



RSA[®] NetWitness

Version 11.7

Core Services API Guide



Contact Information

RSA Link at <https://community.rsa.com> contains a knowledge base that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

RSA Conference Logo, RSA, 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

This software and the associated documentation are proprietary and confidential to RSA Security LLC or its affiliates are furnished under license, and may be used and copied only in accordance with the terms of such license and with the inclusion of the copyright notice below. This 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.

This 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 RSA Link. 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 Security LLC or its affiliates ("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.

© 2020 RSA Security LLC or its affiliates. All Rights Reserved.

November 2021

Contents

OVERVIEW	20
CURRENT VERSION	20
SCHEMA.....	20
HTTP USAGE	21
<i>HTTP Verbs</i>	21
<i>HTTP Status Codes</i>	21
SERVICE API PORTS	22
CASE SENSITIVE.....	23
ERROR RESPONSE.....	23
AUTHENTICATION AND AUTHORIZATION	23
AUTHORIZATION	24
DECODER SERVICE	24
INTRODUCTION	24
API USAGE	24
<i>Sample Request</i>	25
<i>Sample Response</i>	26
HTTP Headers	26
JSON Response.....	26
/CONNECTIONS NODE.....	26
<i>API Messages</i>	26
closeAll	26
count	27
help	27
info	29
ls.....	29
netSpeed	29
/DATABASE NODE.....	30
<i>API Messages</i>	30
count	30
dbState	30
dump	31
hashInfo.....	32
help	33
info	34
ls.....	34
manifest	35
optimize	36
reconfig	37
resetMax	38
sizeRoll	38
stagger.....	40
timeRoll.....	41

wipe.....	42
/DECODER NODE.....	43
<i>API Messages</i>	43
count	43
dpsk	43
help	44
info	45
ls	45
reconfig	46
reset	47
resetMax	48
select	49
sslKeys	49
start	50
stop	50
whoAgg	51
/DECODER/PARSERS NODE.....	51
<i>API Messages</i>	51
content	51
count	52
delete	52
devices.....	52
feed	53
help	53
info	54
ls	54
reload	55
schema	55
snrtStat.....	56
upload	56
/INDEX NODE.....	57
<i>API Messages</i>	57
cacheClr.....	57
count	57
help	57
info	59
inspect.....	59
language.....	61
ls	62
profile	62
reconfig	63
save	64
schema	64
sizeRoll	64
values	65
/LOGS NODE	66
<i>API Messages</i>	66
count	66
download	66
help	68
info	69
ls	69

pull.....	70
timeRoll	71
/REST NODE.....	72
<i>API Messages</i>	72
count	72
help	72
info	73
ls.....	74
/SDK NODE	74
<i>API Messages</i>	74
aliases.....	74
cancel	75
content	75
count	77
delCache.....	77
help	77
hierarch	78
info	79
keyrefs.....	79
language.....	79
ls.....	80
msearch.....	81
packets	82
pin	84
precache.....	84
query	85
reconfig	87
search.....	87
session.....	88
summary	88
timeline	89
validate.....	90
values	91
xforms	93
/SYS NODE	94
<i>API Messages</i>	94
caCert	94
count	94
fileEdit	95
help	95
info	96
ls.....	97
peerCert	97
save	98
servCert	98
shutdown	98
statHist	99
telemetry.....	100
/USERS NODE.....	101
<i>API Messages</i>	101
addOrMod	101
auths.....	104

count	105
delete	105
help	105
info	106
ls	107
unlock	107
whoAml	108
LOGDECODER SERVICE	108
INTRODUCTION	108
API USAGE	108
<i>Sample Request</i>	109
<i>Sample Response</i>	109
HTTP Headers	110
JSON Response	110
/COLLECTIONS NODE	110
<i>API Messages</i>	110
count	110
create	110
help	111
info	112
ls	112
mounts	113
/CONNECTIONS NODE	113
<i>API Messages</i>	113
closeAll	113
count	114
help	114
info	116
ls	116
netSpeed	116
/DATABASE NODE	117
<i>API Messages</i>	117
count	117
dbState	117
dump	118
hashInfo	119
help	120
info	121
ls	121
manifest	122
optimize	123
reconfig	124
resetMax	125
sizeRoll	125
stagger	127
timeRoll	128
wipe	129
/DECODER NODE	130
<i>API Messages</i>	130
count	130

help	130
info	131
logStats	132
ls	133
reconfig	134
reparse	135
reset	136
resetMax	137
select	137
start	138
stop	138
whoAgg	138
/DECODER/PARSERS NODE.....	139
<i>API Messages</i>	139
content	139
count	139
delete	140
devices.....	140
dyndvmap.....	140
feed	141
headers.....	141
help	142
info	143
ipdevice	143
iptmzone	144
ls	145
patterns.....	146
reload	147
schema	147
tzinfo	148
upload	150
/INDEX NODE.....	150
<i>API Messages</i>	151
cacheClr.....	151
count	151
help	151
info	152
inspect.....	152
language.....	154
ls	155
profile	155
reconfig	156
save	157
schema	157
sizeRoll	157
values	159
/LOGS NODE	159
<i>API Messages</i>	159
count	159
download	160
help	161
info	162
ls	163

pull.....	163
timeRoll	164
/REST NODE.....	165
<i>API Messages</i>	165
count	165
help	165
info	166
ls.....	167
/SDK NODE	167
<i>API Messages</i>	167
aliases.....	167
cancel	168
content	168
count	170
delCache.....	170
help	170
hierarch	171
info	172
keyrefs.....	172
language.....	172
ls.....	173
msearch.....	174
packets	175
pin	177
precache.....	177
query	178
reconfig	180
search.....	180
session.....	181
summary	181
timeline	182
validate.....	183
values	184
xforms	186
/SYS NODE	187
<i>API Messages</i>	187
caCert	187
count	187
fileEdit	188
help	188
info	189
ls.....	190
peerCert	190
save	191
servCert	191
shutdown	191
statHist	192
telemetry.....	193
/USERS NODE.....	194
<i>API Messages</i>	194
addOrMod	194
auths.....	197

count	198
delete	198
help	198
info	199
ls	200
unlock	200
whoAml	201
CONCENTRATOR SERVICE.....	201
INTRODUCTION	201
API USAGE	201
<i>Sample Request</i>	203
Using URL encoded request	203
Using special content type	203
<i>Sample Response</i>	203
HTTP Headers	203
JSON Response	204
/CONCENTRATOR NODE	204
<i>API Messages</i>	204
add	204
count	206
delete	206
edit	206
help	208
info	209
ls	209
reconfig	210
reset	211
resetMax	212
start	212
status	212
stop	213
whoAgg	213
/CONNECTIONS NODE.....	214
<i>API Messages</i>	214
closeAll	214
count	215
help	215
info	216
ls	216
netSpeed	217
/DATABASE NODE.....	217
<i>API Messages</i>	217
count	218
dbState	218
dump	218
hashInfo.....	219
help	220
info	221
ls	222
manifest	222
optimize	223

reconfig	224
resetMax	225
sizeRoll	225
stagger.....	227
timeRoll	228
wipe.....	229
/INDEX NODE.....	230
<i>API Messages</i>	230
cacheClr	230
count	230
help	230
info	232
inspect.....	232
language.....	234
ls.....	235
profile.....	235
reconfig	236
save	237
schema	237
sizeRoll	237
values	238
/LOGS NODE	239
<i>API Messages</i>	239
count	239
download	239
help	241
info	242
ls.....	242
pull.....	243
timeRoll	244
/REST NODE.....	245
<i>API Messages</i>	245
count	245
help	245
info	246
ls.....	246
/SDK NODE	247
<i>API Messages</i>	247
aliases.....	247
cancel	248
content	248
count	250
delCache.....	250
deviceId	250
help	251
hierarch	252
info	252
keyrefs.....	252
language.....	252
ls.....	253
msearch.....	254
packets	255

pin	257
precache	258
query	259
reconfig	260
search	261
session	262
summary	262
timeline	262
validate	264
values	265
xforms	267
/SYS NODE	268
<i>API Messages</i>	268
caCert	268
count	268
fileEdit	268
help	269
info	270
ls	270
peerCert	271
save	272
servCert	272
shutdown	272
statHist	273
telemetry	274
/USERS NODE	275
<i>API Messages</i>	275
addOrMod	275
auths	278
count	278
delete	279
help	279
info	280
ls	280
unlock	281
whoAml	281
BROKER SERVICE	282
INTRODUCTION	282
API USAGE	282
<i>Sample Request</i>	283
Using URL encoded request	283
Using special content type	283
<i>Sample Response</i>	284
HTTP Headers	284
JSON Response	284
<i>Sample Request</i>	285
Using special content type	285
<i>Sample Response</i>	285
HTTP Headers	285
JSON Response	286

/BROKER NODE.....	287
<i>API Messages</i>	287
add	287
count	288
delete	288
edit	289
help	290
info	292
ls	292
repair	292
reset	293
resetMax	294
start	294
status.....	294
stop	295
whoAgg	295
/CONNECTIONS NODE.....	295
<i>API Messages</i>	296
closeAll	296
count	296
help	297
info	298
ls	298
netSpeed	299
/INDEX NODE.....	299
<i>API Messages</i>	299
count	299
gaps	299
help	300
info	301
ls	301
ranges.....	302
save	303
/LOGS NODE	303
<i>API Messages</i>	303
count	303
download	303
help	305
info	306
ls	306
pull.....	307
timeRoll	308
/REST NODE.....	309
<i>API Messages</i>	309
count	309
help	309
info	310
ls	311
/SDK NODE	311
<i>API Messages</i>	311
aliases.....	311

cancel	312
content	312
count	314
delCache	314
deviceId	314
help	315
hierarch	316
info	316
keyrefs	316
language	316
ls	318
msearch	319
packets	321
pin	323
precache	323
query	324
reconfig	326
search	327
session	327
summary	328
timeline	328
validate	330
values	331
xforms	334
/SYS NODE	334
<i>API Messages</i>	334
caCert	334
count	335
fileEdit	335
help	336
info	337
ls	337
peerCert	338
save	338
servCert	339
shutdown	339
statHist	340
telemetry	341
/USERS NODE	342
<i>API Messages</i>	342
addOrMod	342
auths	345
count	345
delete	346
help	346
info	347
ls	347
unlock	348
whoAml	348
ARCHIVER SERVICE	349
INTRODUCTION	349
API USAGE	349

<i>Sample Request</i>	350
Using URL encoded request	351
Using special content type	351
HTTP Headers	351
JSON Response	351
/ARCHIVER NODE	352
<i>API Messages</i>	353
add	353
count	354
delete	354
edit	355
help	356
info	357
ls	358
reconfig	358
repair	359
reset	359
resetMax	361
start	361
status	361
stop	362
whoAgg	362
/CONNECTIONS NODE	363
<i>API Messages</i>	363
closeAll	363
count	364
help	364
info	365
ls	365
netSpeed	366
/INDEX NODE	366
<i>API Messages</i>	366
count	366
gaps	367
help	367
info	368
ls	368
ranges	369
save	370
/LOGS NODE	370
<i>API Messages</i>	370
count	370
download	370
help	372
info	373
ls	373
pull	374
timeRoll	375
/REST NODE	376
<i>API Messages</i>	376
count	376
help	376

info	377
ls	378
/SDK NODE	378
<i>API Messages</i>	378
aliases	378
cancel	379
content	379
count	381
delCache	381
deviceId	381
help	382
hierarch	383
info	383
keyrefs	383
language	383
ls	385
msearch	386
packets	387
pin	389
precache	390
query	391
reconfig	393
search	393
session	394
summary	394
timeline	395
validate	397
values	398
xforms	401
/SYS NODE	401
<i>API Messages</i>	401
caCert	401
count	402
fileEdit	402
help	403
info	404
ls	404
peerCert	405
save	405
servCert	406
shutdown	406
statHist	406
telemetry	408
/USERS NODE	409
<i>API Messages</i>	409
addOrMod	409
auths	412
count	412
delete	412
help	413
info	414
ls	414
unlock	415

whoAml	415
APPLIANCE SERVICE.....	415
INTRODUCTION	416
API USAGE	416
<i>Sample Request</i>	416
<i>Sample Response</i>	417
HTTP Headers	417
JSON Response	417
/APPLIANCE NODE.....	417
<i>API Messages</i>	417
addFSMon	417
clock	418
count	418
delFSMon	418
devlist	419
dfiAlloc	419
dpToNic	419
ethtool.....	420
file	420
fwdLogs	421
help	421
info	422
ls.....	423
nicToDp	423
partClr	424
partNew	424
powerOff	425
raidList.....	425
reboot.....	425
reset	425
serial.....	427
setNTP	427
setSNMP	428
srvAlloc.....	428
srvFree.....	429
srvList	430
start	430
stop	430
unreg	431
upgrade	432
vgs	433
vmstat	433
/CONNECTIONS NODE.....	434
<i>API Messages</i>	434
closeAll	434
count	435
help	435
info	436
ls.....	436
netSpeed	437
/LOGS NODE	437

<i>API Messages</i>	437
count	437
download	438
help	439
info	440
ls	441
pull.....	441
timeRoll	442
/REST NODE.....	443
<i>API Messages</i>	443
count	443
help	443
info	444
ls	445
/SYS NODE	445
<i>API Messages</i>	445
caCert	445
count	446
fileEdit	446
help	447
info	448
ls	448
peerCert	449
save	449
servCert	450
shutdown	450
statHist	451
telemetry.....	452
/USERS NODE.....	453
<i>API Messages</i>	453
addOrMod	453
auths.....	456
count	456
delete	457
help	457
info	458
ls	458
unlock.....	459
whoAml	459
MANUALS	460
/DATABASE.....	460
/DATABASE DBSTATE.....	461
/DATABASE HASHINFO	461
/DATABASE MANIFEST	461
/DATABASE RECONFIG	462
/DATABASE SIZEROLL	463
/DATABASE STAGGER.....	464

/DATABASE TIMEROLL	466
/DECODER	467
/DECODER RESET	468
/DECODER SELECT	468
/DECODER SSLKEYS	468
<i>Premaster</i>	469
<i>Private Keys or PEM files</i>	469
<i>Key Management</i>	470
<i>Return Values</i>	470
<i>Viewing Unencrypted Traffic</i>	471
/DECODER START	471
/DECODER STOP	471
/DECODER WHOAGG	471
/SDK	471
/SDK CONTENT	473
Render Types.....	473
NetWitness Content Render Flags.....	475
<i>RESTful API</i>	477
/SDK HIERARCH	478
/SDK KEYREFS	478
/SDK LANGUAGE	478
<i>C API</i>	479
Language Flags.....	480
NetWitness Language Result Flags.....	480
/SDK MSEARCH	483
<i>msearch Flags</i>	485
<i>msearch Index Search Mode</i>	485
<i>Text Search Syntax</i>	485
<i>Search Syntax And Index Modes</i>	486
<i>msearch Tips</i>	487
/SDK PACKETS	487
<i>RESTful API</i>	487
<i>C API</i>	489
/SDK PIN	490
<i>Miscellaneous Operations</i>	491
<i>pinId</i>	491

/SDK QUERY	491
<i>where Clauses</i>	494
Query Operators	495
Text Values	497
IP Addresses	497
MAC Addresses	497
Numeric Values	497
Bucketed Numeric Indexes.....	497
Numeric Value Aliases	498
Date and Time Expressions	498
Relative Time Points.....	499
Special Range Values.....	499
group by Clause (since 10.5).....	500
order by Clause (since 10.5).....	502
<i>search parameter</i>	504
/SDK SEARCH	504
/SDK SESSION	504
/SDK SUMMARY	505
/SDK TIMELINE	506
/SDK VALIDATE	508
/SDK VALUES	508
<i>Parameters</i>	509
<i>values Flags</i>	511
<i>values Call Example</i>	512
<i>Values call and bucketing mode</i>	513
<i>Suggestion Mode</i>	513
<i>search parameter</i>	513
/SYS	514
/SYS CACERT	514
/SYS FILEEDIT	515
/SYS PEERCERT	516
/SYS SAVE	517
/SYS SERVCERT	517
/SYS SHUTDOWN	517
/SYS STATHIST	518
<i>Configuration of the Stats Database</i>	523
/DECODER/PARSERS	524
<i>Configuration</i>	524

Dynamic Header Thresholds	524
Header Prioritization	524
Dynamic Mapping	525
Receiving Protobuf Encoded Logs	525
<i>Messages</i>	526
/DECODER/PARSERS DYNDVMAP.....	526
Parameters and Results.....	526
<i>Display Mapped Sources</i>	526
<i>Purge Mappings</i>	527
<i>Force Persistence</i>	527
/DECODER/PARSERS TZINFO.....	527
Parameters and Results.....	527
<i>Version</i>	528
<i>Timezone Names</i>	528
<i>Timezone Abbreviations</i>	528
<i>Duplicates Abbreviations</i>	529
<i>Overridden Abbreviations</i>	529
<i>Add/Edit an Abbreviation Override</i>	530
<i>Remove an Abbreviation Override</i>	530
/SDK DEVICEID.....	530
/APPLIANCE.....	531
<i>Summary of Array Configuration Commands</i>	531
Commands that deal with direct-attach RAID volumes	531
Commands that deal with allocating block devices as storage	532
Commands that deal with allocating storage to services	532
/APPLIANCE DFLALLOC	532
SET UP STORAGE-RELATED DEFAULT CONFIGURATION FOR A SERVICE	532
HYBRID CONCENTRATOR ALLOCATION.....	533
HYBRID DECODER ALLOCATION	533
HYBRID LOG DECODER ALLOCATION.....	534
/APPLIANCE PARTCLR.....	534
UNINITIALIZE A BLOCK STORAGE DEVICE.....	534
/APPLIANCE PARTNEW	535
INITIALIZE A BLOCK STORAGE DEVICE	535
<i>Usage</i>	535
Example: Assign devices sdd and sde to Decoder	536
Example: Assign device sdc to Archiver.....	536
<i>Netwitness Service Volume Reference</i>	536
/APPLIANCE SRVALLOC	540

ALLOCATE STORAGE TO A SERVICE	540
DECODER, LOG DECODER, AND CONCENTRATOR ALLOCATION	541
ARCHIVER STORAGE ALLOCATION	541
HYBRID ALLOCATION	542
/APPLIANCE SRVFREE.....	542
REMOVE STORAGE FROM A SERVICE CONFIGURATION	542
/APPLIANCE VGS	542
LIST VOLUME GROUPS	542

Overview

Netwitness Core Services API can be accessed through the host 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 requests to the API will use latest version of API available with the service.

Netwitness-Version: 11.7.0

Schema

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

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

The API accepts input requests 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 fields containing human readable timestamps or dates would be in the format

YYYY-MM-DD HH:MM:SS

HTTP Usage

API tries to adhere as closely as possible to standard HTTP and REST conventions in its use of HTTP verbs and status 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

Status Code	Usage
200 OK	Request completed successfully.

Stauts Code	Usage
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 listen for REST requests.

Service	Port
Decoder	50104
LogDecoder	50102
Concentrator	50105
Broker	50103

Service	Port
Archiver	50108
Appliance	50106

Case Sensitive

All URLs, request parameters and JSON fields are case sensitive.

Error Response

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

Authentication and Authorization

All requests must 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 certificates.

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.rsa.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 requests through the API, users must belong to groups associated with roles that have the access permissions, as well as any underlying permissions required to fulfill the request. For details on service user roles and permissions refer <https://community.rsa.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 just time and no other fields 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 messages:
 - `statHist` to retrieve historical stats from stats database
 - `/sys/stats` node for service and system stats information

Sample Request

The following is a sample request to get list of capture adapters on the decoder using `select` message:

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

```
$ curl 'https://decoder-host: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-store, must-revalidate  
Content-Type: application/json
```

JSON Response

```
{  
  "flags": 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	string, optional, {enum-any:The value must be one or more of the following: native rest amqp}	Indicates the type of connection to close, either native, rest or amqp. Default is all types.
limit	uint32, optional	If non-zero, will stop closing connections after limit is reached. NOTE: It starts closing the most idle connections first.
idleLongerThan	string, 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

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

Parameter	Type, Options	Description
		friendly format

[info](#)

Description: Returns detailed information about the node

Security.roles: everyone

[ls](#)

Description: Returns the list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[netSpeed](#)

Description: Network Speed Test

Security.roles: connections.manage,aggregate

Parameters:

Parameter	Type, Options	Description
size	size, optional, {byte-range:The value (in bytes, size modifiers like MB are accepted) must be between 1024 and 536870912, inclusive}	The size of each message sent, default is 64k
count	uint32, optional, {unsigned:The value must 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 state or persists the current state to disk for fast reload

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: filelist summary save}	The operation to perform (filelist, summary or save)
type	string, {enum-any:The value must be one or more of the following: session meta packet}	The database(s) to operate on
options	string, optional, {enum-one:The value must 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 files

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
session	uint64	The session id to dump
type	string, optional, {enum-one:The value must be one of the following: db nwd}	The dump type

Parameter	Type, Options	Description
source	string, optional, {enum-any:The value must 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
file	string, optional	Optional filename to use for NWD type, otherwise filename is assumed to be <code>_sessionid_nwd</code>
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 files that containing session/meta/packet objects for a set of sessions or date range.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
sessions	string, optional	A comma delimited list of sessions and session ranges to retrieve file hashes for.

Parameter	Type, Options	Description
beginDate	string, optional	The beginning of a date range (YYYY-MM-DD HH:MM:SS) of database file creation dates to retrieve file hashes for.
endDate	string, optional	The end of a date range (YYYY-MM-DD HH:MM:SS) of database file creation dates to retrieve file hashes for.
directories	string, optional	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.

[Manual for hashInfo](#)

[help](#)

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

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are

Parameter	Type, Options	Description
		'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[manifest](#)

Description: 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.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must be one of the following: query compress}	The operation to perform (defaults to query)

Parameter	Type, Options	Description
time1	date-time, optional	The beginning time (UTC) for matching offline database files
time2	date-time, optional	The ending time (UTC) for matching offline database files
timeFormat	string, optional, {enum-one:The value must 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 tests to determine the optimal *.write.block.size based on configured drives and using data from existing databases.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
type	string, optional, {enum-any:The value must 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 modifiers like MB are accepted) must be between 104857600 and 107374182400, inclusive}	The amount of data to write for each test

Parameter	Type, Options	Description
compression	string, optional, {enum-one:The value must be one of the following: none gzip bzip2 lzma}	Determines if compression block sizes should be tested, default is none
compressionLevel	uint32, optional, {unsigned:The value must be between 0 and 9, inclusive}	Determines the compression level, 0-9, where 1 is fastest, 9 is the best compression and zero is a predetermined balance between speed and compression

[reconfig](#)

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 filesystem and will only be used for storage 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 just output the calculations
type	string, optional, {enum-any:The value must be one or more of the following: session meta packet}	The database types to reconfigure, default is all

Parameter	Type, Options	Description
percent	uint32, optional, {unsigned:The value must be between 1 and 99, inclusive}	The drive percentage to use for the calculated size per directory, default is 95
op	string, optional, {enum-one:The value must 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 stats or just the ones listed.

Security.roles: database.manage

Parameters:

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

[sizeRoll](#)

Description: 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.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
type	string, {enum-any:The value must be one or more of the following: session meta packet}	The databases to consider for removing the oldest 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, oldest data is deleted first 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, oldest data is deleted first until total size is less than maxSize. This applies only to the Warm tier.
maxPercent	string, optional	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	string, optional	The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When

Parameter	Type, Options	Description
		exceeded, oldest data is deleted first 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 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	size, optional	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

[Manual for sizeRoll](#)

stagger

Description: Staggers database files 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	string, {enum-one:The value must be one of the following: session meta packet}	The databases files to stagger
dryRun	bool, optional	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. Please turn off capture or aggregation before actually performing the stagger operation.

[Manual for stagger](#)

timeRoll

Description: Delete database files that exceed a given age

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
type	string, {enum-any:The value must be one or more of the following: session meta packet}	The database files to remove
timeCalc	string, optional, {enum-one:The value must be one of the following: current last-write}	The time calculation to use (current date or last write date), default is current

Parameter	Type, Options	Description
minutes	uint32, optional, {unsigned:The value must be between 0 and 10000, inclusive}	Remove database files older than the given number of minutes
hours	uint32, optional, {unsigned:The value must be between 0 and 10000, inclusive}	Remove database files older than the given number of hours
days	uint32, optional, {unsigned:The value must be between 0 and 10000, inclusive}	Remove database files older than the given number of days
date	string, optional	Remove database files 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

Parameter	Type, Options	Description
payloadOnly	bool, optional	If true (default), will only overwrite the packet payload
pattern	string, optional	The pattern to use, by default it uses all zeros
metaList	string, optional	Comma separated list of meta to wipe, default (empty) is all meta
source	string, optional, {enum-any:The value must be one or more of the following: m p}	The types of data to wipe, meta and/or packets, default is just packets

[/decoder node](#)

[Manual for /decoder](#)

[API Messages](#)

[count](#)

Description: Returns the number of child nodes

Security.roles: everyone

[dpmk](#)

Description: perform DPDK administration

Security.roles: decoder.manage

Parameters:

Parameter	Type, Options	Description
migrate	string, optional	perform migration of existing capture configuration to DPDK
commit	bool, optional	commit changes

[help](#)

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

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform

Parameter	Type, Options	Description
format	string, optional, {enum-one:The value must be one of the following: default xml html}	(e.g., op=manual would return a man page on this node or the specified message) 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[reconfig](#)

Description: Calculates optimal settings for decoder pools and buffers based on the installed hardware.

Security.roles: decoder.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional, {bool:The value must be one of the following acceptable boolean values: 0,1,yes,no,true,false,on,off}	If true (default is false), will automatically update the respective decoder settings, otherwise it will just output the calculations

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must 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 installed 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, manifests, stats, configuration, or logs for this service. Data automatically deletes index and stats, unless filesCreatedAfter is specified. Service is automatically restarted. Example arguments: data=1 config=1 log=1 This example will reset data, index, logs, and configuration index=1 This example will reset the index only manifest=1 This example will delete everything in the manifest directory filesCreatedAfter="2015-12-01 14:00:00" This example will delete all session, meta and packet files created on or after Dec 1st, 2015 2pm (UTC) from the service. All other files will remain. The index will not be touched, but upon restart will be truncated to match the last 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)

Parameter	Type, Options	Description
index	bool, optional	Reset index, only valid if data is not reset (may not be applicable to all services)
manifest	bool, optional	Reset all manifests stored in the long term manifest directory (may not be applicable to all services)
config	bool, optional	Reset configuration settings to default
stats	bool, optional	Reset the stats database
log	bool, optional	Reset the log database
filesCreatedAfter	date-time, optional	Delete all database files (session, meta, packets) created after a certain date. The index will automatically get rolled back on restart. Not valid with option index.
cl	string, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next restart

[Manual for reset](#)

[resetMax](#)

Description: Resets all max stats to zero.

Security.roles: decoder.manage

[select](#)

Description: Selected a new capture device

Security.roles: decoder.manage

Parameters:

Parameter	Type, Options	Description
adapter	string, optional	The new adapter. If omitted will display a list of available adapters. You may specify a comma-separated list 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 existing keys from storage. 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.

Parameter	Type, Options	Description
random	string, optional	Adds the random that identifies the session key exchange.
premaster	string, optional	Adds the encryption key for the random.
pemFilename	string, optional	Gives the name of a PEM file that is the binary part of the request. This should be a private key that can be used for decrypting SSL traffic.
listPems	bool, optional	Returns the list of all installed PEM files used for decryption.
deletePem	string, optional	Given the PEM name, deletes it from the service. Can be repeated.

[Manual for sslKeys](#)

start

Description: Starts aggregation

Security.roles: decoder.manage

[Manual for start](#)

stop

Description: Stops aggregation

Security.roles: decoder.manage

Parameters:

Parameter	Type, Options	Description
flush	bool, optional	If true (default), process all sessions and packets currently in memory. Otherwise, stop 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

content

Description: Get parsing content file information.

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
file	string, optional	The content filename (type must be specified)
type	string, optional, {enum-one:The value must be one of the following: feeds parsers}	The content type

[count](#)

Description: Returns the number of child nodes

Security.roles: everyone

[delete](#)

Description: Delete parsing content files.

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
file	string	The content filename
type	string, {enum-one:The value must be one of the following: feeds parsers}	The content type (feed parser)

[devices](#)

Description: Returns the log parsers

Security.roles: parsers.manage

[feed](#)

Description: Manages the feed parsers

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: notify reload remove delete}	The feed operation to perform
file	string, optional	The feed filename

[help](#)

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

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are

Parameter	Type, Options	Description
		'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[reload](#)

Description: Reloads all meta parsers

Security.roles: parsers.manage

[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
showRegisteredMeta	bool, optional	If true (the default), show the registered meta for each parser
showOptions	bool, optional	If true, shows the options supported by each parser. Passing true changes the XML structure 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 statistics

Security.roles: parsers.manage

[upload](#)

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

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: start attributes cancel finished}	The operation to perform (start, cancel, finished)
size	uint64, optional	The size of the incoming file
filename	string, optional	The name of the incoming file
reload	uint32, optional	Reload all parsers when upload is complete for the appropriate file types (default). A setting of 0 when sending a finished 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 `=` must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

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

Parameter	Type, Options	Description
		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 specific keys and values

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
key	string, optional	The key to inspect
value	string, optional	The value to

Parameter	Type, Options	Description
format	string, optional, {enum-one:The value must be one of the following: Text IPv4 Int8 UInt8 Int16 UInt16 Int32 UInt32 Int64 UInt64 Int128 Float32 Float64 TimeT DayOfWeek HourOfDay Binary IPv6 MAC}	inspect 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, instead of just most recent

Parameter	Type, Options	Description
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 defined in the language

[ls](#)

Description: Returns the list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[profile](#)

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

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
key	string, optional	The key to profile

Parameter	Type, Options	Description
value	string, optional	The value to profile
max	uint32, optional	Max number of pages to return
flag	uint32, optional	Bitwise option when key is specified. 1: Break report to values. 2: List page sizes. 4: Categorize page counts in different range. 8: Sessions count Per page with each algorithm. 16: Verify session page

[reconfig](#)

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

Security.roles: index.manage

Parameters:

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

save

Description: Saves the index to disk

Security.roles: index.manage

schema

Description: Introspects all open Meta DB files for meta keys and merges those keys with the defined language to return a fairly comprehensive list of all meta keys in existence.

Security.roles: index.manage

sizeRoll

Description: Delete index and other DB files (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 storage.

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	string, optional, {enum-any:The value must be one or more of the following: session meta packet}	The databases to consider for removing the oldest data based on total size or space remaining
maxSize	size, optional	The maximum size of the index hot storage. When exceeded, oldest data is deleted first until total size is less than maxSize.

Parameter	Type, Options	Description
maxSizeWarm	size, optional	The maximum size of the index warm storage. When exceeded, oldest data is delete first until total size is less that maxSizeWarm
maxPercent	string, optional	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 parameter applies to hot storage only.
maxPercentWarm	string, optional	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 parameter applies to warm storage only.
minFree	size, optional	The minimum allowed free space on the volumes before oldest data is deleted. This parameter applies to hot storage only.
minFreeWarm	size, optional	The minimum allowed free space on the volumes before oldest data is deleted. This parameter applies to warm storage only.

[values](#)

Description: Returns the top n values for the given key

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
key	string	The index key for values
top	uint32, {unsigned:The value must 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 starting log number to download

Parameter	Type, Options	Description
id2	uint64, optional	The ending log number to download
time1	date-time, optional	The starting 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	string, {enum-one:The value must be one of the following: start next cancel}	The operation to perform (start, next, cancel)
logTypes	string, optional	The type(s) of log messages to retrieve (debug,info,audit,warning,failure)
match	string, optional	Case insensitive string to match in each log message
regex	string, optional	Regular expression to match in each log message
timeFormat	string, optional, {enum-one:The value must be one of the following: posix simple}	The time format used in each log message, default is posix time (seconds since 1970)

Parameter	Type, Options	Description
batchSize	uint32, optional, {unsigned:The value must 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual

Parameter	Type, Options	Description
format	string, optional, {enum-one:The value must be one of the following: default xml html}	would return a man page on this node or the specified message) 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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 first log id number to retrieve, this is mutually exclusive with id2
id2	uint64, optional	The last log id number that will be sent, defaults to most recent log message when id1 or id2 is not sent

Parameter	Type, Options	Description
count	uint32, optional, {unsigned:The value must be between 1 and 10000, inclusive}	The number of logs to pull
timeFormat	string, optional, {enum-one:The value must 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	string, optional, {enum-any:The value must be one or more of the following: current last-write}	The time calculation to use (current date or last write date), default is current
minutes	uint32, optional, {unsigned:The value must be between 0 and 60, inclusive}	Remove log entries older than the given number of minutes
hours	uint32, optional, {unsigned:The value must be between 0 and 24, inclusive}	Remove log entries older than the given number of hours

Parameter	Type, Options	Description
days	uint32, optional, {unsigned:The value must be between 0 and 365, inclusive}	Remove log entries older than the given number of days
date	string, optional	Remove log entries older than the given date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

[/rest 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 `" "` must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters:	The name of the

Parameter	Type, Options	Description
	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[**/sdk node**](#)

[Manual for /sdk](#)

[**API Messages**](#)

[**aliases**](#)

Description: Retrieves aliases for the specified key

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
key	string	The language key to retrieve aliases for
value	string, optional	An optional value to return a specific 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	string, optional	The device UUID if taking a device offline instead of canceling the entire query (Broker only). Passing uuid will propagate the request upstream to find the first 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 first packet of the session, in case the session has rolled out
pinId	string, optional	The Pin ID of the session to retrieve from long term storage. If provided, the packet and session parameters must 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	string, optional	The render type (see NwContentTypes in NwSDK.h), pass zero to pick best option. Not specifying this parameter always results in an NWD and renderFlags and renderOptions are not considered. The following values are also supported: 'pcap', 'file-list-json', 'session-meta-file-list-json', 'nwd', 'log', 'files'
renderFlags	uint32, optional	Bitwise mask to control options, (see NwContentFlags in NwSDK.h)
renderOptions	string, optional	An encoded string 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 files from either the normal NWD cache or the pin cache

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
type	string, optional, {enum-one:The value must be one of the following: nwd pin all}	Specifies which cache to delete all NWD files 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters:	The name of the message to

Parameter	Type, Options	Description
	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 definition

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The starting language id
id2	uint64, optional	The ending language id
time1	date-time, optional	The starting time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS)
time2	date-time, optional	The ending time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS)

Parameter	Type, Options	Description
options	string, optional	Extra options, currently unused
flags	string, optional	Optional flags to configure 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 must be between 1 and 1677721, inclusive}	The max number of entries to return
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query

[Manual for language](#)

[ls](#)

Description: Returns the list 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

Parameter	Type, Options	Description
options	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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	string, optional	The session ID ranges to search
packets	string, optional	The packet ID ranges to search
search	string	The search string to use
where	string, optional	The where clause used to identify sessions to consider for the search

Parameter	Type, Options	Description
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
flags	string, {enum-any:The value must be one or more of the following: regex sp sm ci pre post ds precache si}	Flags to use for search. This is a comma separated list of one or more of the flag values: regex,sp,sm,ci,pre,post,ds,precache,si
queryPriority	int32, optional, {signed:The value must 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	string, {enum-one:The value must be one of the following: setup start cancel processed}	The operation to perform (start, cancel, processed)

Parameter	Type, Options	Description
sessions	string, optional	A comma separated list of session ids or session id ranges (#-#) whose packets will be streamed back.
packets	string, optional	A comma separated list of packet ids or session/packet ids (#&#) that will be streamed back.
where	string, optional	Where clause which will be evaluated to determine which sessions to stream back
time1	date-time, optional	A starting time range (UTC) where matching packets will be streamed back ("2010-Apr-20 09:00:00")
time2	date-time, optional	An ending time range (UTC) where matching packets will be streamed back ("2010-Apr-20 10:00:00")
flags	uint32, optional	Additional flags as defined by the NwPackets SDK function
queryPriority	int32, optional, {signed:The value must 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 must 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 status 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	string, optional	A comma separated list of session ids or session id ranges (#-#) to pin. This cannot be used with the 'session' parameter.
pinId	string, optional	One or more Pin IDs, separated by commas
op	string, optional, {enum-one:The value must be one of the following: ls unpin validate}	Various operations that can be performed. Will be submitted to upstream devices if not configured for long term cache.

[Manual for pin](#)

[precache](#)

Description: Efficiently caches NWD files for future retrieval via the content message

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
sessions	string, optional	A comma separated list of session ids or session id ranges (#-#) that will be saved in the NWD cache directory
packets	string, optional	A comma separated list 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 against the meta database

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The starting meta id (to run the query from most recent to oldest meta, make id1 larger than id2)
id2	uint64, optional	The ending meta id

Parameter	Type, Options	Description
size	uint32, optional	The max number of entries to return, or just stream back all results if zero
query	string, optional	The query string to use
flags	string, optional	The flags to use for query. Can be a number (bitwise mask) or comma separated values like query-log.
threshold	uint64, optional	Query optimization to stop 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. Must 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	string, optional	A text search clause to apply to the where clause of this query
streamLimit	uint32, optional	Limit the number of sessions returned by a streaming query
queryPriority	int32, optional, {signed:The value must	The query priority for this query

Parameter	Type, Options	Description
	be zero OR between -20 and 20, inclusive}	

[Manual for query](#)

[reconfig](#)

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

Security.roles: sdk.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the config nodes with the defaults
memory	size, optional	If non-zero, will use that setting as installed 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 must be between 1 and 1677721, inclusive}	The max number of entries to return
search	string	The search string 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 starting session id (inclusive). Pass zero to scope to lowest valid id.
id2	uint64	The ending session id (inclusive). Pass zero to scope to highest valid id.

[Manual for session](#)

summary

Description: Retrieves summary information from the databases

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
flags	string, optional	Optional SDK flags. 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 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
time2	date-time, optional	The ending time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS)
timezone	int32, optional	The timezone of the local computer to receive the data, hour offset from GMT [-13 to 13]

Parameter	Type, Options	Description
size	uint32, {unsigned:The value must be between 1 and 1677721, inclusive}	The max number of entries to return
flags	string, optional	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	string, optional	Optional where clause for filtering 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.
fieldName	string, optional	The time field to retrieve values for, defaults to 'time'
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query
search	string, 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	string, {enum-one:The value must 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 just syntax will be validated
fieldName	string, optional	Required if validating a 'values' operation. The field to retrieve values for.
where	string, optional	Required if validating a 'values' operation. The filter criteria to be validated.
query	string, optional	Required if validating a 'query' operation. The query string 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 starting meta id
id2	uint64, optional	The ending meta id
size	uint32, {unsigned:The value must be between 1 and 1677721, inclusive}	The max number of entries to return
flags	string, optional	The flags 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 stop processing large session counts
fieldName	string	The field to retrieve values for
where	string, optional	The filter criteria for the values

Parameter	Type, Options	Description
aggregateFunction	string, optional, {enum-one:The value must be one of the following: count sum}	The aggregate operation to be applied where the aggregate flag is set
aggregateFieldName	string, optional	The meta field to aggregate in the aggregateFunction
min	string, optional	The lower limit of values to return. Only values greater than this value will be returned.
max	string, optional	The upper limit of values to return. Only values less than this value will be returned.
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query
search	string, optional	A text search clause to apply to the where clause of this query

[Manual for values](#)

xforms

Description: Retrieves transforms for the specified key

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
key	string, 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 trusted CA certs

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: list delete add}	CA action to perform (list,delete,add)
id	string, optional	ID of CA certificate to remove when op=delete

[Manual for caCert](#)

[count](#)

Description: Returns the number of child nodes

Security.roles: everyone

[fileEdit](#)

Description: View and edit system configuration files

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: dir get getBackup put create delete}	The operation to perform (dir, get, getBackup, put, create, delete)
type	string, optional, {enum-one:The value must be one of the following: parser filter}	The type of file, if not provided will check file extension
filename	string, optional, {length:The value length must 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[peerCert](#)

Description: Display or modify trusted peer certificates

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must be one of the following: list delete add}	Operation to perform on the list of trusted peers (list,delete,add)
id	string, optional	ID of peer certificate to remove when op=delete

[Manual for peerCert](#)

save

Description: Forces a save of system settings

Security.roles: sys.manage

Parameters:

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

[Manual for save](#)

servCert

Description: Display the current server certificate 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	string, optional, {length:The value length must be between 0 and 512, inclusive}	The reason for shutting down the service
cl	string, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next restart

[Manual for shutdown](#)

statHist

Description: Retrieve historical stats from the stats db. Don't send time1/time2 to get bounding times about stats 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 starting time for retrieving stats
time2	date-time, optional	The ending time for retrieving stats

Parameter	Type, Options	Description
timeFormat	string, optional, {enum-one: The value must be one of the following: posix simple}	Specify the time format for each stat snapshot, default is posix (seconds since 1970)
include	string, optional	Comma separated list of stats to include (wildcards allowed)
exclude	string, optional	Comma separated list of stats to exclude (wildcards allowed, has precedence over include)
onChange	string, optional	Comma separated list of stats that reduce the result set to only when at least one of the stats change from their previous value
reduce	bool, optional	If true, reduces data transmission by replacing stat pathnames with a shorthand and provides a lookup table as first result
showAll	bool, optional	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.

[Manual for statHist](#)

[telemetry](#)

Description: Returns telemetry information about the service

Security.roles: sys.manage

Parameters:

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

[/users node](#)

[API Messages](#)

[addOrMod](#)

Description: Add a new user or update an existing user in the system

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ WXYZ0123456789@!#\$%&'+-=?^_`{ }~.[]\}	The username to add or update, must be alphanumeric or @!#\$%&' + - = ? ^ _ ` { } ~ . [] \
password	string, optional	The user's password
pwdIsHashed	bool, optional	Set to true if the password has already been hashed
groups	string, optional	The groups the user belongs to

Parameter	Type, Options	Description
authType	string, optional, {enum-one:The value must be one of the following: netwitness pam}	The authentication system to use, "netwitness" is the default
queryTimeout	uint32, optional, {unsigned:The value must 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	string, optional	Display name for this user account
email	string, optional	Email address

Parameter	Type, Options	Description
		for this account
description	string, optional	Description of this user account
queryPrefix	string, optional	Query filter applied to every query performed by this user account
sessionThreshold	uint32, optional, {unsigned:The value must 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 system

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789@!#\$%&'+-=?^_`{ }~.[]\}	The username to delete

[help](#)

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

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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	string	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 fields 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 messages:
 - `statHist` to retrieve historical stats from stats database.
 - `/sys/stats` node for service and system stats information

Sample Request

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

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

```
$ curl 'https://logdecoder-host: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-store, must-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 requests.

```
{
  "flags": 1073872897,
  "string": "<addressmap-schema>
    <DeviceEntry ipv4=\"X.X.X.X\" device=\"cisco meraki\" lastUpdated=\"1586804116\"
soft=\"false\"/>
    <DeviceEntry ipv6=\"::1\" device=\"cisco meraki\" lastUpdated=\"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	string, {collection-name:Collection names must be valid path names}	The name of the collection to be created.
path	string, 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 `" "` must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

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

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[mounts](#)

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

Security.roles: everyone

[/connections node](#)

[API Messages](#)

[closeAll](#)

Description: Closes all connections

Security.roles: connections.manage

Parameters:

Parameter	Type, Options	Description
type	string, optional, {enum-any:The value must be one or more of the following: native rest amqp}	Indicates the type of connection to close, either native, rest or amqp. Default is all types.
limit	uint32, optional	If non-zero, will stop closing connections after limit is reached. NOTE: It starts closing the most idle connections first.
idleLongerThan	string, 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[netSpeed](#)

Description: Network Speed Test

Security.roles: connections.manage,aggregate

Parameters:

Parameter	Type, Options	Description
size	size, optional, {byte-range:The value (in bytes, size modifiers like MB are accepted) must be between 1024 and 536870912, inclusive}	The size of each message sent, default is 64k
count	uint32, optional, {unsigned:The value must 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 state or persists the current state to disk for fast reload

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: filelist summary save}	The operation to perform (filelist, summary or save)
type	string, {enum-any:The value must be one or more of the following: session meta packet}	The database(s) to operate on
options	string, optional, {enum-one:The value must 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 files

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
session	uint64	The session id to dump
type	string, optional, {enum-one:The value must be one of the following: db nwd}	The dump type

Parameter	Type, Options	Description
source	string, optional, {enum-any:The value must 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
file	string, optional	Optional filename to use for NWD type, otherwise filename is assumed to be <code>_sessionid_nwd</code>
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 files that containing session/meta/packet objects for a set of sessions or date range.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
sessions	string, optional	A comma delimited list of sessions and session ranges to retrieve file hashes for.

Parameter	Type, Options	Description
beginDate	string, optional	The beginning of a date range (YYYY-MM-DD HH:MM:SS) of database file creation dates to retrieve file hashes for.
endDate	string, optional	The end of a date range (YYYY-MM-DD HH:MM:SS) of database file creation dates to retrieve file hashes for.
directories	string, optional	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.

[Manual for hashInfo](#)

[help](#)

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

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are

Parameter	Type, Options	Description
		'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[manifest](#)

Description: 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.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must be one of the following: query compress}	The operation to perform (defaults to query)

Parameter	Type, Options	Description
time1	date-time, optional	The beginning time (UTC) for matching offline database files
time2	date-time, optional	The ending time (UTC) for matching offline database files
timeFormat	string, optional, {enum-one:The value must 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 tests to determine the optimal *.write.block.size based on configured drives and using data from existing databases.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
type	string, optional, {enum-any:The value must 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 modifiers like MB are accepted) must be between 104857600 and 107374182400, inclusive}	The amount of data to write for each test

Parameter	Type, Options	Description
compression	string, optional, {enum-one:The value must be one of the following: none gzip bzip2 lzma}	Determines if compression block sizes should be tested, default is none
compressionLevel	uint32, optional, {unsigned:The value must be between 0 and 9, inclusive}	Determines the compression level, 0-9, where 1 is fastest, 9 is the best compression and zero is a predetermined balance between speed and compression

[reconfig](#)

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 filesystem and will only be used for storage 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 just output the calculations
type	string, optional, {enum-any:The value must be one or more of the following: session meta packet}	The database types to reconfigure, default is all

Parameter	Type, Options	Description
percent	uint32, optional, {unsigned:The value must be between 1 and 99, inclusive}	The drive percentage to use for the calculated size per directory, default is 95
op	string, optional, {enum-one:The value must 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 stats or just the ones listed.

Security.roles: database.manage

Parameters:

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

[sizeRoll](#)

Description: 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.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
type	string, {enum-any:The value must be one or more of the following: session meta packet}	The databases to consider for removing the oldest 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, oldest data is deleted first 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, oldest data is deleted first until total size is less than maxSize. This applies only to the Warm tier.
maxPercent	string, optional	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	string, optional	The maximum percentage of all the volumes of all databases passed in 'type' parameter combined. When

Parameter	Type, Options	Description
		exceeded, oldest data is deleted first 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 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	size, optional	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

[Manual for sizeRoll](#)

stagger

Description: Staggers database files 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	string, {enum-one:The value must be one of the following: session meta packet}	The databases files to stagger
dryRun	bool, optional	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. Please turn off capture or aggregation before actually performing the stagger operation.

[Manual for stagger](#)

timeRoll

Description: Delete database files that exceed a given age

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
type	string, {enum-any:The value must be one or more of the following: session meta packet}	The database files to remove
timeCalc	string, optional, {enum-one:The value must be one of the following: current last-write}	The time calculation to use (current date or last write date), default is current

Parameter	Type, Options	Description
minutes	uint32, optional, {unsigned:The value must be between 0 and 10000, inclusive}	Remove database files older than the given number of minutes
hours	uint32, optional, {unsigned:The value must be between 0 and 10000, inclusive}	Remove database files older than the given number of hours
days	uint32, optional, {unsigned:The value must be between 0 and 10000, inclusive}	Remove database files older than the given number of days
date	string, optional	Remove database files 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

Parameter	Type, Options	Description
payloadOnly	bool, optional	If true (default), will only overwrite the packet payload
pattern	string, optional	The pattern to use, by default it uses all zeros
metaList	string, optional	Comma separated list of meta to wipe, default (empty) is all meta
source	string, optional, {enum-any:The value must be one or more of the following: m p}	The types of data to wipe, meta and/or packets, default is just 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 last receive times for log devices, sources and forwarders.

Security.roles: decoder.manage

Parameters:

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must be one of the following: clear remove save buckets}	The log stats operation. If clear, all log stats state will be reset and no results will be returned. If save, the current stats are saved for the next time the log decoder is restarted.
deviceFilter	string, optional	A regex that will only return matching devices.
forwarderFilter	string, optional	A regex that will only return matching forwarders.
sourceFilter	string, optional	A regex that will only return matching sources.
discoveredFilter	string, optional	A regex that will only return matching discovered device types.
timeFilter	string, optional	This parameter forms a window from the current time which will

Parameter	Type, Options	Description
		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 just number of seconds). Not compatible with the lastReceived parameter.
lastReceived	string, optional	Compare the last received log time from each device with the passed in operator {
lastUpdated	string, optional	Compare the last updated log time from each device with the passed in operator {
timeFormat	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[reconfig](#)

Description: Calculates optimal settings for decoder pools and buffers based on the installed hardware.

Security.roles: decoder.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional, {bool:The value must be one of the following acceptable boolean values: 0,1,yes,no,true,false,on,off}	If true (default is false), will automatically update the respective decoder settings, otherwise it will just output the calculations

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must 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 installed 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 specified set of packet IDs

Security.roles: decoder.manage

Parameters:

Parameter	Type, Options	Description
packets	string	A comma separated list of packet ids or packet id ranges to reparse
collection	string, 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.

Parameter	Type, Options	Description
source	string, 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, manifests, stats, configuration, or logs for this service. Data automatically deletes index and stats, unless filesCreatedAfter is specified. Service is automatically restarted. Example arguments: data=1 config=1 log=1 This example will reset data, index, logs, and configuration index=1 This example will reset the index only manifest=1 This example will delete everything in the manifest directory filesCreatedAfter="2015-12-01 14:00:00" This example will delete all session, meta and packet files created on or after Dec 1st, 2015 2pm (UTC) from the service. All other files will remain. The index will not be touched, but upon restart will be truncated to match the last 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)
manifest	bool, optional	Reset all manifests stored in the long term manifest directory (may not be applicable to all services)

Parameter	Type, Options	Description
config	bool, optional	Reset configuration settings to default
stats	bool, optional	Reset the stats database
log	bool, optional	Reset the log database
filesCreatedAfter	date-time, optional	Delete all database files (session, meta, packets) created after a certain date. The index will automatically get rolled back on restart. Not valid with option index.
cl	string, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next restart

[Manual for reset](#)

[resetMax](#)

Description: Resets all max stats to zero.

Security.roles: decoder.manage

[select](#)

Description: Selected a new capture device

Security.roles: decoder.manage

Parameters:

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

[Manual for select](#)

start

Description: Starts aggregation

Security.roles: decoder.manage

[Manual for start](#)

stop

Description: Stops aggregation

Security.roles: decoder.manage

Parameters:

Parameter	Type, Options	Description
flush	bool, optional	If true (default), process all sessions and packets currently in memory. Otherwise, stop 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](#)

[content](#)

Description: Get parsing content file information.

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
file	string, optional	The content filename (type must be specified)
type	string, optional, {enum-one:The value must be one of the following: feeds parsers}	The content type

[count](#)

Description: Returns the number of child nodes

Security.roles: everyone

[delete](#)

Description: Delete parsing content files.

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
file	string	The content filename
type	string, {enum-one:The value must be one of the following: feeds parsers device}	The content type (feed parser device)

[devices](#)

Description: Returns the log parsers

Security.roles: parsers.manage

[dyndvmap](#)

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

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must be one of	The operation to perform. If describe (default), return a listing of mappings. If clear, the all mappings

Parameter	Type, Options	Description
	the following: describe clear save}	will be removed. If save, the mappings will be persisted.
deviceFilter	string, optional	A regex that will only return matching devices.
sourceFilter	string, optional	A regex that will only return matching sources.
forwarderFilter	string, optional	A regex that will only return matching forwarders.

[Manual for dyndvmap](#)

feed

Description: Manages the feed parsers

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: notify reload remove delete}	The feed operation to perform
file	string, optional	The feed filename

headers

Description: Return header information about particular devices.

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
devices	string, 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following:	The specific

Parameter	Type, Options	Description
	messages parameters description values roles extra manual }	help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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/host are prioritized in the order in which they are listed. Takes effect immediately.

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: add edit remove describe}	The operation to perform.
entries	string, 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	string, optional	Used with 'describe' to return the mapped devices that have been updated AFTER the time specified (eg. "2010-Apr-20 10:00:00" will return all elements updated after April 20, 2010 10:00).
sources	string, optional	Used with 'describe' to return the mapped devices of the specified 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 effect after parser reload.

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: add edit remove describe}	The operation to perform (edit describe).
entries	string, optional	The IP entries. StringParam in format of '+/-ip='
mdformat	string, optional, {enum-one:The value must 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 be 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 list 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

Parameter	Type, Options	Description
options	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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	string, optional	Use 'parse' to get detailed information about how parse rules are parsed, otherwise leave blank for default behavior.
devices	string, optional	A CSV of device names or '*'. If empty, a list of all devices that have at least one content node defined is returned. If '*', a list of content nodes for all devices. Otherwise, the content nodes of the requested devices.

Parameter	Type, Options	Description
stems	string, optional, {enum-any:The value must be one or more of the following: default custom tokens}	A CSV of source content filename stems to list. Default is all stems.
showDisabled	bool, optional	Include disabled rules. Default is false.
type	string, optional, {enum-one:The value must be one of the following: RULE DataType}	If not specified then returns all content types, otherwise the name of the content type requested.

[reload](#)

Description: Reloads all meta parsers

Security.roles: parsers.manage

[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

Parameter	Type, Options	Description
showRegisteredMeta	bool, optional	If true (the default), show the registered meta for each parser
showOptions	bool, optional	If true, shows the options supported by each parser. Passing true changes the XML structure returned so each parser (potentially) has
prettyPrint	bool, optional	If true, returns the XML formatted to be easier to read. Default is false.

[tzinfo](#)

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

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: version tznames abbreviations duplicates overrides add remove}	The operation to perform.
abbr	string, optional	A timezone abbreviation to add or remove to the overrides

Parameter	Type, Options	Description
		list. This parameter is required for the add and remove operations.
gmtoffset	string, optional	The offset from GMT of a timezone abbreviation to add to the override list. Format is [-+][HH][MM].
tzname	string, optional	A timezone name to override the default timezone for a duplicate abbreviation. The named timezone must exist and the abbreviation must be valid for that timezone

Parameter	Type, Options	Description
		(see tznames).

[Manual for tzinfo](#)

upload

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

Security.roles: parsers.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: start attributes cancel finished}	The operation to perform (start, cancel, finished)
size	uint64, optional	The size of the incoming file
filename	string, optional	The name of the incoming file
reload	uint32, optional	Reload all parsers when upload is complete for the appropriate file types (default). A setting of 0 when sending a finished 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following:	The specific help operation

Parameter	Type, Options	Description
	messages parameters description values roles extra manual }	to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 specific keys and values

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
key	string, optional	The key to inspect
value	string, optional	The value to inspect
format	string, optional, {enum-one:The value must 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

Parameter	Type, Options	Description
		information on all slices, instead of just most 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 defined in the language

[ls](#)

Description: Returns the list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[profile](#)

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

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
key	string, optional	The key to profile
value	string, optional	The value to profile
max	uint32, optional	Max number of pages to return
flag	uint32, optional	Bitwise option when key is specified. 1: Break report to values. 2: List page sizes. 4: Categorize page counts in different range. 8: Sessions count Per page with each algorithm. 16: Verify session page

[reconfig](#)

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

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the respective config node settings, otherwise it will just output the calculations

Parameter	Type, Options	Description
percent	uint32, optional, {unsigned:The value must be between 1 and 99, inclusive}	The drive percentage to use for the index database, default is 90

save

Description: Saves the index to disk

Security.roles: index.manage

schema

Description: Introspects all open Meta DB files for meta keys and merges those keys with the defined language to return a fairly comprehensive list of all meta keys in existence.

Security.roles: index.manage

sizeRoll

Description: Delete index and other DB files (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 storage.

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.

Parameter	Type, Options	Description
type	string, optional, {enum-any:The value must be one or more of the following: session meta packet}	The databases to consider for removing the oldest data based on total size or space remaining
maxSize	size, optional	The maximum size of the index hot storage. When exceeded, oldest data is deleted first until total size is less than maxSize.
maxSizeWarm	size, optional	The maximum size of the index warm storage. When exceeded, oldest data is delete first until total size is less that maxSizeWarm
maxPercent	string, optional	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 parameter applies to hot storage only.
maxPercentWarm	string, optional	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 parameter applies to warm storage only.

Parameter	Type, Options	Description
minFree	size, optional	The minimum allowed free space on the volumes before oldest data is deleted. This parameter applies to hot storage only.
minFreeWarm	size, optional	The minimum allowed free space on the volumes before oldest data is deleted. This parameter applies to warm storage only.

values

Description: Returns the top n values for the given key

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
key	string	The index key for values
top	uint32, {unsigned:The value must 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 starting log number to download
id2	uint64, optional	The ending log number to download
time1	date-time, optional	The starting 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	string, {enum-one:The value must be one of the following: start next cancel}	The operation to perform (start, next, cancel)
logTypes	string, optional	The type(s) of log messages to retrieve (debug,info,audit,warning,failure)

Parameter	Type, Options	Description
match	string, optional	Case insensitive string to match in each log message
regex	string, optional	Regular expression to match in each log message
timeFormat	string, optional, {enum-one:The value must 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 must 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters:	The name of the message to retrieve

Parameter	Type, Options	Description
	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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 first log id number to retrieve, this is mutually exclusive with id2

Parameter	Type, Options	Description
id2	uint64, optional	The last log id number that will be sent, defaults to most recent log message when id1 or id2 is not sent
count	uint32, optional, {unsigned:The value must be between 1 and 10000, inclusive}	The number of logs to pull
timeFormat	string, optional, {enum-one:The value must 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	string, optional, {enum-any:The value must be one or more of the following: current last-write}	The time calculation to use (current date or last write date), default is current
minutes	uint32, optional, {unsigned:The value must be between 0 and 60, inclusive}	Remove log entries older than the given number of minutes

Parameter	Type, Options	Description
hours	uint32, optional, {unsigned:The value must be between 0 and 24, inclusive}	Remove log entries older than the given number of hours
days	uint32, optional, {unsigned:The value must be between 0 and 365, inclusive}	Remove log entries older than the given number of days
date	string, optional	Remove log entries older than the given date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

[/rest 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[/sdk node](#)

[Manual for /sdk](#)

[API Messages](#)

[aliases](#)

Description: Retrieves aliases for the specified key

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
key	string	The language key to retrieve aliases for
value	string, optional	An optional value to return a specific 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	string, optional	The device UUID if taking a device offline instead of canceling the entire query (Broker only). Passing uuid will propagate the request upstream to find the first 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 first packet of the session, in case the session has rolled out
pinId	string, optional	The Pin ID of the session to retrieve from long term storage. If provided, the packet and session parameters must 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	string, optional	The render type (see NwContentTypes in NwSDK.h), pass zero to pick best option. Not specifying this parameter always results in an NWD and renderFlags and renderOptions are not considered. The following values are also supported: 'pcap', 'file-list-json', 'session-meta-file-list-json', 'nwd', 'log', 'files'
renderFlags	uint32, optional	Bitwise mask to control options, (see NwContentFlags in NwSDK.h)
renderOptions	string, optional	An encoded string 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 files from either the normal NWD cache or the pin cache

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
type	string, optional, {enum-one:The value must be one of the following: nwd pin all}	Specifies which cache to delete all NWD files 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters:	The name of the message to

Parameter	Type, Options	Description
	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 definition

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The starting language id
id2	uint64, optional	The ending language id
time1	date-time, optional	The starting time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS)
time2	date-time, optional	The ending time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS)

Parameter	Type, Options	Description
options	string, optional	Extra options, currently unused
flags	string, optional	Optional flags to configure 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 must be between 1 and 1677721, inclusive}	The max number of entries to return
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query

[Manual for language](#)

[ls](#)

Description: Returns the list 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

Parameter	Type, Options	Description
options	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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	string, optional	The session ID ranges to search
packets	string, optional	The packet ID ranges to search
search	string	The search string to use
where	string, optional	The where clause used to identify sessions to consider for the search

Parameter	Type, Options	Description
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
flags	string, {enum-any:The value must be one or more of the following: regex sp sm ci pre post ds precache si}	Flags to use for search. This is a comma separated list of one or more of the flag values: regex,sp,sm,ci,pre,post,ds,precache,si
queryPriority	int32, optional, {signed:The value must 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	string, {enum-one:The value must be one of the following: setup start cancel processed}	The operation to perform (start, cancel, processed)

Parameter	Type, Options	Description
sessions	string, optional	A comma separated list of session ids or session id ranges (#-#) whose packets will be streamed back.
packets	string, optional	A comma separated list of packet ids or session/packet ids (#&#) that will be streamed back.
where	string, optional	Where clause which will be evaluated to determine which sessions to stream back
time1	date-time, optional	A starting time range (UTC) where matching packets will be streamed back ("2010-Apr-20 09:00:00")
time2	date-time, optional	An ending time range (UTC) where matching packets will be streamed back ("2010-Apr-20 10:00:00")
flags	uint32, optional	Additional flags as defined by the NwPackets SDK function
queryPriority	int32, optional, {signed:The value must 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 must 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 status 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	string, optional	A comma separated list of session ids or session id ranges (#-#) to pin. This cannot be used with the 'session' parameter.
pinId	string, optional	One or more Pin IDs, separated by commas
op	string, optional, {enum-one:The value must be one of the following: ls unpin validate}	Various operations that can be performed. Will be submitted to upstream devices if not configured for long term cache.

[Manual for pin](#)

[precache](#)

Description: Efficiently caches NWD files for future retrieval via the content message

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
sessions	string, optional	A comma separated list of session ids or session id ranges (#-#) that will be saved in the NWD cache directory
packets	string, optional	A comma separated list 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 against the meta database

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The starting meta id (to run the query from most recent to oldest meta, make id1 larger than id2)
id2	uint64, optional	The ending meta id

Parameter	Type, Options	Description
size	uint32, optional	The max number of entries to return, or just stream back all results if zero
query	string, optional	The query string to use
flags	string, optional	The flags to use for query. Can be a number (bitwise mask) or comma separated values like query-log.
threshold	uint64, optional	Query optimization to stop 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. Must 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	string, optional	A text search clause to apply to the where clause of this query
streamLimit	uint32, optional	Limit the number of sessions returned by a streaming query
queryPriority	int32, optional, {signed:The value must	The query priority for this query

Parameter	Type, Options	Description
	be zero OR between -20 and 20, inclusive}	

[Manual for query](#)

[reconfig](#)

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

Security.roles: sdk.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the config nodes with the defaults
memory	size, optional	If non-zero, will use that setting as installed 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 must be between 1 and 1677721, inclusive}	The max number of entries to return
search	string	The search string 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 starting session id (inclusive). Pass zero to scope to lowest valid id.
id2	uint64	The ending session id (inclusive). Pass zero to scope to highest valid id.

[Manual for session](#)

summary

Description: Retrieves summary information from the databases

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
flags	string, optional	Optional SDK flags. 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 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
time2	date-time, optional	The ending time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS)
timezone	int32, optional	The timezone of the local computer to receive the data, hour offset from GMT [-13 to 13]

Parameter	Type, Options	Description
size	uint32, {unsigned:The value must be between 1 and 1677721, inclusive}	The max number of entries to return
flags	string, optional	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	string, optional	Optional where clause for filtering 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.
fieldName	string, optional	The time field to retrieve values for, defaults to 'time'
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query
search	string, 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	string, {enum-one:The value must 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 just syntax will be validated
fieldName	string, optional	Required if validating a 'values' operation. The field to retrieve values for.
where	string, optional	Required if validating a 'values' operation. The filter criteria to be validated.
query	string, optional	Required if validating a 'query' operation. The query string 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 starting meta id
id2	uint64, optional	The ending meta id
size	uint32, {unsigned:The value must be between 1 and 1677721, inclusive}	The max number of entries to return
flags	string, optional	The flags 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 stop processing large session counts
fieldName	string	The field to retrieve values for
where	string, optional	The filter criteria for the values

Parameter	Type, Options	Description
aggregateFunction	string, optional, {enum-one:The value must be one of the following: count sum}	The aggregate operation to be applied where the aggregate flag is set
aggregateFieldName	string, optional	The meta field to aggregate in the aggregateFunction
min	string, optional	The lower limit of values to return. Only values greater than this value will be returned.
max	string, optional	The upper limit of values to return. Only values less than this value will be returned.
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query
search	string, optional	A text search clause to apply to the where clause of this query

[Manual for values](#)

xforms

Description: Retrieves transforms for the specified key

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
key	string, 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 trusted CA certs

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: list delete add}	CA action to perform (list,delete,add)
id	string, optional	ID of CA certificate to remove when op=delete

[Manual for caCert](#)

[count](#)

Description: Returns the number of child nodes

Security.roles: everyone

[fileEdit](#)

Description: View and edit system configuration files

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: dir get getBackup put create delete}	The operation to perform (dir, get, getBackup, put, create, delete)
type	string, optional, {enum-one:The value must be one of the following: parser filter}	The type of file, if not provided will check file extension
filename	string, optional, {length:The value length must 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[peerCert](#)

Description: Display or modify trusted peer certificates

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must be one of the following: list delete add}	Operation to perform on the list of trusted peers (list,delete,add)
id	string, optional	ID of peer certificate to remove when op=delete

[Manual for peerCert](#)

save

Description: Forces a save of system settings

Security.roles: sys.manage

Parameters:

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

[Manual for save](#)

servCert

Description: Display the current server certificate 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	string, optional, {length:The value length must be between 0 and 512, inclusive}	The reason for shutting down the service
cl	string, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next restart

[Manual for shutdown](#)

[statHist](#)

Description: Retrieve historical stats from the stats db. Don't send time1/time2 to get bounding times about stats 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 starting time for retrieving stats
time2	date-time, optional	The ending time for retrieving stats

Parameter	Type, Options	Description
timeFormat	string, optional, {enum-one: The value must be one of the following: posix simple}	Specify the time format for each stat snapshot, default is posix (seconds since 1970)
include	string, optional	Comma separated list of stats to include (wildcards allowed)
exclude	string, optional	Comma separated list of stats to exclude (wildcards allowed, has precedence over include)
onChange	string, optional	Comma separated list of stats that reduce the result set to only when at least one of the stats change from their previous value
reduce	bool, optional	If true, reduces data transmission by replacing stat pathnames with a shorthand and provides a lookup table as first result
showAll	bool, optional	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.

[Manual for statHist](#)

[telemetry](#)

Description: Returns telemetry information about the service

Security.roles: sys.manage

Parameters:

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

[/users node](#)

[API Messages](#)

[addOrMod](#)

Description: Add a new user or update an existing user in the system

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789@!#\$%&'+-=?^_`{ }~.[]\}	The username to add or update, must be alphanumeric or @!#\$%&' + - = ? ^ _ ` { } ~ . [] \
password	string, optional	The user's password
pwdIsHashed	bool, optional	Set to true if the password has already been hashed
groups	string, optional	The groups the user belongs to

Parameter	Type, Options	Description
authType	string, optional, {enum-one:The value must be one of the following: netwitness pam}	The authentication system to use, "netwitness" is the default
queryTimeout	uint32, optional, {unsigned:The value must 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	string, optional	Display name for this user account
email	string, optional	Email address

Parameter	Type, Options	Description
		for this account
description	string, optional	Description of this user account
queryPrefix	string, optional	Query filter applied to every query performed by this user account
sessionThreshold	uint32, optional, {unsigned:The value must 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 system

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789@!#\$%&'+-=?^_`{ }~.[]\}	The username to delete

[help](#)

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

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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	string	The username or account to unlock

[whoAmI](#)

Description: Returns information about the authenticated user

Security.roles: everyone

[Concentrator service](#)

Introduction

Concentrator stores and indexes metadata for fast 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 requested from a Concentrator, the request 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 stores the data. It simply knows where the data is kept, requests the data on the client's behalf and streams 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

- o `/sys/stats` node for service and system stats information

Sample Request

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

Request path is `/sdk` and URL request 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 request timeout and continue with user profile timeout.

Using URL encoded request

```
$ curl "https://concentrator-  
host:50105/sdk?msg=query&query=select%20count(sessionid)%20where%20did%3D%2  
7decoder-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-string-params`

```
$ curl 'https://concentrator-host:50105/sdk?msg=query&expiry=0' \  
-H 'Accept: application/json;charset=UTF-8' \  
-H 'Content-Type: application/x-netwitness-string-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-store, must-revalidate  
Content-Type: application/json
```

JSON Response

```
{
  "flags": 1074200577,
  "results": {
    "id1": 3216363401111,
    "id2": 3216363401110,
    "fields": [
      {
        "id1": 0,
        "id2": 0,
        "count": 6864785,
        "format": 8,
        "type": "sessionid",
        "flags": 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	string, optional, {ip-address:The value must 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 must be set
ip	string, optional	The IP address of the device, either this or device must be set
port	uint32, optional, {unsigned:The value must be between 0 and 65535, inclusive}	The port number the device is listening on
username	string	The username to run as
password	string, optional	The account password. Assumes service trust aggregation if no password is provided.
options	string, optional	The device options
ssl	string, optional	If true, the device requires SSL to connect
backup	string, optional	If true, the device will be used as a backup

Parameter	Type, Options	Description
reconnect	string, optional	If true, will automatically reconnect when not connected
name	string, optional	Device collection name if device is a collection

[count](#)

Description: Returns the number of child nodes

Security.roles: everyone

[delete](#)

Description: Delete an existing aggregation device

Security.roles: concentrator.manage

Parameters:

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

[edit](#)

Description: Edit an existing aggregation device

Security.roles: concentrator.manage

Parameters:

Parameter	Type, Options	Description
device	string, {ip-address:The value must be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to edit
username	string, optional	The username to run as
password	string, optional	The account password. For service trust aggregation you must not provide this parameter
options	string, optional	The device options
ssl	string, optional	If true, the device requires SSL to connect
backup	string, optional	If true, the device will be used as a backup
reconnect	string, optional	If true, will automatically reconnect when not connected
name	string, optional	Device collection name if device is a collection

Parameter	Type, Options	Description
switchToTrustedAuth	bool, optional	If true, removes any password associated with this account and will use trusted authentication henceforth. When switching to trusted, only the 'device' parameter is allowed.

[help](#)

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

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform

Parameter	Type, Options	Description
format	string, optional, {enum-one:The value must be one of the following: default xml html}	(e.g., op>manual would return a man page on this node or the specified message) 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[reconfig](#)

Description: Calculates optimal settings for concentrator pools and buffers based on the installed 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 just output the calculations
memory	size, optional	If non-zero, will use that setting as installed physical RAM (e.g., memory="24 GiB"). This is useful for hybrids and/or all-

Parameter	Type, Options	Description
		in-one machines where the service will need to share memory.

[reset](#)

Description: Reset data, index, manifests, stats, configuration, or logs for this service. Data automatically deletes index and stats, unless filesCreatedAfter is specified. Service is automatically restarted. Example arguments: data=1 config=1 log=1 This example will reset data, index, logs, and configuration index=1 This example will reset the index only manifest=1 This example will delete everything in the manifest directory filesCreatedAfter="2015-12-01 14:00:00" This example will delete all session, meta and packet files created on or after Dec 1st, 2015 2pm (UTC) from the service. All other files will remain. The index will not be touched, but upon restart will be truncated to match the last 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)
manifest	bool, optional	Reset all manifests stored in the long term manifest directory (may not be applicable to all services)

Parameter	Type, Options	Description
config	bool, optional	Reset configuration settings to default
stats	bool, optional	Reset the stats database
log	bool, optional	Reset the log database
filesCreatedAfter	date-time, optional	Delete all database files (session, meta, packets) created after a certain date. The index will automatically get rolled back on restart. Not valid with option index.
cl	string, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next restart

[resetMax](#)

Description: Resets all max stats, including device max stats back to zero.

Security.roles: concentrator.manage

[start](#)

Description: Starts aggregation

Security.roles: concentrator.manage

[status](#)

Description: Change the online/offline status of an aggregation device

Security.roles: concentrator.manage

Parameters:

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

[stop](#)

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	string, optional, {enum-any:The value must be one or more of the following: native rest amqp}	Indicates the type of connection to close, either native, rest or amqp. Default is all types.
limit	uint32, optional	If non-zero, will stop closing connections after limit is reached. NOTE: It starts closing the most idle connections first.
idleLongerThan	string, 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 "" must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the

Parameter	Type, Options	Description
		specified message)
format	string, optional, {enum-one:The value must 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 list 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

Parameter	Type, Options	Description
options	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[netSpeed](#)

Description: Network Speed Test

Security.roles: connections.manage,aggregate

Parameters:

Parameter	Type, Options	Description
size	size, optional, {byte-range:The value (in bytes, size modifiers like MB are accepted) must be between 1024 and 536870912, inclusive}	The size of each message sent, default is 64k
count	uint32, optional, {unsigned:The value must 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 state or persists the current state to disk for fast reload

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: filelist summary save}	The operation to perform (filelist, summary or save)
type	string, {enum-any:The value must be one or more of the following: session meta packet}	The database(s) to operate on
options	string, optional, {enum-one:The value must 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 files

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
session	uint64	The session id to dump
type	string, optional, {enum-one:The value must be one of the following: db nwd}	The dump type
source	string, optional, {enum-any:The value must 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
file	string, optional	Optional filename to use for NWD type, otherwise filename 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 files that containing session/meta/packet objects for a set of sessions or date range.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
sessions	string, optional	A comma delimited list of sessions and session ranges to retrieve file hashes for.
beginDate	string, optional	The beginning of a date range (YYYY-MM-DD HH:MM:SS) of database file creation dates to retrieve file hashes for.
endDate	string, optional	The end of a date range (YYYY-MM-DD HH:MM:SS) of database file creation dates to retrieve file hashes for.
directories	string, optional	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.

[Manual for hashInfo](#)

[help](#)

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

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters:	The name of the message to retrieve

Parameter	Type, Options	Description
	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[manifest](#)

Description: 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.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must be one of the following: query compress}	The operation to perform (defaults to query)
time1	date-time, optional	The beginning time (UTC) for matching offline database files
time2	date-time, optional	The ending time (UTC) for matching offline database files
timeFormat	string, optional, {enum-one:The value must 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 tests to determine the optimal `*.write.block.size` based on configured drives and using data from existing databases.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
type	string, optional, {enum-any:The value must be one or more of the following: session meta}	The database types to optimize, default is all

Parameter	Type, Options	Description
size	size, optional, {byte-range:The value (in bytes, size modifiers like MB are accepted) must be between 104857600 and 107374182400, inclusive}	The amount of data to write for each test
compression	string, optional, {enum-one:The value must be one of the following: none gzip bzip2 lzma}	Determines if compression block sizes should be tested, default is none
compressionLevel	uint32, optional, {unsigned:The value must be between 0 and 9, inclusive}	Determines the compression level, 0-9, where 1 is fastest, 9 is the best compression and zero is a predetermined balance between speed and compression

[reconfig](#)

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 filesystem and will only be used for storage 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 just output the calculations

Parameter	Type, Options	Description
type	string, optional, {enum-any:The value must be one or more of the following: session meta}	The database types to reconfigure, default is all
percent	uint32, optional, {unsigned:The value must be between 1 and 99, inclusive}	The drive percentage to use for the calculated size per directory, default is 95
op	string, optional, {enum-one:The value must 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 stats or just the ones listed.

Security.roles: database.manage

Parameters:

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

[sizeRoll](#)

Description: 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.

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
type	string, {enum-any:The value must be one or more of the following: session meta}	The databases to consider for removing the oldest 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, oldest data is deleted first 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, oldest data is deleted first until total size is less than maxSize. This applies only to the Warm tier.
maxPercent	string, optional	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.

Parameter	Type, Options	Description
maxPercentWarm	string, optional	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	size, optional	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	size, optional	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

[Manual for sizeRoll](#)

stagger

Description: Staggers database files 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	string, {enum-one:The value must be one of the following: session meta}	The databases files to stagger
dryRun	bool, optional	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. Please turn off capture or aggregation before actually performing the stagger operation.

[Manual for stagger](#)

timeRoll

Description: Delete database files that exceed a given age

Security.roles: database.manage

Parameters:

Parameter	Type, Options	Description
type	string, {enum-any:The value must be one or more of the following: session meta}	The database files to remove
timeCalc	string, optional, {enum-one:The value must be one of the following: current last-write}	The time calculation to use (current date or last write date), default is current

Parameter	Type, Options	Description
minutes	uint32, optional, {unsigned:The value must be between 0 and 10000, inclusive}	Remove database files older than the given number of minutes
hours	uint32, optional, {unsigned:The value must be between 0 and 10000, inclusive}	Remove database files older than the given number of hours
days	uint32, optional, {unsigned:The value must be between 0 and 10000, inclusive}	Remove database files older than the given number of days
date	string, optional	Remove database files 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	string, optional	The pattern to use, by default it uses all zeros
metaList	string, optional	Comma separated list of meta to wipe, default (empty) is all meta
source	string, optional, {enum-any:The value must be one or more of the following: m p}	The types of data to wipe, meta and/or packets, default is just 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 `""` must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

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

Parameter	Type, Options	Description
		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 specific keys and values

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
key	string, optional	The key to inspect
value	string, optional	The value to

Parameter	Type, Options	Description
format	string, optional, {enum-one:The value must be one of the following: Text IPv4 Int8 UInt8 Int16 UInt16 Int32 UInt32 Int64 UInt64 Int128 Float32 Float64 TimeT DayOfWeek HourOfDay Binary IPv6 MAC}	inspect 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, instead of just most recent

Parameter	Type, Options	Description
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 defined in the language

[ls](#)

Description: Returns the list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[profile](#)

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

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
key	string, optional	The key to profile

Parameter	Type, Options	Description
value	string, optional	The value to profile
max	uint32, optional	Max number of pages to return
flag	uint32, optional	Bitwise option when key is specified. 1: Break report to values. 2: List page sizes. 4: Categorize page counts in different range. 8: Sessions count Per page with each algorithm. 16: Verify session page

[reconfig](#)

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

Security.roles: index.manage

Parameters:

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

save

Description: Saves the index to disk

Security.roles: index.manage

schema

Description: Introspects all open Meta DB files for meta keys and merges those keys with the defined language to return a fairly comprehensive list of all meta keys in existence.

Security.roles: index.manage

sizeRoll

Description: Delete index and other DB files (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 storage.

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	string, optional, {enum-any:The value must be one or more of the following: session meta}	The databases to consider for removing the oldest data based on total size or space remaining
maxSize	size, optional	The maximum size of the index hot storage. When exceeded, oldest data is deleted first until total size is less than maxSize.

Parameter	Type, Options	Description
maxSizeWarm	size, optional	The maximum size of the index warm storage. When exceeded, oldest data is delete first until total size is less that maxSizeWarm
maxPercent	string, optional	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 parameter applies to hot storage only.
maxPercentWarm	string, optional	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 parameter applies to warm storage only.
minFree	size, optional	The minimum allowed free space on the volumes before oldest data is deleted. This parameter applies to hot storage only.
minFreeWarm	size, optional	The minimum allowed free space on the volumes before oldest data is deleted. This parameter applies to warm storage only.

values

Description: Returns the top n values for the given key

Security.roles: index.manage

Parameters:

Parameter	Type, Options	Description
key	string	The index key for values
top	uint32, {unsigned:The value must 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 starting log number to download
id2	uint64, optional	The ending log number to download

Parameter	Type, Options	Description
time1	date-time, optional	The starting 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	string, {enum-one:The value must be one of the following: start next cancel}	The operation to perform (start, next, cancel)
logTypes	string, optional	The type(s) of log messages to retrieve (debug,info,audit,warning,failure)
match	string, optional	Case insensitive string to match in each log message
regex	string, optional	Regular expression to match in each log message
timeFormat	string, optional, {enum-one:The value must 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 must	The number of logs to retrieve per batch, default is 2000

Parameter	Type, Options	Description
	be between 2000 and 10000, inclusive}	

[help](#)

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

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page)

Parameter	Type, Options	Description
		on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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

Parameter	Type, Options	Description
options	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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 first log id number to retrieve, this is mutually exclusive with id2
id2	uint64, optional	The last log id number that will be sent, defaults to most recent log message when id1 or id2 is not sent
count	uint32, optional, {unsigned:The value must be between 1 and 10000, inclusive}	The number of logs to pull

Parameter	Type, Options	Description
timeFormat	string, optional, {enum-one:The value must 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	string, optional, {enum-any:The value must be one or more of the following: current last-write}	The time calculation to use (current date or last write date), default is current
minutes	uint32, optional, {unsigned:The value must be between 0 and 60, inclusive}	Remove log entries older than the given number of minutes
hours	uint32, optional, {unsigned:The value must be between 0 and 24, inclusive}	Remove log entries older than the given number of hours
days	uint32, optional, {unsigned:The value must be between 0 and 365, inclusive}	Remove log entries older than the given number of days

Parameter	Type, Options	Description
date	string, optional	Remove log entries older than the given date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

[/rest 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are

Parameter	Type, Options	Description
		'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[/sdk node](#)

[Manual for /sdk](#)

[API Messages](#)

[aliases](#)

Description: Retrieves aliases for the specified key

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
key	string	The language key to retrieve aliases for
value	string, optional	An optional value to return a specific 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	string, optional	The device UUID if taking a device offline instead of canceling the entire query (Broker only). Passing uuid will propagate the request upstream to find the first 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 first packet of the session, in case the session has rolled out
pinId	string, optional	The Pin ID of the session to retrieve from long term storage. If provided, the packet and session parameters must 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	string, optional	The render type (see NwContentTypes in NwSDK.h), pass zero to pick best option. Not specifying this parameter always results in an NWD and renderFlags and renderOptions are not considered. The following values are also supported: 'pcap', 'file-list-json', 'session-meta-file-list-json', 'nwd', 'log', 'files'
renderFlags	uint32, optional	Bitwise mask to control options, (see NwContentFlags in NwSDK.h)
renderOptions	string, optional	An encoded string 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 files from either the normal NWD cache or the pin cache

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
type	string, optional, {enum-one:The value must be one of the following: nwd pin all}	Specifies which cache to delete all NWD files 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 `" "` must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)

Parameter	Type, Options	Description
format	string, optional, {enum-one:The value must 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 definition

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The starting language id
id2	uint64, optional	The ending language id
time1	date-time, optional	The starting time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS)
time2	date-time, optional	The ending time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS)
options	string, optional	Extra options, currently unused
flags	string, optional	Optional flags to configure 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 must be between 1 and 1677721, inclusive}	The max number of entries to return
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query

[Manual for language](#)

[ls](#)

Description: Returns the list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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	string, optional	The session ID ranges to search
packets	string, optional	The packet ID ranges to search

Parameter	Type, Options	Description
search	string	The search string to use
where	string, 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
flags	string, {enum-any:The value must be one or more of the following: regex sp sm ci pre post ds precache si}	Flags to use for search. This is a comma separated list of one or more of the flag values: regex,sp,sm,ci,pre,post,ds,precache,si
queryPriority	int32, optional, {signed:The value must 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	string, {enum-one:The value must be one of the following: setup start cancel processed}	The operation to perform (start, cancel, processed)
sessions	string, optional	A comma separated list of session ids or session id ranges (#-#) whose packets will be streamed back.
packets	string, optional	A comma separated list of packet ids or session/packet ids (#&#) that will be streamed back.
where	string, optional	Where clause which will be evaluated to determine which sessions to stream back
time1	date-time, optional	A starting time range (UTC) where matching packets will be streamed back ("2010-Apr-20 09:00:00")
time2	date-time, optional	An ending time range (UTC) where matching packets will be streamed back ("2010-Apr-20 10:00:00")
flags	uint32, optional	Additional flags as defined by the NwPackets SDK function

Parameter	Type, Options	Description
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query
devices	string, optional	Optional list of comma separated device names that will be queried. If this parameter is not present, then all devices are queried. The device names must match the list defined under /concentrator/devices
excludeDevices	string, optional	Optional list of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs must match the UUID defined under /sys/stats/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 must 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 status 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	string, optional	A comma separated list of session ids or session id ranges (#-#) to pin. This cannot be used with the 'session' parameter.
pinId	string, optional	One or more Pin IDs, separated by commas
op	string, optional, {enum-one:The value must be one of the following: ls unpin validate}	Various operations that can be performed. Will be submitted to upstream devices if not configured for long term cache.

[Manual for pin](#)

[precache](#)

Description: Efficiently caches NWD files for future retrieval via the content message

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
sessions	string, optional	A comma separated list of session ids or session id ranges (#-#) that will be saved in the NWD cache directory

Parameter	Type, Options	Description
packets	string, optional	A comma separated list 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 against the meta database

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The starting meta id (to run the query from most recent to oldest meta, make id1 larger than id2)
id2	uint64, optional	The ending meta id
size	uint32, optional	The max number of entries to return, or just stream back all results if zero
query	string, optional	The query string to use

Parameter	Type, Options	Description
flags	string, optional	The flags to use for query. Can be a number (bitwise mask) or comma separated values like query-log.
threshold	uint64, optional	Query optimization to stop 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. Must 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	string, optional	A text search clause to apply to the where clause of this query
streamLimit	uint32, optional	Limit the number of sessions returned by a streaming query
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query

[Manual for query](#)

[reconfig](#)

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

Security.roles: sdk.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the config nodes with the defaults
memory	size, optional	If non-zero, will use that setting as installed 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 must be between 1 and 1677721, inclusive}	The max number of entries to return
search	string	The search string 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 starting session id (inclusive). Pass zero to scope to lowest valid id.
id2	uint64	The ending session id (inclusive). Pass zero to scope to highest valid id.

[Manual for session](#)

summary

Description: Retrieves summary information from the databases

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
flags	string, optional	Optional SDK flags. 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 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
time2	date-time, optional	The ending time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS)
timezone	int32, optional	The timezone of the local computer to receive the data, hour offset from GMT [-13 to 13]
size	uint32, {unsigned:The value must be between 1 and 1677721, inclusive}	The max number of entries to return
flags	string, optional	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	string, optional	Optional where clause for filtering 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.

Parameter	Type, Options	Description
fieldName	string, optional	The time field to retrieve values for, defaults to 'time'
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query
search	string, 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	string, {enum-one:The value must 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 just syntax will be validated

Parameter	Type, Options	Description
fieldName	string, optional	Required if validating a 'values' operation. The field to retrieve values for.
where	string, optional	Required if validating a 'values' operation. The filter criteria to be validated.
query	string, optional	Required if validating a 'query' operation. The query string 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 starting meta id
id2	uint64, optional	The ending meta id

Parameter	Type, Options	Description
size	uint32, {unsigned:The value must be between 1 and 1677721, inclusive}	The max number of entries to return
flags	string, optional	The flags 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 stop processing large session counts
fieldName	string	The field to retrieve values for
where	string, optional	The filter criteria for the values
aggregateFunction	string, optional, {enum-one:The value must be one of the following: count sum}	The aggregate operation to be applied where the aggregate flag is set
aggregateFieldName	string, optional	The meta field to aggregate in the aggregateFunction

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

[Manual for values](#)

xforms

Description: Retrieves transforms for the specified key

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
key	string, 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 trusted CA certs

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: list delete add}	CA action to perform (list,delete,add)
id	string, optional	ID of CA certificate to remove when op=delete

[Manual for caCert](#)

[count](#)

Description: Returns the number of child nodes

Security.roles: everyone

[fileEdit](#)

Description: View and edit system configuration files

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: dir get getBackup put create delete}	The operation to perform (dir, get, getBackup, put, create, delete)
type	string, optional, {enum-one:The value must be one of the following: parser filter}	The type of file, if not provided will check file extension
filename	string, optional, {length:The value length must 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are

Parameter	Type, Options	Description
		'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[peerCert](#)

Description: Display or modify trusted peer certificates

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must be one of the following: list delete add}	Operation to perform on the list of trusted peers (list,delete,add)

Parameter	Type, Options	Description
id	string, optional	ID of peer certificate to remove when op=delete

[Manual for peerCert](#)

save

Description: Forces a save of system settings

Security.roles: sys.manage

Parameters:

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

[Manual for save](#)

servCert

Description: Display the current server certificate 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	string, optional, {length:The value length must be between 0 and 512, inclusive}	The reason for shutting down the service
cl	string, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next restart

[Manual for shutdown](#)

statHist

Description: Retrieve historical stats from the stats db. Don't send time1/time2 to get bounding times about stats 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 starting time for retrieving stats
time2	date-time, optional	The ending time for retrieving stats
timeFormat	string, optional, {enum-one:The value must be one of the following: posix simple}	Specify the time format for each stat snapshot, default is posix (seconds since 1970)

Parameter	Type, Options	Description
include	string, optional	Comma separated list of stats to include (wildcards allowed)
exclude	string, optional	Comma separated list of stats to exclude (wildcards allowed, has precedence over include)
onChange	string, optional	Comma separated list of stats that reduce the result set to only when at least one of the stats change from their previous value
reduce	bool, optional	If true, reduces data transmission by replacing stat pathnames with a shorthand and provides a lookup table as first result
showAll	bool, optional	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.

[Manual for statHist](#)

[telemetry](#)

Description: Returns telemetry information about the service

Security.roles: sys.manage

Parameters:

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

[/users node](#)

[API Messages](#)

[addOrMod](#)

Description: Add a new user or update an existing user in the system

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#\$%&'+-=?^_`{ }~.[]\}	The username to add or update, must be alphanumeric or @!#\$%&' + - = ? ^ _ ` { } ~ . [] \
password	string, optional	The user's password
pwdIsHashed	bool, optional	Set to true if the password has already been hashed
groups	string, optional	The groups the user belongs to
authType	string, optional, {enum-one:The value must be one of the following: netwitness pam}	The authentication

Parameter	Type, Options	Description
queryTime out	uint32, optional, {unsigned:The value must be between 0 and 43200, inclusive}	system to use, "netwitness" is the default The maximum number of minutes that a query is allowed to execute for this user. Zero means unlimited.
displayName	string, optional	Display name for this user account
email	string, optional	Email address for this account

Parameter	Type, Options	Description
description	string, optional	Description of this user account
queryPrefix	string, optional	Query filter applied to every query performed by this user account
sessionThreshold	uint32, optional, {unsigned:The value must 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 system

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789@!#\$%&'+-=?^_`{ }~.[]\}	The username to delete

[help](#)

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

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ WXYZ?}	The name of the message to retrieve detailed

Parameter	Type, Options	Description
		help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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	string	The username or account to unlock

[whoAmI](#)

Description: Returns information about the authenticated user

Security.roles: everyone

Broker service

Introduction

Broker aggregates and store 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 Request

The following is a sample request to get top 3 values of ip.src through sdk values on the Broker.

```
Request path is /sdk , message is msg=values and parameters are fieldName=ip.src
where=\"time=\"'2021-08-10 16:30:00\"' - \"'2021-08-10 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 request timeout and continue with user profile timeout.

Using URL encoded request

```
$ curl 'https://broker-
host:50103/sdk?msg=values&fieldName=ip.src%20where=\"time=\"'2021-08-
10%2016:30:00\"'%20-%20'2021-08-10%2016:34:59\"'\\"&size=3&flags=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-string-params

```
$ curl 'https://broker-host:50103/sdk?msg=values&expiry=0' \
-H 'Accept: application/json;charset=UTF-8' \
-H 'Content-Type: application/x-netwitness-string-params' \
-d \"fieldName=ip.src where=\"time=\"'2021-08-10 16:30:00\"' - \"'2021-08-10 16:34:59\"'\\"
```

```
size=3 flags=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-store, must-revalidate  
Content-Type: application/json
```

JSON Response

```
{  
  "flags": 1074200577,  
  "results": {  
    "id1": 0,  
    "id2": 0,  
    "fields": [  
      {  
        "id1": 79390446,  
        "id2": 85993604,  
        "count": 104240,  
        "format": 128,  
        "type": "ip.src",  
        "flags": 1,  
        "group": 100000,  
        "value": "10.XX.XX.81"  
      },  
      {  
        "id1": 79115795,  
        "id2": 85854695,  
        "count": 101206,  
        "format": 128,  
        "type": "ip.src",  
        "flags": 1,  
        "group": 100000,  
        "value": "10.XX.XX.83"  
      },  
      {  
        "id1": 79114441,
```

```
"id2": 85709910,
"count": 100845,
"format": 128,
"type": "ip.src",
"flags": 1,
"group": 100000,
"value": "10.XX.XX.152"
}
]
},
"resultsInfo": {
"devices": "XX.XX.XX.85:50005=on XX.XX.XX.84:56005=on",
"offlineDeviceCount": "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-e3ff-4a23-a931-c9e0fbc2ebc8"
}
}
```

Sample Request

The following is a sample request to select 3 values of ip.src in ssl traffic originating from a country using sdk query on the Broker.

```
Request path is /sdk , message is msg=query and parameters are query="select
ip.src where time='\2021-08-10 18:50:00\' - \'2021-08-10
18:54:59\' && tcp.dstport=443 && country.src='United States'"
size=3
```

Using special content type

```
$ curl 'https://broker-host:50103/sdk?msg=values&expiry=0' \
-H 'Accept: application/json;charset=UTF-8' \
-H 'Content-Type: application/x-netwitness-string-params' \
-d "query=\"select ip.src where time=\"'\2021-08-10 18:50:00\' - '\2021-08-10 18:54:59\'
&& tcp.dstport=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-store, must-revalidate
Content-Type: application/json

JSON Response

```
{
  "flags": 1074200577,
  "results": {
    "id1": 2009111894,
    "id2": 2370947558,
    "fields": [
      {
        "id1": 2009111743,
        "id2": 2009111743,
        "count": 0,
        "format": 128,
        "type": "ip.src",
        "flags": 0,
        "group": 79178639,
        "value": "11.XX.XX.152"
      },
      {
        "id1": 2009111773,
        "id2": 2009111773,
        "count": 0,
        "format": 128,
        "type": "ip.src",
        "flags": 0,
        "group": 79178640,
        "value": "11.XX.XX.152"
      },
      {
        "id1": 2009111893,
        "id2": 2009111893,
        "count": 0,
        "format": 128,
        "type": "ip.src",
        "flags": 0,
        "group": 79178644,
        "value": "11.XX.XX.198"
      }
    ]
  },
  "resultsInfo": {
```

```

"devices": "10.XX.XX.85:50005=on 10.XX.XX.84:56005=on",
"offlineDeviceCount": "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-e3ff-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	string, optional, {ip-address:The value must 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 must be set
ip	string, optional	The IP address of the device, either this or device must be set
port	uint32, optional, {unsigned:The value must be between 0 and 65535, inclusive}	The port number the device is listening on

Parameter	Type, Options	Description
username	string	The username to run as
password	string, optional	The account password. Assumes service trust aggregation if no password is provided.
options	string, optional	The device options
ssl	string, optional	If true, the device requires SSL to connect
backup	string, optional	If true, the device will be used as a backup
reconnect	string, optional	If true, will automatically reconnect when not connected
name	string, optional	Device collection name if device is a collection

[count](#)

Description: Returns the number of child nodes

Security.roles: everyone

[delete](#)

Description: Delete an existing aggregation device

Security.roles: concentrator.manage

Parameters:

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

[edit](#)

Description: Edit an existing aggregation device

Security.roles: concentrator.manage

Parameters:

Parameter	Type, Options	Description
device	string, {ip-address:The value must be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to edit
username	string, optional	The username to run as

Parameter	Type, Options	Description
password	string, optional	The account password. For service trust aggregation you must not provide this parameter
options	string, optional	The device options
ssl	string, optional	If true, the device requires SSL to connect
backup	string, optional	If true, the device will be used as a backup
reconnect	string, optional	If true, will automatically reconnect when not connected
name	string, optional	Device collection name if device is a collection
switchToTrustedAuth	bool, optional	If true, removes any password associated with this account and will use trusted authentication henceforth. When switching to trusted, only the 'device' parameter is allowed.

[help](#)

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

Security.roles: everyone

Parameters:

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

Parameter	Type, Options	Description
		friendly format

[info](#)

Description: Returns detailed information about the node

Security.roles: everyone

[ls](#)

Description: Returns the list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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, manifests, stats, configuration, or logs for this service. Data automatically deletes index and stats, unless filesCreatedAfter is specified. Service is automatically restarted. Example arguments: data=1 config=1 log=1 This example will reset data, index, logs, and configuration index=1 This example will reset the index only manifest=1 This example will delete everything in the manifest directory filesCreatedAfter="2015-12-01 14:00:00" This example will delete all session, meta and packet files created on or after Dec 1st, 2015 2pm (UTC) from the service. All other files will remain. The index will not be touched, but upon restart will be truncated to match the last 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)
manifest	bool, optional	Reset all manifests stored in the long term manifest directory (may not be applicable to all services)
config	bool, optional	Reset configuration settings to default

Parameter	Type, Options	Description
stats	bool, optional	Reset the stats database
log	bool, optional	Reset the log database
filesCreatedAfter	date-time, optional	Delete all database files (session, meta, packets) created after a certain date. The index will automatically get rolled back on restart. Not valid with option index.
cl	string, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next restart

[resetMax](#)

Description: Resets all max stats, including device max stats back to zero.

Security.roles: concentrator.manage

[start](#)

Description: Starts aggregation

Security.roles: concentrator.manage

[status](#)

Description: Change the online/offline status of an aggregation device

Security.roles: concentrator.manage

Parameters:

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

[stop](#)

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	string, optional, {enum-any:The value must be one or more of the following: native rest amqp}	Indicates the type of connection to close, either native, rest or amqp. Default is all types.
limit	uint32, optional	If non-zero, will stop closing connections after limit is reached. NOTE: It starts closing the most idle connections first.
idleLongerThan	string, 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 `" "` must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ? WXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)

Parameter	Type, Options	Description
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.

Parameter	Type, Options	Description
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[netSpeed](#)

Description: Network Speed Test

Security.roles: connections.manage,aggregate

Parameters:

Parameter	Type, Options	Description
size	size, optional, {byte-range:The value (in bytes, size modifiers like MB are accepted) must be between 1024 and 536870912, inclusive}	The size of each message sent, default is 64k
count	uint32, optional, {unsigned:The value must 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 starting 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 `" "` must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

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

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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	string, optional	The device to report ranges for, defaults to 'all' devices
name	string, optional	The device collection name if the device is a collection
id1	uint64, optional	The starting session id

Parameter	Type, Options	Description
id2	uint64, optional	The ending session id
format	string, optional, {enum-one:The value must 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 starting log number to download

Parameter	Type, Options	Description
id2	uint64, optional	The ending log number to download
time1	date-time, optional	The starting 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	string, {enum-one:The value must be one of the following: start next cancel}	The operation to perform (start, next, cancel)
logTypes	string, optional	The type(s) of log messages to retrieve (debug,info,audit,warning,failure)
match	string, optional	Case insensitive string to match in each log message
regex	string, optional	Regular expression to match in each log message
timeFormat	string, optional, {enum-one:The value must be one of the following: posix simple}	The time format used in each log message, default is posix time (seconds since 1970)

Parameter	Type, Options	Description
batchSize	uint32, optional, {unsigned:The value must 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 `" "` must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual)

Parameter	Type, Options	Description
format	string, optional, {enum-one:The value must be one of the following: default xml html}	would return a man page on this node or the specified message) 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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 first log id number to retrieve, this is mutually exclusive with id2
id2	uint64, optional	The last log id number that will be sent, defaults to most recent log message when id1 or id2 is not sent

Parameter	Type, Options	Description
count	uint32, optional, {unsigned:The value must be between 1 and 10000, inclusive}	The number of logs to pull
timeFormat	string, optional, {enum-one:The value must 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	string, optional, {enum-any:The value must be one or more of the following: current last-write}	The time calculation to use (current date or last write date), default is current
minutes	uint32, optional, {unsigned:The value must be between 0 and 60, inclusive}	Remove log entries older than the given number of minutes
hours	uint32, optional, {unsigned:The value must be between 0 and 24, inclusive}	Remove log entries older than the given number of hours

Parameter	Type, Options	Description
days	uint32, optional, {unsigned:The value must be between 0 and 365, inclusive}	Remove log entries older than the given number of days
date	string, optional	Remove log entries older than the given date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

[/rest 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 `" "` must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters:	The name of the

Parameter	Type, Options	Description
	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[**/sdk node**](#)

[Manual for /sdk](#)

[**API Messages**](#)

[**aliases**](#)

Description: Retrieves aliases for the specified key

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
key	string	The language key to retrieve aliases for
value	string, optional	An optional value to return a specific 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	string, optional	The device UUID if taking a device offline instead of canceling the entire query (Broker only). Passing uuid will propagate the request upstream to find the first 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 first packet of the session, in case the session has rolled out
pinId	string, optional	The Pin ID of the session to retrieve from long term storage. If provided, the packet and session parameters must 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	string, optional	The render type (see NwContentTypes in NwSDK.h), pass zero to pick best option. Not specifying this parameter always results in an NWD and renderFlags and renderOptions are not considered. The following values are also supported: 'pcap', 'file-list-json', 'session-meta-file-list-json', 'nwd', 'log', 'files'
renderFlags	uint32, optional	Bitwise mask to control options, (see NwContentFlags in NwSDK.h)
renderOptions	string, optional	An encoded string 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 files from either the normal NWD cache or the pin cache

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
type	string, optional, {enum-one:The value must be one of the following: nwd pin all}	Specifies which cache to delete all NWD files 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 `" "` must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)

Parameter	Type, Options	Description
format	string, optional, {enum-one:The value must 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 definition

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The starting language id
id2	uint64, optional	The ending language id
time1	date-time, optional	The starting time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS)
time2	date-time, optional	The ending time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS)
options	string, optional	Extra options, currently unused
flags	string, optional	Optional flags to configure 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 must be between 1 and 1677721, inclusive}	The max number of entries to return
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query

Parameter	Type, Options	Description
devices	string, optional	Optional list 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 must match the list defined under /broker/devices or match the /sys/stats/uuid of each device to be queried.
excludeDevices	string, optional	Optional list of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs must match the UUID defined under /sys/stats/uuid of each device to be queried.

[Manual for language](#)

ls

Description: Returns the list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated

Parameter	Type, Options	Description
		values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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	string, optional	The session ID ranges to search
packets	string, optional	The packet ID ranges to search
search	string	The search string to use
where	string, optional	The where clause used to identify sessions to consider for the search

Parameter	Type, Options	Description
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
flags	string, {enum-any:The value must be one or more of the following: regex sp sm ci pre post ds precache si}	Flags to use for search. This is a comma separated list of one or more of the flag values: regex,sp,sm,ci,pre,post,ds,precache,si
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query
devices	string, optional	Optional list 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 must match the list defined under /broker/devices or match the /sys/stats/uuid of each device to be queried.
excludeDevices	string, optional	Optional list of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs must match the UUID defined under

Parameter	Type, Options	Description
		/sys/stats/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	string, {enum-one:The value must be one of the following: setup start cancel processed}	The operation to perform (start, cancel, processed)
sessions	string, optional	A comma separated list of session ids or session id ranges (#-#) whose packets will be streamed back.
packets	string, optional	A comma separated list of packet ids or session/packet ids (#&#) that will be streamed back.
where	string, optional	Where clause which will be evaluated to determine which sessions to stream back

Parameter	Type, Options	Description
time1	date-time, optional	A starting time range (UTC) where matching packets will be streamed back ("2010-Apr-20 09:00:00")
time2	date-time, optional	An ending time range (UTC) where matching packets will be streamed back ("2010-Apr-20 10:00:00")
flags	uint32, optional	Additional flags as defined by the NwPackets SDK function
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query
devices	string, optional	Optional list of comma separated device names that will be queried. If this parameter is not present, then all devices are queried. The device names must match the list defined under /broker/devices
excludeDevices	string, optional	Optional list of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs must match the UUID defined under /sys/stats/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 must 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 status 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	string, optional	A comma separated list of session ids or session id ranges (#-#) to pin. This cannot be used with the 'session' parameter.
pinId	string, optional	One or more Pin IDs, separated by commas
op	string, optional, {enum-one:The value must be one of the following: ls unpin validate}	Various operations that can be performed. Will be submitted to upstream devices if not configured for long term cache.

[Manual for pin](#)**[precache](#)**

Description: Efficiently caches NWD files for future retrieval via the content message

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
sessions	string, optional	A comma separated list of session ids or session id ranges (#-#) that will be saved in the NWD cache directory
packets	string, optional	A comma separated list 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 against the meta database

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The starting meta id (to run the query from most recent to oldest meta, make id1 larger than id2)
id2	uint64, optional	The ending meta id

Parameter	Type, Options	Description
size	uint32, optional	The max number of entries to return, or just stream back all results if zero
query	string, optional	The query string to use
flags	string, optional	The flags to use for query. Can be a number (bitwise mask) or comma separated values like query-log.
threshold	uint64, optional	Query optimization to stop 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. Must 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	string, optional	A text search clause to apply to the where clause of this query
streamLimit	uint32, optional	Limit the number of sessions returned by a streaming query
queryPriority	int32, optional, {signed:The value	The query priority for this query

Parameter	Type, Options	Description
devices	must be zero OR between -20 and 20, inclusive} string, optional	Optional list 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 must match the list defined under /broker/devices or match the /sys/stats/uuid of each device to be queried.
excludeDevices	string, optional	Optional list of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs must match the UUID defined under /sys/stats/uuid of each device to be queried.

[Manual for query](#)

[reconfig](#)

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

Security.roles: sdk.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the config nodes with the defaults

Parameter	Type, Options	Description
memory	size, optional	If non-zero, will use that setting as installed 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 must be between 1 and 1677721, inclusive}	The max number of entries to return
search	string	The search string 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 starting session id (inclusive). Pass zero to scope to lowest valid id.
id2	uint64	The ending session id (inclusive). Pass zero to scope to highest valid id.

[Manual for session](#)

summary

Description: Retrieves summary information from the databases

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
flags	string, optional	Optional SDK flags. 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 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
time2	date-time, optional	The ending time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS)
timezone	int32, optional	The timezone of the local computer to receive the data, hour offset from GMT [-13 to 13]
size	uint32, {unsigned:The value must be between 1 and 1677721, inclusive}	The max number of entries to return
flags	string, optional	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	string, optional	Optional where clause for filtering 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.
fieldName	string, optional	The time field to retrieve values for, defaults to 'time'

Parameter	Type, Options	Description
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query
search	string, optional	A text search clause to apply to the where clause of this query
devices	string, optional	Optional list 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 must match the list defined under /broker/devices or match the /sys/stats/uuid of each device to be queried.
excludeDevices	string, optional	Optional list of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs must match the UUID defined under /sys/stats/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	string, {enum-one:The value must 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 just syntax will be validated
fieldName	string, optional	Required if validating a 'values' operation. The field to retrieve values for.
where	string, optional	Required if validating a 'values' operation. The filter criteria to be validated.
query	string, optional	Required if validating a 'query' operation. The query string 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 starting meta id
id2	uint64, optional	The ending meta id
size	uint32, {unsigned:The value must be between 1 and 1677721, inclusive}	The max number of entries to return
flags	string, optional	The flags 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 stop processing large session counts
fieldName	string	The field to retrieve values for
where	string, optional	The filter criteria for the values
aggregateFunction	string, optional, {enum-one:The value must be one of the following: count sum}	The aggregate operation to be applied where the aggregate flag is set

Parameter	Type, Options	Description
aggregateFieldName	string, optional	The meta field to aggregate in the aggregateFunction
min	string, optional	The lower limit of values to return. Only values greater than this value will be returned.
max	string, optional	The upper limit of values to return. Only values less than this value will be returned.
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query
search	string, optional	A text search clause to apply to the where clause of this query
devices	string, optional	Optional list 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 must match the list defined under /broker/devices or match the /sys/stats/uuid of each device to be queried.
excludeDevices	string, optional	Optional list of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices

Parameter	Type, Options	Description
		are queried. The device UUIDs must match the UUID defined under /sys/stats/uuid of each device to be queried.

[Manual for values](#)

[xforms](#)

Description: Retrieves transforms for the specified key

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
key	string, 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 trusted CA certs

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: list delete add}	CA action to perform (list,delete,add)
id	string, optional	ID of CA certificate to remove when op=delete

[Manual for caCert](#)

count

Description: Returns the number of child nodes

Security.roles: everyone

fileEdit

Description: View and edit system configuration files

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: dir get getBackup put create delete}	The operation to perform (dir, get, getBackup, put, create, delete)
type	string, optional, {enum-one:The value must be one of the following: parser filter}	The type of file, if not provided will check file extension

Parameter	Type, Options	Description
filename	string, optional, {length:The value length must 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would

Parameter	Type, Options	Description
format	string, optional, {enum-one:The value must be one of the following: default xml html}	<p>return a man page on this node or the specified message)</p> <p>The format of the response, default returns in a human friendly format</p>

[info](#)

Description: Returns detailed information about the node

Security.roles: everyone

[ls](#)

Description: Returns the list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[peerCert](#)

Description: Display or modify trusted peer certificates

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must be one of the following: list delete add}	Operation to perform on the list of trusted peers (list,delete,add)
id	string, optional	ID of peer certificate to remove when op=delete

[Manual for peerCert](#)

[save](#)

Description: Forces a save of system settings

Security.roles: sys.manage

Parameters:

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

[Manual for save](#)

[servCert](#)

Description: Display the current server certificate 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	string, optional, {length:The value length must be between 0 and 512, inclusive}	The reason for shutting down the service

Parameter	Type, Options	Description
cl	string, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next restart

[Manual for shutdown](#)

statHist

Description: Retrieve historical stats from the stats db. Don't send time1/time2 to get bounding times about stats 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 starting time for retrieving stats
time2	date-time, optional	The ending time for retrieving stats
timeFormat	string, optional, {enum-one: The value must be one of the following: posix simple}	Specify the time format for each stat snapshot, default is posix (seconds since 1970)
include	string, optional	Comma separated list of stats to include (wildcards allowed)

Parameter	Type, Options	Description
exclude	string, optional	Comma separated list of stats to exclude (wildcards allowed, has precedence over include)
onChange	string, optional	Comma separated list of stats that reduce the result set to only when at least one of the stats change from their previous value
reduce	bool, optional	If true, reduces data transmission by replacing stat pathnames with a shorthand and provides a lookup table as first result
showAll	bool, optional	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.

[Manual for statHist](#)

[telemetry](#)

Description: Returns telemetry information about the service

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
level	uint32, optional, {signed:The value must	The level of information to be sent by the telemetry request. A value of 1 (default) will send all stat nodes, a value of 2 will send stat

Parameter	Type, Options	Description
	be between 1 and 3, inclusive}	and config nodes, and a value of 3 will send stat nodes, config nodes and usage metrics for rules, parsers and devices (where applicable).
include	string, optional	Comma separated list of nodes to include (wildcard usage is ok).
exclude	string, optional	Comma separated list of nodes to exclude (wildcard usage is ok). If "include" is also specified, exclusions are applied to the set of nodes specified by "include".
url	string, optional	Endpoint URL to send the JSON response.
op	string, optional, {enum-one:The value must be one of the following: get configure}	The telemetry operation to execute. Default is 'get'.

[/users node](#)

[API Messages](#)

[addOrMod](#)

Description: Add a new user or update an existing user in the system

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#\$%&'+-=?^_`{ }~.[]\}	The username to add or update, must be alphanumeric or @!#\$%&'+-=?^_`{ }~.[]\
password	string, optional	The user's password
pwdIsHashed	bool, optional	Set to true if the password has already been hashed
groups	string, optional	The groups the user belongs to
authType	string, optional, {enum-one:The value must be one of the following: netwitness pam}	The authentication

Parameter	Type, Options	Description
queryTimeout	uint32, optional, {unsigned:The value must be between 0 and 43200, inclusive}	system to use, "netwitness" is the default The maximum number of minutes that a query is allowed to execute for this user. Zero means unlimited.
displayName	string, optional	Display name for this user account
email	string, optional	Email address for this account

Parameter	Type, Options	Description
description	string, optional	Description of this user account
queryPrefix	string, optional	Query filter applied to every query performed by this user account
sessionThreshold	uint32, optional, {unsigned:The value must 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 system

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789@!#\$%&'+-=?^_`{ }~.[]\}	The username to delete

[help](#)

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

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ WXYZ?}	The name of the message to retrieve detailed

Parameter	Type, Options	Description
		help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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	string	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 investigations. Because it aggregates logs, it is almost always paired with one or more LogDecoders, not Decoders.

The aggregated data is stored in collections which can be configured 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.

Requests for the raw data (logs) are not forwarded to LogDecoder since the data is already stored 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 messages for a specific collection the request 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 against all available collections and queries issued on collection sdk path would be run against that collection.

Sample Request

The following is a sample request query on Archiver to get 4 values of distinct ip.dst, grouped by ip.src and ordered by distinct ip.dst in descending order.

Request path is /sdk, message is msg=query and parameters are query=\"select distinct(ip.dst) where time=\"'2021-08-10 16:50:00' - '2021-08-10 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 request timeout and continue with user profile timeout.

Using URL encoded request

```
$ curl 'https://archiver-host:50108/sdk?msg=query&expiry=0&query=select%20distinct(ip.dst)%20where%20time='2021-08-10%2016:50:00'%20-%20'2021-08-10%2016:54:59'%20GROUP%20BY%20ip.src%20ORDER%20BY%20distinct(ip.dst)%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-string-params

```
$ curl 'https://archiver-host:50108/sdk?msg=query&expiry=0' \
  -H 'Accept: application/json;charset=UTF-8' \
  -H 'Content-Type: application/x-netwitness-string-params' \
  -d "query=\"select distinct(ip.dst) where time=\"2021-08-10 16:50:00\" - \"2021-08-10 16:54:59\" GROUP BY ip.src ORDER BY distinct(ip.dst) 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-store, must-revalidate
Content-Type: application/json
```

JSON Response

```
{
  "flags": 1074200577,
  "results": {
    "id1": 1231176344,
    "id2": 1267146902,
    "fields": [
      {
        "id1": 1126481600,
        "id2": 1212151333,
        "count": 142,
        "format": 128,
```

```
"type": "ip.dst",
"flags": 6,
"group": 0,
"value": "192.XX.XX.246"
},
{
"count": 342,
"format": 128,
"type": "ip.dst",
"flags": 6,
"group": 0,
"value": "10.XX.XX.22"
},
{
"count": 320,
"format": 128,
"type": "ip.dst",
"flags": 6,
"group": 0,
"value": "10.XX.XX.174"
},
{
"count": 660,
"format": 128,
"type": "ip.dst",
"flags": 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	string, optional, {ip-address:The value must 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 must be set
ip	string, optional	The IP address of the device, either this or device must be set
port	uint32, optional, {unsigned:The value must be between 0 and 65535, inclusive}	The port number the device is listening on
username	string	The username to run as
password	string, optional	The account password. Assumes service trust aggregation if no password is provided.
options	string, optional	The device options

Parameter	Type, Options	Description
ssl	string, optional	If true, the device requires SSL to connect
backup	string, optional	If true, the device will be used as a backup
reconnect	string, optional	If true, will automatically reconnect when not connected
name	string, optional	Device collection name if device is a collection

count

Description: Returns the number of child nodes

Security.roles: everyone

delete

Description: Delete an existing aggregation device

Security.roles: archiver.manage

Parameters:

Parameter	Type, Options	Description
device	string, {ip-address:The value must be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to delete

Parameter	Type, Options	Description
name	string, optional	Device collection name if device is a collection

[edit](#)

Description: Edit an existing aggregation device

Security.roles: archiver.manage

Parameters:

Parameter	Type, Options	Description
device	string, {ip-address:The value must be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to edit
username	string, optional	The username to run as
password	string, optional	The account password. For service trust aggregation you must not provide this parameter
options	string, optional	The device options
ssl	string, optional	If true, the device requires SSL to connect

Parameter	Type, Options	Description
backup	string, optional	If true, the device will be used as a backup
reconnect	string, optional	If true, will automatically reconnect when not connected
name	string, optional	Device collection name if device is a collection
switchToTrustedAuth	bool, optional	If true, removes any password associated with this account and will use trusted authentication henceforth. When switching to trusted, only the 'device' parameter is allowed.

[help](#)

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

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters:	The name of the message to retrieve

Parameter	Type, Options	Description
	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[reconfig](#)

Description: Calculates optimal settings for archiver pools and buffers based on the installed 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 just output the calculations
memory	size, optional	If non-zero, will use that setting as installed 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 specified collection's mapping ranges and regenerate them anew.

Security.roles: archiver.manage

Parameters:

Parameter	Type, Options	Description
name	string, optional	The name of the collection if regenerating ranges

[reset](#)

Description: Reset data, index, manifests, stats, configuration, or logs for this service. Data automatically deletes index and stats, unless filesCreatedAfter is specified. Service is automatically restarted. Example arguments: data=1 config=1 log=1 This example will reset data, index, logs, and configuration index=1 This example will reset the index only manifest=1 This example will delete everything in the manifest directory filesCreatedAfter="2015-12-01 14:00:00" This example will delete all session, meta and packet files created on or after Dec 1st, 2015 2pm (UTC) from the service. All other files will remain. The index will not be touched, but upon restart will be truncated to match the last 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)
manifest	bool, optional	Reset all manifests stored in the long term manifest directory (may not be applicable to all services)
config	bool, optional	Reset configuration settings to default
stats	bool, optional	Reset the stats database
log	bool, optional	Reset the log database
filesCreatedAfter	date-time, optional	Delete all database files (session, meta, packets) created after a certain date. The index will automatically get rolled back on restart. Not valid with option index.

Parameter	Type, Options	Description
cl	string, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next restart

[resetMax](#)

Description: Resets all max stats, including device max stats back to zero.

Security.roles: archiver.manage

[start](#)

Description: Starts aggregation

Security.roles: archiver.manage

[status](#)

Description: Change the online/offline status of an aggregation device

Security.roles: archiver.manage

Parameters:

Parameter	Type, Options	Description
device	string, {ip-address:The value must be an IP address and port in the following format: [ip-address]:[0-65535]}	The device to change status
status	string, optional, {enum-one:The value must be one of the following: online offline}	Whether to take device offline or put online

Parameter	Type, Options	Description
name	string, optional	Device collection name if device is a collection

[stop](#)

Description: Stops aggregation

Security.roles: archiver.manage

Parameters:

Parameter	Type, Options	Description
flush	bool, optional	If true (default), process all sessions and packets currently in memory. Otherwise, stop 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	string, optional, {enum-any:The value must be one or more of the following: native rest amqp}	Indicates the type of connection to close, either native, rest or amqp. Default is all types.
limit	uint32, optional	If non-zero, will stop closing connections after limit is reached. NOTE: It starts closing the most idle connections first.
idleLongerThan	string, 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the

Parameter	Type, Options	Description
		specified message)
format	string, optional, {enum-one:The value must 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 list 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

Parameter	Type, Options	Description
options	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[netSpeed](#)

Description: Network Speed Test

Security.roles: connections.manage,aggregate

Parameters:

Parameter	Type, Options	Description
size	size, optional, {byte-range:The value (in bytes, size modifiers like MB are accepted) must be between 1024 and 536870912, inclusive}	The size of each message sent, default is 64k
count	uint32, optional, {unsigned:The value must 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 starting 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 `" "` must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed

Parameter	Type, Options	Description
		help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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	string, optional	The device to report ranges for, defaults to 'all' devices

Parameter	Type, Options	Description
name	string, optional	The device collection name if the device is a collection
id1	uint64, optional	The starting session id
id2	uint64, optional	The ending session id
format	string, optional, {enum-one:The value must 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 starting log number to download
id2	uint64, optional	The ending log number to download
time1	date-time, optional	The starting 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	string, {enum-one:The value must be one of the following: start next cancel}	The operation to perform (start, next, cancel)
logTypes	string, optional	The type(s) of log messages to retrieve (debug,info,audit,warning,failure)
match	string, optional	Case insensitive string to match in each log message
regex	string, optional	Regular expression to match in each log message
timeFormat	string, optional, {enum-one:The value must be one	The time format used in each log message, default is posix time (seconds since 1970)

Parameter	Type, Options	Description
	of the following: posix simple}	
batchSize	uint32, optional, {unsigned:The value must 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following:	The specific help

Parameter	Type, Options	Description
	messages parameters description values roles extra manual }	operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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 first log id number to retrieve, this is mutually exclusive with id2
id2	uint64, optional	The last log id number that will be sent, defaults to most recent log message when id1 or id2 is not sent

Parameter	Type, Options	Description
count	uint32, optional, {unsigned:The value must be between 1 and 10000, inclusive}	The number of logs to pull
timeFormat	string, optional, {enum-one:The value must 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	string, optional, {enum-any:The value must be one or more of the following: current last-write}	The time calculation to use (current date or last write date), default is current
minutes	uint32, optional, {unsigned:The value must be between 0 and 60, inclusive}	Remove log entries older than the given number of minutes
hours	uint32, optional, {unsigned:The value must be between 0 and 24, inclusive}	Remove log entries older than the given number of hours

Parameter	Type, Options	Description
days	uint32, optional, {unsigned:The value must be between 0 and 365, inclusive}	Remove log entries older than the given number of days
date	string, optional	Remove log entries older than the given date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

[/rest 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 `" "` must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters:	The name of the

Parameter	Type, Options	Description
	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[/sdk node](#)

[Manual for /sdk](#)

[API Messages](#)

[aliases](#)

Description: Retrieves aliases for the specified key

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
key	string	The language key to retrieve aliases for
value	string, optional	An optional value to return a specific 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	string, optional	The device UUID if taking a device offline instead of canceling the entire query (Broker only). Passing uuid will propagate the request upstream to find the first 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 first packet of the session, in case the session has rolled out
pinId	string, optional	The Pin ID of the session to retrieve from long term storage. If provided, the packet and session parameters must 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	string, optional	The render type (see NwContentTypes in NwSDK.h), pass zero to pick best option. Not specifying this parameter always results in an NWD and renderFlags and renderOptions are not considered. The following values are also supported: 'pcap', 'file-list-json', 'session-meta-file-list-json', 'nwd', 'log', 'files'
renderFlags	uint32, optional	Bitwise mask to control options, (see NwContentFlags in NwSDK.h)
renderOptions	string, optional	An encoded string 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 files from either the normal NWD cache or the pin cache

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
type	string, optional, {enum-one:The value must be one of the following: nwd pin all}	Specifies which cache to delete all NWD files 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 `" "` must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)

Parameter	Type, Options	Description
format	string, optional, {enum-one:The value must 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 definition

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The starting language id
id2	uint64, optional	The ending language id
time1	date-time, optional	The starting time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS)
time2	date-time, optional	The ending time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS)
options	string, optional	Extra options, currently unused
flags	string, optional	Optional flags to configure 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 must be between 1 and 1677721, inclusive}	The max number of entries to return
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query

Parameter	Type, Options	Description
devices	string, optional	Optional list 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 must match the list defined under /archiver/devices or match the /sys/stats/uuid of each device to be queried.
excludeDevices	string, optional	Optional list of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs must match the UUID defined under /sys/stats/uuid of each device to be queried.

[Manual for language](#)

[ls](#)

Description: Returns the list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.

Parameter	Type, Options	Description
exclude	string, optional	Comma separated list 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	string, optional	The session ID ranges to search
packets	string, optional	The packet ID ranges to search
search	string	The search string to use
where	string, 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

Parameter	Type, Options	Description
size	uint64, optional	The maximum number of results to return for this search
flags	string, {enum-any:The value must be one or more of the following: regex sp sm ci pre post ds precache si}	Flags to use for search. This is a comma separated list of one or more of the flag values: regex,sp,sm,ci,pre,post,ds,precache,si
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query
devices	string, optional	Optional list 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 must match the list defined under /archiver/devices or match the /sys/stats/uuid of each device to be queried.
excludeDevices	string, optional	Optional list of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs must match the UUID defined under /sys/stats/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	string, {enum-one:The value must be one of the following: setup start cancel processed}	The operation to perform (start, cancel, processed)
sessions	string, optional	A comma separated list of session ids or session id ranges (#-#) whose packets will be streamed back.
packets	string, optional	A comma separated list of packet ids or session/packet ids (#&#) that will be streamed back.
where	string, optional	Where clause which will be evaluated to determine which sessions to stream back
time1	date-time, optional	A starting time range (UTC) where matching packets will be streamed back ("2010-Apr-20 09:00:00")
time2	date-time, optional	An ending time range (UTC) where matching packets will be streamed back ("2010-Apr-20 10:00:00")

Parameter	Type, Options	Description
flags	uint32, optional	Additional flags as defined by the NwPackets SDK function
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query
devices	string, optional	Optional list of comma separated device names that will be queried. If this parameter is not present, then all devices are queried. The device names must match the list defined under /archiver/devices
excludeDevices	string, optional	Optional list of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs must match the UUID defined under /sys/stats/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 must 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 status 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	string, optional	A comma separated list of session ids or session id ranges (#-#) to pin. This cannot be used with the 'session' parameter.
pinId	string, optional	One or more Pin IDs, separated by commas
op	string, optional, {enum-one:The value must be one of the following: ls unpin validate}	Various operations that can be performed. Will be submitted to upstream devices if not configured for long term cache.

[Manual for pin](#)

[precache](#)

Description: Efficiently caches NWD files for future retrieval via the content message

Security.roles: sdk.content

Parameters:

Parameter	Type, Options	Description
sessions	string, optional	A comma separated list of session ids or session id ranges (#-#) that will be saved in the NWD cache directory

Parameter	Type, Options	Description
packets	string, optional	A comma separated list 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 against the meta database

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
id1	uint64, optional	The starting meta id (to run the query from most recent to oldest meta, make id1 larger than id2)
id2	uint64, optional	The ending meta id
size	uint32, optional	The max number of entries to return, or just stream back all results if zero
query	string, optional	The query string to use

Parameter	Type, Options	Description
flags	string, optional	The flags to use for query. Can be a number (bitwise mask) or comma separated values like query-log.
threshold	uint64, optional	Query optimization to stop 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. Must 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	string, optional	A text search clause to apply to the where clause of this query
streamLimit	uint32, optional	Limit the number of sessions returned by a streaming query
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query
devices	string, optional	Optional list of comma separated device names or UUIDs that will be queried. If this parameter is not present, then all devices are queried. The

Parameter	Type, Options	Description
excludeDevices	string, optional	device names must match the list defined under /archiver/devices or match the /sys/stats/uuid of each device to be queried. Optional list of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs must match the UUID defined under /sys/stats/uuid of each device to be queried.

[Manual for query](#)

[reconfig](#)

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

Security.roles: sdk.manage

Parameters:

Parameter	Type, Options	Description
update	bool, optional	If true (default is false), will automatically update the config nodes with the defaults
memory	size, optional	If non-zero, will use that setting as installed 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 must be between 1 and 1677721, inclusive}	The max number of entries to return
search	string	The search string 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 starting session id (inclusive). Pass zero to scope to lowest valid id.
id2	uint64	The ending session id (inclusive). Pass zero to scope to highest valid id.

[Manual for session](#)

summary

Description: Retrieves summary information from the databases

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
flags	string, optional	Optional SDK flags. 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 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
time2	date-time, optional	The ending time in seconds since 1970 or a time string (YYYY-MMM-DD HH:MM:SS)
timezone	int32, optional	The timezone of the local computer to receive the data, hour offset from GMT [-13 to 13]

Parameter	Type, Options	Description
size	uint32, {unsigned:The value must be between 1 and 1677721, inclusive}	The max number of entries to return
flags	string, optional	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	string, optional	Optional where clause for filtering 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.
fieldName	string, optional	The time field to retrieve values for, defaults to 'time'
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query
search	string, optional	A text search clause to apply to the where clause of this query
devices	string, optional	Optional list of comma separated device names or UUIDs that will be queried. If this

Parameter	Type, Options	Description
		parameter is not present, then all devices are queried. The device names must match the list defined under /archiver/devices or match the /sys/stats/uuid of each device to be queried.
excludeDevices	string, optional	Optional list of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs must match the UUID defined under /sys/stats/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	string, {enum-one:The value must 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 just syntax will be validated

Parameter	Type, Options	Description
fieldName	string, optional	Required if validating a 'values' operation. The field to retrieve values for.
where	string, optional	Required if validating a 'values' operation. The filter criteria to be validated.
query	string, optional	Required if validating a 'query' operation. The query string 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 starting meta id
id2	uint64, optional	The ending meta id

Parameter	Type, Options	Description
size	uint32, {unsigned:The value must be between 1 and 1677721, inclusive}	The max number of entries to return
flags	string, optional	The flags 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 stop processing large session counts
fieldName	string	The field to retrieve values for
where	string, optional	The filter criteria for the values
aggregateFunction	string, optional, {enum-one:The value must be one of the following: count sum}	The aggregate operation to be applied where the aggregate flag is set
aggregateFieldName	string, optional	The meta field to aggregate in the aggregateFunction

Parameter	Type, Options	Description
min	string, optional	The lower limit of values to return. Only values greater than this value will be returned.
max	string, optional	The upper limit of values to return. Only values less than this value will be returned.
queryPriority	int32, optional, {signed:The value must be zero OR between -20 and 20, inclusive}	The query priority for this query
search	string, optional	A text search clause to apply to the where clause of this query
devices	string, optional	Optional list 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 must match the list defined under /archiver/devices or match the /sys/stats/uuid of each device to be queried.
excludeDevices	string, optional	Optional list of UUIDs that will be excluded from being queried. If this parameter is not present, then all devices are queried. The device UUIDs must match the UUID defined under /sys/stats/uuid of each device to be queried.

[Manual for values](#)[xforms](#)

Description: Retrieves transforms for the specified key

Security.roles: sdk.meta

Parameters:

Parameter	Type, Options	Description
key	string, 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 trusted CA certs

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: list delete add}	CA action to perform (list,delete,add)

Parameter	Type, Options	Description
id	string, optional	ID of CA certificate to remove when op=delete

[Manual for caCert](#)

count

Description: Returns the number of child nodes

Security.roles: everyone

fileEdit

Description: View and edit system configuration files

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: dir get getBackup put create delete}	The operation to perform (dir, get, getBackup, put, create, delete)
type	string, optional, {enum-one:The value must be one of the following: parser filter}	The type of file, if not provided will check file extension
filename	string, optional, {length:The value length must 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 `" "` must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash `\`.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., <code>op=manual</code> would return a man page on this node or the specified message)

Parameter	Type, Options	Description
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.

Parameter	Type, Options	Description
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[peerCert](#)

Description: Display or modify trusted peer certificates

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must be one of the following: list delete add}	Operation to perform on the list of trusted peers (list,delete,add)
id	string, optional	ID of peer certificate to remove when op=delete

[Manual for peerCert](#)

[save](#)

Description: Forces a save of system settings

Security.roles: sys.manage

Parameters:

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

[Manual for save](#)

servCert

Description: Display the current server certificate 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	string, optional, {length:The value length must be between 0 and 512, inclusive}	The reason for shutting down the service
cl	string, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next restart

[Manual for shutdown](#)

statHist

Description: Retrieve historical stats from the stats db. Don't send time1/time2 to get bounding times about stats 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 starting time for retrieving stats
time2	date-time, optional	The ending time for retrieving stats
timeFormat	string, optional, {enum-one: The value must be one of the following: posix simple}	Specify the time format for each stat snapshot, default is posix (seconds since 1970)
include	string, optional	Comma separated list of stats to include (wildcards allowed)
exclude	string, optional	Comma separated list of stats to exclude (wildcards allowed, has precedence over include)
onChange	string, optional	Comma separated list of stats that reduce the result set to only when at least one of the stats change from their previous value

Parameter	Type, Options	Description
reduce	bool, optional	If true, reduces data transmission by replacing stat pathnames with a shorthand and provides a lookup table as first result
showAll	bool, optional	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.

[Manual for statHist](#)

[telemetry](#)

Description: Returns telemetry information about the service

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
level	uint32, optional, {signed:The value must be between 1 and 3, inclusive}	The level of information to be sent by the telemetry request. A value of 1 (default) will send all stat nodes, a value of 2 will send stat and config nodes, and a value of 3 will send stat nodes, config nodes and usage metrics for rules, parsers and devices (where applicable).
include	string, optional	Comma separated list of nodes to include (wildcard usage is ok).

Parameter	Type, Options	Description
exclude	string, optional	Comma separated list of nodes to exclude (wildcard usage is ok). If "include" is also specified, exclusions are applied to the set of nodes specified by "include".
url	string, optional	Endpoint URL to send the JSON response.
op	string, optional, {enum-one:The value must be one of the following: get configure}	The telemetry operation to execute. Default is 'get'.

[/users node](#)

API Messages

addOrMod

Description: Add a new user or update an existing user in the system

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#\$%&' +=?^`{ }~.[]\}	The username to add or update, must be

Parameter	Type, Options	Description
password	string, optional	<p>alphanumeric or @!#\$%&' +- =?^_`{ }~. []\</p> <p>The user's password</p>
pwdIsHashed	bool, optional	Set to true if the password has already been hashed
groups	string, optional	The groups the user belongs to
authType	string, optional, {enum-one:The value must be one of the following: netwitness pam}	The authentication system to use, "netwitness" is the default

Parameter	Type, Options	Description
queryTimeout	uint32, optional, {unsigned:The value must 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	string, optional	Display name for this user account
email	string, optional	Email address for this account
description	string, optional	Description of this user account

Parameter	Type, Options	Description
queryPrefix	string, optional	Query filter applied to every query performed by this user account
sessionThreshold	uint32, optional, {unsigned:The value must 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 system

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#\$%&'+-=?^_`{ }~.[]\}	The username to delete

[help](#)

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

Security.roles: everyone

Parameters:

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

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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	string	The username or account to unlock

[whoAmI](#)

Description: Returns information about the authenticated user

Security.roles: everyone

[Appliance service](#)

Introduction

Appliance is a service that co-exist along with other core services and provide options to collect appliance stats like disk, cpu, temperature etc..., also manage core services and configure storage 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 Request

The following is a sample request to get list of core services on the appliance.

Request path is `/appliance` and message is `msg=srvList`

```
$ curl 'https://appliance-host:50106/appliance?msg=srvList' \  
-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-store, must-revalidate  
Content-Type: application/json
```

JSON Response

It is possible for response values to contain text/plain key-value pairs in content for certain requests.

```
{  
  "flags": 1073938433,  
  "params": {  
    "localhost: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 filesystem monitor Example arguments:
path=/var/lib/netwitness/decoder/packetdb

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
path	string, {file:The file or directory must exist or be creatable on the system}	Filesystem 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".

[count](#)

Description: Returns the number of child nodes

Security.roles: everyone

[delFSMon](#)

Description: Delete a filesystem monitor Example arguments:
path=/var/lib/netwitness/decoder/packetdb

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
path	string, {file-system-monitor:Validates the file system exists}	Monitored filesystem path to remove

[devlist](#)

Description: list storage devices

Security.roles: appliance.manage

[dfAlloc](#)

Description: create a default configuration for storage used by a service

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
service	string, {enum-one:The value must be one of the following: archiver concentrator decoder logdecoder}	service that will use storage
volume	string, {enum-one:The value must be one of the following: centos}	volume group name
commit	bool, optional	commit changes

[Manual for dfAlloc](#)

[dpToNic](#)

Description: Remove an Ethernet device from DPDK

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
eth	string, optional, {enum-one:The value must be one of the following: }	Network interface to remove from DPDK

[ethtool](#)

Description: Show interface information

Security.roles: appliance.manage

[file](#)

Description: List, retrieve and delete files from approved directories on this device

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: ls get delete}	The operation to perform (ls, get, delete)
location	string, {enum-one:The value must be one of the following: crashreporter config update feeds parsers}	The location on the device where the operation will be performed

Parameter	Type, Options	Description
filename	string, optional	The name of the file to retrieve or delete

[fwdLogs](#)

Description: Enable or disable syslog forwarding Example arguments:
host=loghost.localdomain

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
host	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789=:/.-}	Remote syslog server

[help](#)

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

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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	string, optional, {enum-one:The value must be one of the following: enp0s3 enp0s8 lo}	Network interface to send to DPDK

[partClr](#)

Description: remove partitions on a block device

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
name	string, {enum-one:The value must be one of the following: 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	string, {enum-one:The value must be one of the following: }	block device name

Parameter	Type, Options	Description
service	string, {enum-one:The value must be one of the following: archiver concentrator decoder logdecoder}	service that will use storage
volume	string, optional, {enum-one:The value must 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](#)

[**powerOff**](#)

Description: Shut down this appliance No arguments required

Security.roles: appliance.manage

[**raidList**](#)

Description: list drive shelves attached to this appliance

Security.roles: appliance.manage

[**reboot**](#)

Description: Reboot this appliance No arguments required

Security.roles: appliance.manage

[**reset**](#)

Description: Reset data, index, manifests, stats, configuration, or logs for this service. Data automatically deletes index and stats, unless filesCreatedAfter is specified. Service is automatically restarted. Example arguments: data=1 config=1 log=1 This example will reset data, index, logs, and configuration index=1 This example will reset the index only manifest=1 This example will delete everything in the manifest directory filesCreatedAfter="2015-12-01 14:00:00" This example will delete all session, meta and packet files created on or after Dec 1st, 2015 2pm (UTC) from the service. All other files will remain. The index will not be touched, but upon restart will be truncated to match the last session in the session database.

Security.roles: appliance.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)
manifest	bool, optional	Reset all manifests stored in the long term manifest directory (may not be applicable to all services)
config	bool, optional	Reset configuration settings to default
stats	bool, optional	Reset the stats database
log	bool, optional	Reset the log database

Parameter	Type, Options	Description
filesCreatedAfter	date-time, optional	Delete all database files (session, meta, packets) created after a certain date. The index will automatically get rolled back on restart. Not valid with option index.
cl	string, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next restart

[serial](#)

Description: Retrieve the appliance serial number

Security.roles: everyone

[setNTP](#)

Description: Set the clock source for this appliance Example arguments:
source=tictoc.localdomain

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
source	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789=:/.-}	The hostname or address of an NTP server that will serve as

Parameter	Type, Options	Description
		the hostname 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: enable=1

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
enable	uint32, {unsigned:The value must be between 0 and 1, inclusive}	Enable the SNMP service. Set to 1 to enable the service, and 0 to disable the service

[srvAlloc](#)

Description: apply volume group storage to a service on this appliance

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
service	string, {enum-one:The value must be one of the following: archiver concentrator decoder logdecoder}	service that will use storage
volume	string, {enum-one:The value must be one of the following: centos}	volume group name
commit	bool, optional	commit changes

[Manual for srvAlloc](#)

[srvFree](#)

Description: remove volume group storage from a service on this appliance

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
service	string, {enum-one:The value must be one of the following: archiver concentrator decoder logdecoder}	service that will use storage
volume	string, {enum-one:The value must be one of the following: centos}	volume group name
commit	bool, optional	commit changes

[Manual for srvFree](#)

[srvList](#)

Description: list of core services on this appliance

Security.roles: appliance.manage

[start](#)

Description: Start a Netwitness service on this appliance Example arguments:
service=decoder

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
service	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzaBCDEFGHIJKLMNOPQRSTUVWXYZ0123456789=:/.-}	The name of the service to start (decoder, concentrator, etc.)

[stop](#)

Description: Stop a Netwitness service on this appliance Example arguments:
service=decoder

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
service	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789=:/.-}	The name of the service to stop (decoder, concentrator, etc.)

unreg

Description: Collect missing parameters and save them to `/var/netwitness/missing.json`

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
op	string, optional	If 'list', then returns all params currently unregistered
m	string, optional	The pathname and message name
p	string, optional	The parameter name
module	string, optional	The module
version	string, optional	The version

Parameter	Type, Options	Description
buildDate	string, optional	The build date

upgrade

Description: Upgrade the server process

Security.roles: appliance.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: start check cancel finished}	The operation to perform (start, cancel, finished)
size	uint64, optional	The size of the incoming upgrade package
filename	string, optional	The name of the incoming upgrade package
saveFile	bool, optional	If one, the upgrade file will be saved, otherwise it is deleted
crcFile	bool, optional	If true, will perform a CRC of the file and compare to the client's CRC

Parameter	Type, Options	Description
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 restarted after upgrade

[vgs](#)

Description: list volume groups

Security.roles: appliance.manage

[Manual for vgs](#)

[vmstat](#)

Description: Get CPU and VM statistics

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
count	uint32, optional, {unsigned:The value must be between 1 and 30, inclusive}	The number of times to gather statistics 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	string, optional, {enum-any:The value must be one or more of the following: native rest amqp}	Indicates the type of connection to close, either native, rest or amqp. Default is all types.
limit	uint32, optional	If non-zero, will stop closing connections after limit is reached. NOTE: It starts closing the most idle connections first.
idleLongerThan	string, 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the

Parameter	Type, Options	Description
		specified message)
format	string, optional, {enum-one:The value must 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 list 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

Parameter	Type, Options	Description
options	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[netSpeed](#)

Description: Network Speed Test

Security.roles: connections.manage

Parameters:

Parameter	Type, Options	Description
size	size, optional, {byte-range:The value (in bytes, size modifiers like MB are accepted) must be between 1024 and 536870912, inclusive}	The size of each message sent, default is 64k
count	uint32, optional, {unsigned:The value must 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 starting log number to download
id2	uint64, optional	The ending log number to download
time1	date-time, optional	The starting 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	string, {enum-one:The value must be one of the following: start next cancel}	The operation to perform (start, next, cancel)
logTypes	string, optional	The type(s) of log messages to retrieve (debug,info,audit,warning,failure)

Parameter	Type, Options	Description
match	string, optional	Case insensitive string to match in each log message
regex	string, optional	Regular expression to match in each log message
timeFormat	string, optional, {enum-one: The value must 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 must 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars: The value can only contain the characters:	The name of the message to retrieve

Parameter	Type, Options	Description
	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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 first log id number to retrieve, this is mutually exclusive with id2

Parameter	Type, Options	Description
id2	uint64, optional	The last log id number that will be sent, defaults to most recent log message when id1 or id2 is not sent
count	uint32, optional, {unsigned:The value must be between 1 and 10000, inclusive}	The number of logs to pull
timeFormat	string, optional, {enum-one:The value must 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	string, optional, {enum-any:The value must be one or more of the following: current last-write}	The time calculation to use (current date or last write date), default is current
minutes	uint32, optional, {unsigned:The value must be between 0 and 60, inclusive}	Remove log entries older than the given number of minutes

Parameter	Type, Options	Description
hours	uint32, optional, {unsigned:The value must be between 0 and 24, inclusive}	Remove log entries older than the given number of hours
days	uint32, optional, {unsigned:The value must be between 0 and 365, inclusive}	Remove log entries older than the given number of days
date	string, optional	Remove log entries older than the given date (YYYY-MM-DD HH:MM:SS), not compatible with minutes, hours, or days parameters

[/rest 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[/sys node](#)

[Manual for /sys](#)

[API Messages](#)

[caCert](#)

Description: Display or delete trusted CA certs

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: list delete add}	CA action to perform (list,delete,add)
id	string, optional	ID of CA certificate to remove when op=delete

[Manual for caCert](#)

count

Description: Returns the number of child nodes

Security.roles: everyone

fileEdit

Description: View and edit system configuration files

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, {enum-one:The value must be one of the following: dir get getBackup put create delete}	The operation to perform (dir, get, getBackup, put, create, delete)
type	string, optional, {enum-one:The value must be one of the following: parser filter}	The type of file, if not provided will check file extension

Parameter	Type, Options	Description
filename	string, optional, {length:The value length must 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 must be in double quotes if there is whitespace. To pass a quote in the value, you must escape it by preceding it with a backslash \.

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would

Parameter	Type, Options	Description
format	string, optional, {enum-one:The value must be one of the following: default xml html}	<p>return a man page on this node or the specified message)</p> <p>The format of the response, default returns in a human friendly format</p>

[info](#)

Description: Returns detailed information about the node

Security.roles: everyone

[ls](#)

Description: Returns the list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list of nodes or node pathnames to exclude (wildcards allowed: * and ?)

[peerCert](#)

Description: Display or modify trusted peer certificates

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
op	string, optional, {enum-one:The value must be one of the following: list delete add}	Operation to perform on the list of trusted peers (list,delete,add)
id	string, optional	ID of peer certificate to remove when op=delete

[Manual for peerCert](#)

[save](#)

Description: Forces a save of system settings

Security.roles: sys.manage

Parameters:

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

[Manual for save](#)

[servCert](#)

Description: Display the current server certificate 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	string, optional, {length:The value length must be between 0 and 512, inclusive}	The reason for shutting down the service

Parameter	Type, Options	Description
cl	string, optional	A (cl="name=value name2=value2") set of options for setting the command line on the next restart

[Manual for shutdown](#)

statHist

Description: Retrieve historical stats from the stats db. Don't send time1/time2 to get bounding times about stats 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 starting time for retrieving stats
time2	date-time, optional	The ending time for retrieving stats
timeFormat	string, optional, {enum-one: The value must be one of the following: posix simple}	Specify the time format for each stat snapshot, default is posix (seconds since 1970)
include	string, optional	Comma separated list of stats to include (wildcards allowed)

Parameter	Type, Options	Description
exclude	string, optional	Comma separated list of stats to exclude (wildcards allowed, has precedence over include)
onChange	string, optional	Comma separated list of stats that reduce the result set to only when at least one of the stats change from their previous value
reduce	bool, optional	If true, reduces data transmission by replacing stat pathnames with a shorthand and provides a lookup table as first result
showAll	bool, optional	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.

[Manual for statHist](#)

[telemetry](#)

Description: Returns telemetry information about the service

Security.roles: sys.manage

Parameters:

Parameter	Type, Options	Description
level	uint32, optional, {signed:The value must	The level of information to be sent by the telemetry request. A value of 1 (default) will send all stat nodes, a value of 2 will send stat

Parameter	Type, Options	Description
	be between 1 and 3, inclusive}	and config nodes, and a value of 3 will send stat nodes, config nodes and usage metrics for rules, parsers and devices (where applicable).
include	string, optional	Comma separated list of nodes to include (wildcard usage is ok).
exclude	string, optional	Comma separated list of nodes to exclude (wildcard usage is ok). If "include" is also specified, exclusions are applied to the set of nodes specified by "include".
url	string, optional	Endpoint URL to send the JSON response.
op	string, optional, {enum-one:The value must be one of the following: get configure}	The telemetry operation to execute. Default is 'get'.

[/users node](#)

[API Messages](#)

[addOrMod](#)

Description: Add a new user or update an existing user in the system

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#\$%&'+-=?^_`{ }~.[]\}	The username to add or update, must be alphanumeric or @!#\$%&'+-=?^_`{ }~.[]\
password	string, optional	The user's password
pwdIsHashed	bool, optional	Set to true if the password has already been hashed
groups	string, optional	The groups the user belongs to
authType	string, optional, {enum-one:The value must be one of the following: netwitness pam}	The authentication

Parameter	Type, Options	Description
queryTime out	uint32, optional, {unsigned:The value must be between 0 and 43200, inclusive}	system to use, "netwitness" is the default The maximum number of minutes that a query is allowed to execute for this user. Zero means unlimited.
displayName	string, optional	Display name for this user account
email	string, optional	Email address for this account

Parameter	Type, Options	Description
description	string, optional	Description of this user account
queryPrefix	string, optional	Query filter applied to every query performed by this user account
sessionThreshold	uint32, optional, {unsigned:The value must 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 system

Security.roles: users.manage

Parameters:

Parameter	Type, Options	Description
name	string, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@!#\$%&'+-=?^_`{ }~.[]\}	The username to delete

[help](#)

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

Security.roles: everyone

Parameters:

Parameter	Type, Options	Description
msg	string, optional, {chars:The value can only contain the characters: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?}	The name of the message to retrieve detailed

Parameter	Type, Options	Description
		help about (aliases are 'm' or 'message')
op	string, optional, {enum-one:The value must be one of the following: messages parameters description values roles extra manual }	The specific help operation to perform (e.g., op=manual would return a man page on this node or the specified message)
format	string, optional, {enum-one:The value must 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 list 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	string, optional	What types of nodes to return information about, default is all nodes. Can be a number (bitwise mask) or comma separated values like config, stat, folder, session, connection, channel, restart-needed or pretty-print.
exclude	string, optional	Comma separated list 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	string	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 persist the state of the databases so they can be reloaded in seconds upon restart. There is an automatically scheduled command to write out the state every so often and it will always be written out on a normal shutdown. There are also options to view key information about a specific 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 files. This is only useful if hashing has been enabled (see configuration 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 manifest

Manifest files are created with every session, meta, and packet (log) DB file and index slice directory. A manifest file is a file that describes several key pieces of information about the data to which it refers. Manifest files are written as a JSON record. Manifest files travel with the data they represent from tier to tier. If the data they represent is deleted, the manifest file is also deleted, except in the following special case. If the service has `/database/config/manifest.dir` configured to a valid directory, at the point when the manifest data is deleted, a copy of the manifest file is placed into the directory pointed at by `manifest.dir` (the directory is created if it does not exist). This enables a NetWitness Platform feature called historical manifest searching.

The intention of this process is to keep historical manifest files for years, in one location for offline querying. As you might imagine from a service running for many years, this can potentially generate hundreds of thousands of files. This should not be a concern however, as the service automatically compresses files into a single archive in order to save space when they grow too numerous. Manifest files 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: `manifest time1="2014-04-20 11:00:00" time2="2014-04-11 11:20:00" timeFormat=simple`

More information on manifests can be found in the Core Database Tuning Guide.

/database reconfig

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 filesystem and will only be used for storage 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 filling up, it's not ideal when one database has too much retention and the oldest data could be orphaned because the other database is aging out faster. 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 specified databases aging out at roughly the same time period. In order for this to work, the databases being managed by *sizeRoll* must all share the same storage space(s). Since they share the same storage, there is a tradeoff in I/O performance. Typically, Archiver uses the *sizeRoll* command to manage its space, as maximum retention is more important than I/O performance.

There are two *sizeRoll* commands, one on the *database* node and one on the *index* node. The only difference 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 files based on your criteria. For more information on running scheduled commands, see the Core Database Tuning Guide.

/database stagger

The stagger command is typically only useful for a 10G Decoder and usually just for the packet database. Maximum performance is achieved for storing and retrieving packets when multiple volumes are present. In this scenario, the Decoder always fills the volume with the most free space. When the volumes are roughly the same size, this results in a staggered write pattern, which allows maximum throughput for reading and writing across all volumes. However, this only naturally occurs when multiple packet storage volumes are present at the time the Decoder is first deployed.

A typical use case is adding more storage to an existing Decoder to increase retention. However, when adding storage to a deployment that has already filled the existing volumes with stored packets, the Decoder will naturally fill the new storage with packets before rolling out any packets on the existing storage. This results in a suboptimal read/write

pattern because most 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 stop ALL reads on that volume, because the file is buffered in memory before being written, but it does result in suboptimal read performance.

With the **stagger** command, you can add more storage and then have the service naturally stagger the files across ALL volumes (existing and new) so that read performance is optimized. This command should only be performed AFTER the storage is mounted and the Decoder configured 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 stagger and the Decoder should **not** be capturing during the stagger operation.

Recommended workflow:

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 instance, if you have a requirement to only store 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 files that are older than 3.5 days from the current time. If you stop capture, then it will continue to age out the files until only the most recent file is left. It is not permitted to remove the most recent file 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 stopped, then change the timeCalc parameter to be "last-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 files 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 configuration of those processes.

The following is a brief overview of the Decoder commands and what they do:

- **sslKeys** - Provides an interface to upload premaster 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 start from a fresh configuration. 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 step, so that the enabled parsers will see the unencrypted packet payload and create meta accordingly. Otherwise, most 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 stage. You should plan accordingly by making sure the incoming traffic bandwidth does not overwhelm the available compute power. In other words, you may need more Decoders to decrypt traffic than you would if not decrypting.

Packets captured on a Decoder normally have a timeout of around 60 seconds (in the assembly stage) before they are sent to the parse step. 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 adjusted through configuration 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 must be received by the Decoder before the parsing stage.

NOTE: Currently, 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.

There are two types of encryption keys that 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.

Premaster

The *premaster* key is generated randomly and is ephemeral for the life of one specific TLS session. Normally, there is not an easy way to get *premaster* keys to a Decoder in time for the parsing step. However, both Chrome and Firefox can write the premaster keys they generate to a file. This is useful for testing purposes. To configure 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 file to write the keys to. Decoder will accept the file exactly as it is written and will use all the decryption keys in the file for any encrypted traffic it captures. The following is a sample `NwConsole` script that uploads the file 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 stored in memory and there is a limit to how many can be stored at any point in time. As more are added, the earliest keys will be aged out. You can also add premaster keys by just passing the *random* and *premaster* parameters to `sslKeys`.

Private Keys or PEM files

Private keys are normally stored in PEM files and are the asymmetric keys generated by services that accept TLS traffic. These keys are used during the TLS handshake to encrypt the premaster symmetric key that will be used for the rest of the payload encryption. For example, if you have a web server whose traffic you want visibility into, then you would upload the private key it uses to encrypt traffic. You only need to do this once, as it is stored permanently (or until a delete command). Once installed, 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 installed key becomes visible to the HTTPS parser. The following are some sample commands that will upload a PEM file 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 must 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.p  
em"
```

Private keys are automatically encrypted before storing to protect them.

Key Management

The following is the full list 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.
- **premaster** - The premaster key that will be installed for the previous random parameter. They must show up in pairs and random must be first.

Return Values

Most commands return name/value pairs of stats about the premaster keys in memory. The following explains what they mean:

- **added** - The number of premaster keys just added during this command

- **total** - The total number of premaster keys loaded in memory
- **agedOut** - The total number of premaster keys that were removed during this command (this is not a lifetime stat)
- **maxKeys** - The current maximum allowed premaster keys

Viewing Unencrypted Traffic

Even though the packets are decrypted during the parse stage, only the encrypted packets are written to disk. However, the matching premaster key used for decrypting is written to the *tls.premaster* meta key. This key can then be used to subsequently decrypt the packets on the fly. 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 *flags* 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 *flags* parameter to see what it looks like encrypted (or at least remove the 0x80 mask). For more information on the `/sdk/content` service, see the manual page for `/sdk/content`.

`/decoder start`

Starts capture on the selected interface.

`/decoder stop`

Stops capture and flushes 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. Most 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 configured 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 stored on disk and return it as one of many different representations. Or it may just 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 Investigator thick client. Nowadays, most third party API access revolves around extracting one or more files out of a specific 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

C Enum	Value	String Equivalent	Description
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.

C Enum	Value	Comment
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.

C Enum	Value	Comment
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 just `/sdk/content`, but would be different 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 files from well known protocols. For instance, it is possible to extract all files transferred over HTTP, SMTP, POP3 or SMB, assuming the full session was captured. When requesting files over the RESTful interface, the files will be returned as [multipart/mixed](#).

Example usage for extracting files 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 consists 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 `hierarch` command returns it's information as a `MessagePack` object, which may be translated into different 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 upstream devices, such as a Broker or Concentrator, the `hierarch` message will contain a `devices` member. The `devices` member is an array that holds the contents of the `hierarch` message as executed on each upstream 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 list 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 list has two types of entries:

Renamed keys can be distinguished from entities in the result set because renamed keys also include the key itself in the list of referenced keys.

/sdk language

Returns the current meta database schema as well as the level of indexing for each key as defined by the current configuration 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

C Enum Flag	Value	Comment
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.

C Enum Flag	Value	Comment
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 against all meta types. This type of search does not require users to define their queries in terms of known meta types. Instead, 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 *Investigation 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
<code>sp</code>	Scans raw packet data
<code>sm</code>	Scans all meta data
<code>si</code>	Does index lookups for all search parameters before scanning meta
<code>ci</code>	Performs a case insensitive search. Returned results are case-preserving.
<code>regex</code>	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, specified by using the `si` flag, causes results to be returned significantly faster 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 filtered out. If you want to search for sessions containing any of two or more terms, you must 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 value-indexed keys in the index. Search terms provided to `msearch` will find 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 Customization](#).

The wildcard search allows the use of the `*` and `?` characters as wildcards in search terms. The `*` can stand for 0 or more characters, while the `?` may stand 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 finding text that begins with a known prefix.

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"

Search input	Non-indexed	IndexValues	IndexValues with Edge N-grams	IndexValues with All N-grams
"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 starting 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 files 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 just `/sdk/packets`, but would be different 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 list 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 structure, and is passed verbatim to the packets handler.

Examples of the parameters string:

```
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 store sessions (with meta and packets) for long term retention, outside of the normal rollover mechanics. For instance, if several sessions indicate there might be an attack on a customer'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 storage. When pinning multiple sessions, you will get back multiple responses, each response will be for a single session.

The Pin ID should be stored for later retrieval. For instance, Respond will store the Pin IDs for each incident in which the customer requested 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 must 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-meta-param}, {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 mostly benefits developers who are writing applications directly against the NetWitness Core database. Normally, results are returned in the order of oldest to newest data (higher meta IDs are always more recent). In order to return results from most recent to oldest, reverse the IDs such that `id1` is larger than `id2`. This has a slight performance penalty, because the where clause must be completely evaluated before processing in reverse order can begin.

When `size` is left off or set to zero, the system streams 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 specifies how many groups of results will be returned when the query is operating in streaming mode. For example, the query may be in streaming mode by virtue of not having a 'size' parameter, but the `streamLimit` can be used to limit how many results are actually returned. Specifying the `streamLimit` implies that streaming mode is requested. If the client requests a 'streamLimit' but the query cannot be processed in streaming mode due, then an error is generated.

The `query` parameter is a query command string with its own NetWitness-specific 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 field 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 stored 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 consists of `ip.dst` and `ip.src`, then a query containing `select ip` will only return `ip` fields, with nothing to distinguish `ip.dst` meta items from `ip.src` meta items in the result set.

The `select` clause may contain renamed meta key names. Any fields 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` fields, even if the underlying meta had type `tcp.srcport`.

Note: Usage of renamed meta key pairs in the `select` clause cannot be combined with fixed-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 still preserve the correctness of the result set for the requested 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 effect on the query result set.

Function	Result
<code>sum</code>	Add all meta values together; only works on numbers
<code>count</code>	The total number of meta fields that would have been returned
<code>min</code>	The minimum value seen
<code>max</code>	The maximum value seen

Function	Result
<code>avg</code>	The average value for the number
<code>distinct</code>	Returns a list of all unique values seen
<code>countdistinct</code>	Returns the number of unique values seen. <code>countdistinct</code> is equivalent to the number of metas that would have been returned by the <code>distinct</code> function.
<code>first</code>	Returns the first value seen
<code>last</code>	Returns the last value seen
<code>len</code>	Converts all field values to a <code>UInt32</code> 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 <code>ip.src</code> , return 4 bytes.
<code>avglen</code>	Returns a single value which is the average value returned from the <code>len</code> function. The result is always a <code>float64</code> value.

where Clauses

The `where` clause is a filter specification 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 specified 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" specified for the key. See "Key Renaming" under the [Index Customization](#) 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

Operator	Function
	range is considered. If multiple ranges are specified, the behavior is undefined. For text metas, a lexicographical comparison is performed.
<code><=</code>	Same behavior as <code><</code> , but sessions containing meta that equals the value exactly are also considered matches.
<code>></code>	Similar to the <code><</code> 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.
<code>>=</code>	Same behavior as <code>></code> , but sessions containing meta that equals the value exactly are also considered matches.
<code>begins</code>	Matches sessions that contain text meta value that starts with the same characters as the right side.
<code>ends</code>	Matches sessions that contain text meta that ends with the same characters as the right side.
<code>contains</code>	Matches sessions that contain text meta that contains the substring given on the right side.
<code>regex</code>	Matches sessions that contain text meta that matches the regex given on the right side. The regex parsing is handled by <code>boost::regex</code> .
<code>exists</code>	Matches sessions that contain any meta value with the given meta key.
<code>!exists</code>	Matches sessions that do not contain any meta value with the given meta key.

Operator	Function
length	Matches sessions that contain text meta values of a certain length. The expression on the right side must be a non-negative number.

Text Values

The system 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 must be escaped, as `\\`. Note that if you are wrapping the query parameters within another language, such as the parameter fields 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 standard 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 specified using standard 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 most situations such searches will return sessions that have a meta value that exactly matches the requested 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 specified is exactly equal to one of the bucket values, then the system 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 system 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 Customization](#).

Numeric Value Aliases

For numeric values, aliases specified in the index can be used in a query as a quoted string 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 list of values (and/or ranges).

Refer to the topic [Index Customization](#) for details of how value aliases can be specified 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 string representation. The string representation for the date and time is "`YYYY-mm-DD HH:MM:SS`". A three-letter abbreviation represents the month. You can also express the Month as a two-digit number, 01-12.

Time values must be quoted.

All times specified in queries are expected to be in UTC.

Relative Time Points

Relative time points allow a where clause to reference a value at some fixed offset relative to the earliest or latest 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 earliest time present in the collection. If the boundary is `latest`, the duration represents the amount of time **before** the earliest 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 system clock of the Core service host 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 offset. This is most useful in the case of the `now` boundary, since it is possible that the highest, latest, 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 earliest and latest 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 start of the range, and `u` to represent the upper bound. The bounds are determined by looking at the smallest or largest 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 most recent time found in the collection.

Notice that it is easy to confuse a range expression with a text string. 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 specified, the result set for the query is subdivided into groups. Each group of results is uniquely identified 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.dst` 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.dst` metas found for each unique `ip.src`.

As of version 10.5, you can utilize up to 6 meta fields in a `group by` clause.

The `group by` clause shares some of the same functionality as the `values` call, but it offers significantly 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 defined by the `select` clause. The `select` clause can contain any of the aggregate functions or meta fields selected. If multiple aggregates are selected, the result of the aggregate function is defined for each group. If nonaggregate fields are selected, the meta fields 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 field 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 differ from a SQL-like database in terms of what meta fields 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 first.

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 fixed-size result set is requested. 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 definition.

The following table describes the database honors configuration parameters that limit I/O or memory impact of a group by query.

Parameter	Function
<code>/sdk/config/max.query.groups</code>	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.

Parameter	Function
<code>/sdk/config/max.where.clause.sessions</code>	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` consists of a set of items to sort by in ascending or descending order. Sorting can be performed on any data field that will be returned in the result set. This includes meta specified by the `select` clause, aggregate function results specified by the `select` clause, or `group by` meta fields.

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 exists in that each of the `order by` columns must 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 first 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 first in the result set in the case of an ascending sort, and last 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 distinct meta values.

If an `order by` clause references one of the fields in the group that has multiple values, the behavior of the returned result set depends on whether the query is processed as a **streaming query** or a **non-streaming 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 clause refers to an indexed meta key. The index for the meta key is used to determine which sessions to stream back to the client first. Streaming queries allow for fast server-side sorting of results. They can be used to allow `order by` clauses on much larger data sets than non-streaming queries, and they return results much faster than non-streaming queries. When a streaming 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 effect this has on sessions that might get fetched for more than one meta distinct value in step 2. When that happens the session might appear more than once in the result set stream. This is logically valid because the session might have more than one 'correct' position in the sorted result set.

Non-streaming queries are those queries that cannot be streamed back to the client and require the Core database to hold the entire result set in memory before determining the final sort order. The most 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 streamed 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 fields into distinct 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 distinct 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 different 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 field 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 specified in the `query` parameter are combined with the search filter as if they were joined with an `AND` operation.

Searches through the `query` API are always done against indexed meta, in case-insensitive mode. It has the same behavior as specifying flags `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 statistics 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.

Definitions:

- **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 fields that are of type *TimeT*.

Internally, the *timeline* call performs a *values* call on the passed in fieldName (usually "time"). The *values* call returns the counts for the where clause in minute intervals (you can literally get the same data just using *values*). However, what the *timeline* does do is perform what's known as a "roll up". For request intervals less than or equal to 24 hours, it just 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 specified by the caller, but is controlled by the amount of data requested. 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 must 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 stores them accurately.

Each result returned is made up of the following fields:

- **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 `fieldName` 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 stored 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-func-param}, {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" | "order-ascending" | "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 `flags` parameter.

Parameters

The following table describes the function of each parameter.

Parameter	Function
<code>fieldName</code>	This is the meta key name for which you retrieve unique values. For example, if <code>fieldName</code> is <code>ip.src</code> , 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 <code>fieldName</code> 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.
<code>where</code>	This is a <code>where</code> clause which filters the set of sessions for which the unique values are returned. For example, if the <code>fieldName</code> is <code>ip.src</code> , and the <code>where</code> clause is <code>ip.src = 192.168.0.0/16</code> , only values in the range of

Parameter	Function
	192.168.0.0 to 192.168.255.255 are returned. For information on the <code>where</code> clause syntax, see <i>Where Clauses</i> .
<code>size</code>	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.
<code>id1, id2</code>	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 <code>id1</code> and <code>id2</code> parameters to a limited range of the meta database is very important to running searches quickly on large collections.
<code>flags</code>	Flags control how the values are sorted and totaled. Flags are described in the following Values Flags section.
<code>threshold</code>	Setting the <code>threshold</code> parameter allows the <code>values</code> 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 <code>threshold</code> parameter is omitted or set to 0, this optimization is not used.
<code>aggregateFunction</code>	Optional parameter used to change the default behavior from counting sessions, packets, or size to counting or summing the numeric field defined by <code>aggregateFieldName</code> . Both parameters must be specified when either is defined. Pass either <code>sum</code> or <code>count</code> to specify which behavior to perform.
<code>aggregateFieldName</code>	The meta field on which to perform the <code>aggregateFunction</code> . Both <code>aggregateFunction</code> and <code>aggregateFieldName</code> parameters must be specified when the <code>aggregate</code> flag is set. Performing a <code>values</code> call using one of the aggregate functions can be significantly slower than a <code>values</code> call that collects totals of sessions, packets, or size.

Parameter	Function
	The reason for this is that each session that matches the <code>where</code> 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 <code>values</code> call is linearly proportional to the number of sessions that match the <code>where</code> clause.
<code>min, max</code>	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 conjunction with the flags <code>sort-value</code> , <code>sort-ascending</code> such that the highest value returned would be used in a subsequent call as the <code>min</code> parameter value. The values are exclusive. If <code>min="rsa"</code> was specified and <code>rsa</code> was a valid value, <code>rsa</code> would not be returned; instead, the next highest value would be returned.
<code>search</code>	Text search pattern to be used to further refine the <code>where</code> parameter

values Flags

The `flags` parameter controls how the `values` call operates. There are three groups of flags that correspond to the different modes of operation as shown in the following table.

Flag	Description
<code>sessions, size, packets</code>	The <code>values</code> call allows you to specify one of these flags to determine how the total for each value is calculated. If the flag is <code>sessions</code> , the <code>values</code> call returns a count of sessions that contain each value. If the flag is <code>size</code> , the <code>values</code> 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 <code>packets</code> , the values

Flag	Description
	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, sort-value	These flags control how results are sorted. If the flag is <code>sort-total</code> , the result set is sorted in order of the totals collected. If the flag is <code>sort-value</code> , the results are returned in order of the sorting order of the values.
order-ascending, order-descending	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
database-scan	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, clear-cache	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 requests unique values for `ip.src`. It requests 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 specifies a bucketed indexed meta, the system will only return the bucket values present within the rest of the criteria. This has the side effect 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 on.

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.

Suggestion Mode

The values call has an additional execution mode that is used to provide suggested search values. In this mode of operation, the values call only identifies unique values stored 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 flags. The suggestion mode does utilize the `where` parameter to refine suggestions, but it only utilizes the time range clause if provided. Other portions of the `where` clause are not utilized to refine suggestion.

Suggestion mode is enabled by setting the `suggest` flag in the `flags` parameter.

Suggestion mode gives special meaning to the `min` parameter. The `min` parameter can contain the starting point for the suggested values. The return values of suggest mode will only include values that start 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 field 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 specified is combined with the search filter as if they were joined with an AND operation.

Searches through the `values` API are always done against indexed meta, in case-insensitive mode.

The `values` API only operates on index entries for most requests, in order to provide fast 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 instead of narrowing the search to exact matches.

/sys

The system tree holds the main functionality related to overall service configuration, scheduler and access to the Stats Database.

The following is a brief overview of the system 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)

Parameter	Description
id	ID of CA certificate to remove (only valid with op=delete)

Manipulate the certificate authority (CA) list used by this service. The Certificate Authority list is used to verify the authenticity of peer certificates.

You must specify what operation to perform on the certificate authority list: list, delete, or add

The **op=list** command will output a list of the known certificate authority certificates.

The **op=delete** command will remove a certificate from the certificate authority list. The delete command must be accompanied by an id parameter. The id refers to the id of the certificate authority certificate to remove from the certificate authority list. Use the list command to see a list of certificate authority id's.

The **op=add** command will upload an additional certificate to the service, and add it to the certificate authority list. This command can only be invoked programmatically. The REST interface provides a simple HTML interface to upload a certificate authority certificate using the add operation. This is available at `/sys/caupload` on the REST interface.

/sys fileEdit

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 files that can be edited by the user. This includes Lua parsers, custom index files, scheduler jobs and a few other miscellaneous files. The workflow 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 file must 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
--file-data="/Users/user/Documents/index-concentrator-
custom.xml"
```

The **op=create type=parser filename=MyLuaParser.lua** command can be used to add new Lua parsers to the system. This will not automatically reload the parser.

To delete files, you can use **op=delete filename=<filename 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 certificate list used by this service. The peer certificate list is used to grant super-user, or "trusted" access to a connection. The if the client provides a certificate that matches one of the certificates in the peer certificate list, the client is trusted. The peer certificate is *not* trusted based on certificate chain trust. Trust is only granted if the client's certificate is explicitly added to the peer list.

You must specify what operation to perform on the peer certifiacate list: list, delete, or add

The **op=list** command will output a list of the known peer certificates.

The **op=delete** command will remove a certificate from the peer certificate list. The delete command must be accompanied by an id parameter. The id refers to the id of the peer certificate to remove from the peer list. Use the list command to see a list of peer id's.

The **op=add** command will upload an additional certificate to the service, and add it to the peer certificate list. This command can only be invoked programmatically. The REST interface provides a simple HTML interface to upload a peer certificate using the add operation. This is available at `/sys/trustpeer` 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 configuration to disk. This command is redundant because the system automatically saves changes within a couple of minutes and always on shutdown.

/sys servCert

Display the public certificate 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 files and then exit cleanly. Typically, the service will restart based on the systemd configuration.

/sys statHist

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 historical stats from the stats db. Don't send time1/time2 to get bounding times about stats 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 stats 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 first second has a bunch of stats, but the next 2 do not? What happens is the stat db only stores values that have changed since the last update. However, that would be a problem for stats that rarely change if you wanted to plot them in a chart, so every 5 mins on the 5 min mark, *every* stat is stored again. When the service first starts, it also stores every stat in the db, then follows with only stat changes until the next 5 min mark. Bottom line: when charting, it's best to make sure time1 is on a 5 minute mark.

You can also limit the stats you are interested 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 stats under the whole /sys tree and the immediate stats under /logs/stats.

If you would like to see every stat value with every time period, you can use the parameter **showAll=true**. This parameter will carry forward every stat 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 best 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 stat values that didn't change are included.

A very useful parameter is **onChange=<stat pathname>[,<stat pathname>]**. This will only return results when the listed stats change. Only one stat 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 stat **capture.dropped** changed twice in the 10 minute period requested. 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.

Configuration of the Stats Database

There are 3 configuration nodes that affect how the stat database works. They are all under `/sys/config`.

- **stat.compression** - Determines whether or not the records in the stat db are compressed or not. The values can be none or gzip. If you enable gzip 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

Configuration

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 reflecting the complexity of the `content` attribute (pattern) of the header. The more generic a header is the lower the match priority. The more distinct 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 against the header with this highest 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 increase the likelihood of a single device match for strong 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 configurable 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 flagged by the content elements <HEADER> or <MESSAGE> with the attribute `discoverable="mapped"` trigger a temporary event source mapping which directs future parsing to the matched device type. A mapped source is valid for a configured amount of time. Each time a subsequent match is flagged 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 prefixed `com.rsa.netwitness.carlos.nextgen.Event` protobuf objects on port 50202 (default). The connection must be initiated with byte string:

```
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 fields in the `Event` instances:

All fields 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 persistence 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 stat 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 Persistence

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 timestamps. This information is derived from the IANA Time Zone Database installed on the service host. As timezone abbreviations do not necessarily represent a single GMT offset, **tzinfo** also provides a means to override the definition of an abbreviation as well as define 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 offsets 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 **dstAbbr** indicates that a timezone does not currently observe daylight savings time. Also note that the offsets returned here reflect 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 offset and the **Abbreviation Override** offset if one exists.

```
tzinfo op=abbrs
```

For each abbreviation definition, 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 offset in distinct timezones.

```
tzinfo op=duplicates
```

For each abbreviation definition that has multiple offset definitions 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 offset or timezone explicitly set will not be displayed as a duplicate.

Overridden Abbreviations

View all abbreviations that have had GMT offset 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 whether 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 definition of an existing abbreviation. Adding an existing abbreviation override will update that abbreviation.

An abbreviation can be defined as a specific GMT offset. The following creates the abbreviation `EQT` with a GMT offset of -5 hours:

```
tzinfo op=add abbr=EQT gmtoffset=-0500
```

A duplicate abbreviation can be overridden from the default definition. The following changes the default offset of `CST` from -6 hours to -5 hours as defined by the timezone `Cuba`:

```
tzinfo op=add abbr=CST tzname=Cuba
```

The timezone must be known (see **Timezone Names**) and define the provided abbreviations as standard 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 Configuration Utilities

The steps for attaching storage 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 Configuration Commands

Each of the commands listed 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 storage

- **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 storage 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 dflAlloc

Set up Storage-related default configuration for a Service

This command configures services on this host to use storage in a volume group. It does this by creating a 'defaults' file which will be read by the service the first time, **and only the first time** the service starts.

You must provide the name of the service to configure, and the volume group to assign to the service.

Currently this command only supports the Decoder, Log Decoder, or Concentrator service on a hybrid.

The volume group must be one of the unused volume groups on this host. The volume group **must** be a volume group that was created with the `partNew` command. To see information about the volume groups on this host, use the `vgs` command.

By default, this command does not make changes to the defaults file. To actually make changes to the defaults file, add the parameter `commit=true`.

Note that this command does not do anything for services that are already installed and running. For that scenario, the `srvAlloc` command might be useful.

This command assigns storage in the `/database` and `/index` nodes. In all cases it applies a configuration that allows the service to use **95%** of the space on the volume. 5% of the space on the volume is reserved for rollover.

Hybrid Concentrator Allocation

For Concentrator devices, you must assign the `hybrid-concentrator` and `index` volume groups.

The `hybrid-concentrator` volume group is allocated as follows:

- 30 GB reserved for the Concentrator "root" directory `/var/netwitness/concentrator`
- 91% of space for Meta database
- 9% of space for Session database

The `index` volume group is allocated as follows:

- 100% of the space for Index

Hybrid Decoder Allocation

For Decoder devices, you must assign the `hybrid-decoder-meta` and `packet` volume groups.

The `hybrid-decoder-meta` volume group is allocated as follows:

- 30 GB reserved for the Decoder "root" directory `/var/netwitness/decoder`
- 87% of space for Meta database
- 9% of space for Session database
- 4% of space for index

The `packet` volume group is allocated as follows:

- 100% of the space for Packet database

Hybrid Log Decoder Allocation

For Decoder devices, you must assign the `hybrid-logdecoder-meta`, `logpacket`, and `endpoint-log-hybrid` volume groups.

The `hybrid-decoder-meta` volume group is allocated as follows:

- 30 GB reserved for the Log Decoder "root" directory `/var/netwitness/logdecoder`
- 87% of space for Meta database
- 9% of space for Session database
- 4% of space for index

The `logpacket` volume group is allocated as follows:

- 100% of the space for Packet database

The `endpoint-log-hybrid` volume group is allocated as follows:

- 50% of the space is reserved for Endpoint services
- 50% of the space is used by the Packet database

The `log-indexed-decoder` volume group is allocated as follows:

- 30 GB reserved for the Log Decoder root directory `/var/netwitness/logdecoder`
- 6% of space for Session database
- 47% of space for Meta database
- 47% of space for Packet database

Note that the above percentages are portions of the **usable** space left over after accounting for the reserved space.

`/appliance partClr`

Uninitialize a block storage device

This command clears data from a block storage 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 storage device

This command prepares a storage 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 must provide the block device name, for example, "sdc" or "/dev/sdc". Use the `devlist` command to locate unused block devices. You must provide a name for the service that will

be used with the storage, 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. Instead it prints a report of the actions that will be taken. To actually make the changes to the system, 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 must be created before the `decoder` volume, because it holds the small filesystem 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	Filesystems Created
decoder	decoder	packetdb
decoder	decodersmall	decoder root, index, sessiondb, metadb
decoder	hybrid-decoder-meta	decoder root
decoder	packet	packetdb

Service	Volume Name	Filesystems Created
logdecoder	logdecoder	packetdb
logdecoder	logdecodersmall	logdecoder root, index, sessiondb, metadb
logdecoder	hybrid-logdecoder-meta	logdecoder root
logdecoder	logpacket	packetdb
logdecoder	endpoint-log-hybrid	mongo, packetdb
logdecoder	log-indexed-decoder	logdecoder root
logdecoder	logindex	logdecoder index
concentrator	concentrator	concentrator root, metadb, sessiondb
concentrator	hybrid-concentrator	concentrator root
concentrator	index	index
archiver	archiver	database

Volume sizing

Volume	Filesystem	Mount Point	Size
decodersmall	decoroot	/var/netwitness/decoder	10 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
logdecodersmal l	decoroot	/var/netwitness/logdecoder	10 GB
logdecodersmal l	index	/var/netwitness/logdecoder/index	30 GB
logdecodersmal l	sessiondb	/var/netwitness/logdecoder/sessiondb	600 GB

Volume	Filesystem	Mount Point	Size
logdecodersmall	metadb	/var/netwitness/logdecoder/metadb	all remaining free space on logdecodersmall volume
hybrid-logdecoder-meta	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

Volume	Filesystem	Mount Point	Size
concentrator	metadb	<i>/var/netwitness/concentrator/metadb</i>	all remaining free space on concentrator volume
hybrid-concentrator	root	<i>/var/netwitness/concentrator</i>	all space on hybrid-concentrator volume
index	index	<i>/var/netwitness/concentrator/index</i>	all space on index volume
archiver	database	<i>/var/netwitness/archiver/database</i>	all space on archiver volume
endpoint-log-hybrid	mongo	<i>/var/netwitness/mongo</i>	50% of free space on endpoint-log-hybrid volume
endpoint-log-hybrid	packetdb	<i>/var/netwitness/logdecoder/packetdb</i>	all remaining free space on endpoint-log-hybrid volume

***/appliance* srvAlloc**

Allocate Storage to a Service

This command configures services on this host to use storage in a volume group.

You must provide the name of the service to configure, and the volume group to assign to the service.

The name of the service must be one of the services installed on this host. To see a list of services installed on this host, use the `srvList` command. This command communicates with the service using a local connection to the service's SSL port.

The volume group must be one of the unused volume groups on this host. The volume group **must** be a volume group that was created with the `partNew` command. To see information about the volume groups on this host, use the `vgs` command.

By default, this command does not make changes to the service. To actually make changes to the service, add the parameter `commit=true`.

This command will restart the specified service after making changes.

Decoder, Log Decoder, and Concentrator Allocation

This command assigns storage in the `/database` and `/index` nodes under these services. For each path entry, it applies a configuration that allows the service to use 95% of the usable space on the volume.

For Decoder devices, you must assign the `decodersmall` volume group to Decoder before you assign the `decoder` volume group.

For Log Decoder devices, you must assign the `logdecodersmall` volume group to Log Decoder before you assign the `logdecoder` volume group.

For Concentrator devices, you must assign the `concentrator` volume group to the Concentrator before you assign the `index` volume group.

Important note: volume groups must be assigned in order. For example, `decodersmall` must be allocated before `decodersmall0`.

Archiver Storage Allocation

When allocating storage to an Archiver, a configuration entry is added in `/archiver/config/mounts.dir`. Note that this does **not** actually allocate the storage to a collection in the Archiver. Use the Selective Retention configuration pages in the NetWitness Administration user interface to assign storage from the Archiver's `mounts.dir` parameter to collections.

Hybrid Allocation

On a hybrid, only the `decoder` and `concentrator` volumes should be extended. For example on a hybrid expansion enclosure, one RAID volume would be assigned to the Decoder service's `decoder` volume, and the other would be assigned to the Concentrator service's `concentrator` volume.

/appliance srvFree

Remove storage from a service configuration

This command reconfigures other services on this host to **not** use storage in a volume group.

You must provide the name of the service to configure, and the volume group to remove from it. You can view the services on the host using the `srvList` command, which will also show the filesystems that are currently configured on the host.

The following steps are performed:

1. Check that any of the referenced filesystems are empty, or contain empty collections. If it does not appear that the filesystems are empty, then the command will not proceed.
2. If the filesystem path removed from the service is the last storage path configured, it creates a temporary directory under `/var/netwitness/temp` and assigns that as the last storage path.
3. If the configuration of the service changes, then the service is restarted.

Note: This command will not allow you to remove the last volume group from a service, since it is needed to store the working directory of the service. To fully remove the last volume group from a service, first uninstall the service's RPM and then use `partClr` on the remaining unused storage device.

/appliance vgs

List Volume Groups

This command enumerates all the volume groups on this host. In addition, it displays the physical volumes that the volume group consists of, and the logical volumes within the volume group.