these tasks:

1. Creates the partition table on the block device.
2. Creates the Linux Volume Manager physical device partition.
3. Creates a volume group containing the physical device.
4. Creates logical volumes in the volume group.
5. Creates XFS filesystems on each logical volume.
6. Creates /etc/fstab entries for each logical volume.
7. Mounts each logical volume.

Usage

You mus provide the block device name, for example, "sdc" or "/dev/sdc". Use the devlis command to locate unused block devices. You mus provide a name for the service that will be used with the sorage, for example, "decoder" or "concentrator". You may optionally provide the volume type that will be created. The default volume type has the same name as the service.

By default, the partNew command does **not** make changes. Insead it prints a report of the actions that will be taken. To actually make the changes to the sysem, add the commit=true parameter to the command.

Example: Assign devices `sdd` and `sde` to Decoder.

```
send /appliance partNew name=sdd service=decoder volume=decodersmall send
/appliance partNew name=sde service=decoder volume=decoder
```

Note that the decodersmall volume mus be created before the decoder volume, because it holds the small flesysem mounted at /var/netwitness/decoder.

Example: Assign device `sdc` to Archiver

```
send /appliance partNew name=sdc service=archiver
```

Note that we do not need to specify a volume type. It is assumed that the volume type is "archiver".

Netwitness Service Volume Reference

Service volume names

Service	Volume Name	Internal/External Storage Config (JBOD/PV)	Filesystems Created
decoder	decoder	External	packetdb
decoder	decodersmall	External	decoder root, index, sessiondb, metadb
decoder	hybrid-decoder-meta	Internal	decoder root
decoder	packet	Internal	packetdb
logdecoder	logdecoder	External	packetdb
logdecoder	logdecodersmall	External	logdecoder root, index, sessiondb, metadb
logdecoder	hybrid-logdecoder-meta	Internal	logdecoder root
logdecoder	logpacket	Internal	packetdb
logdecoder	endpoint-log-hybrid	Internal	mongo, packetdb
logdecoder	log-indexed-decoder	Internal	logdecoder root
logdecoder	logindex	Internal	logdecoder index
concentrator	concentrator	External	concentrator root, metadb, sessiondb
concentrator	hybrid-concentrator	Internal	concentrator root

concentrator	index	External	index
archiver	archiver	External	database

Volume sizing

Volume	Filesystem	Mount Point	Size
decodersmall	decoroot	/var/netwitness/decoder	20 GB
decodersmall	index	/var/netwitness/decoder/index	30 GB
decodersmall	sessiondb	/var/netwitness/decoder/sessiondb	600 GB
decodersmall	metadb	/var/netwitness/decoder/metadb	all remaining free space on decodersmall volume
decoder	packetdb	/var/netwitness/decoder/packetdb	all space on decoder volume
packet	packetdb	/var/netwitness/decoder/packetdb	all space on packet volume
hybrid-decoder-meta	decoroot	/var/netwitness/decoder	all space on hybrid-decoder-meta volume
logdecodersmall	decoroot	/var/netwitness/logdecoder	20 GB
logdecodersmall	index	/var/netwitness/logdecoder/index	30 GB
logdecodersmall	sessiondb	/var/netwitness/logdecoder/sessiondb	600 GB
logdecodersmall	metadb	/var/netwitness/logdecoder/metadb	all remaining free space on logdecodersmall volume
hybrid-logdecodermeta	decoroot	/var/netwitness/logdecoder	all space on hybrid-logdecoder-meta volume
log-indexed-decoder	decoroot	/var/netwitness/logdecoder	all space on log-indexed-decoder volume
logdecoder	packetdb	/var/netwitness/logdecoder/packetdb	all space on logdecoder volume
logpacket	packetdb	/var/netwitness/logdecoder/packetdb	all space on logpacket volume
logindex	index	/var/netwitness/logdecoder/index	all space on logindex volume
concentrator	root	/var/netwitness/concentrator	30 GB
concentrator	sessiondb	/var/netwitness/concentrator/sessiondb	10% of free space on concentrator volume
concentrator	metadb	/var/netwitness/concentrator/metadb	all remaining free space on concentrator volume
hybrid-concentrator	root	/var/netwitness/concentrator	all space on hybrid-concentrator volume
index	index	/var/netwitness/concentrator/index	all space on index volume
archiver	database	/var/netwitness/archiver/database	all space on archiver volume
endpoint-log-hybrid	mongo	/var/netwitness/mongo	50% of free space on endpoint-log-hybrid volume
endpoint-log-hybrid	packetdb	/var/netwitness/logdecoder/packetdb	all remaining free space on endpoint-log-hybrid volume

/appliance vgs

Lis Volume Groups

This command enumerates all the volume groups on this hos. In addition, it displays the physical volumes that the volume group consiss of, and the logical volumes within the volume group.