NETROPY 4.0

RESTFUL API

QUICK REFERENCE GUIDE

APPOSITE
TECHNOLOGIES

1 OVERVIEW

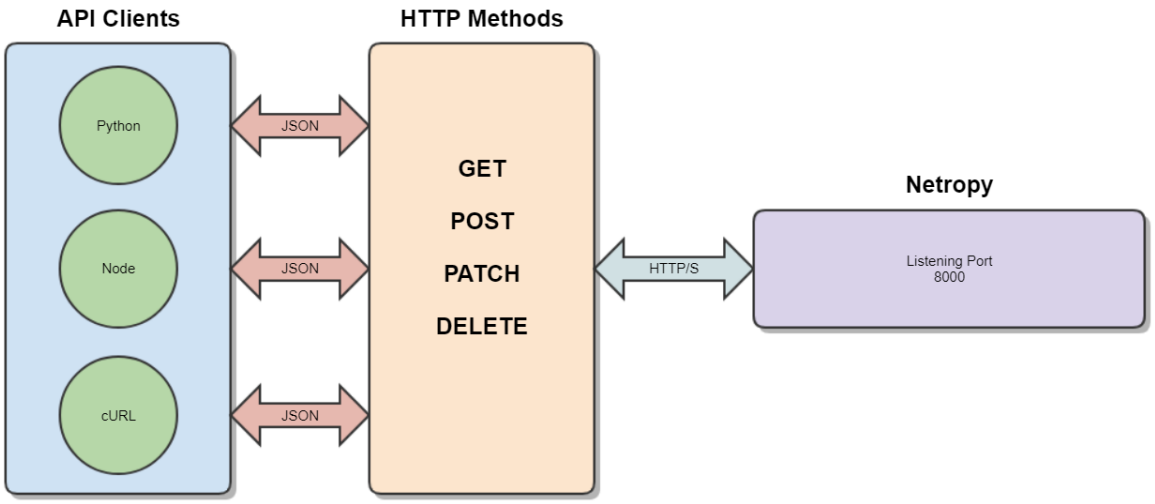
Objective: To use the RESTful API to interface with the Apposite Netropy WAN emulation products.

Assumptions: The reader of the document is familiar with using the Netropy product. User also understand RESTful API and how to interface using curl, libcurl, or any other programing language that can interface to REST. Reader understands that this guide is not a complete API document but a guide to get started.

This Example: This document uses the cURL command line as an example to interface to the RESTful API. Using curl most programmers should be able to translate the commands to their preferred programing language.

API Information:

Port: 8000
Format: JSON
Commands: GET, POST, PATCH, DELETE



2 GET NETROPY INFORMATION

Gather Administration Information:

curl -X GET "http://192.168.184.100:8000/apposite-netropy-system:admin" -H "accept: application/json"

Response:


```
{
  "apposite-netropy-system:admin": {
    "network-settings": {
      "revision": 0,
      "hostname": "netropy",
      "dhcp": true,
      "address": "10.0.0.10",
      "netmask": "255.0.0.0",
      "ipv6-enabled": false,
      "dns": [
        "8.8.8.8",
        "8.8.4.4"
      ],
      "ntp": [],
      "ldap": false
    },
    "users": {
      "user": [
        {
          "name": "admin"
        }
      ]
    },
    "ethernet-settings": {
      "port": [
        {
          "id": "PORT_1",
          "hardware": "ETHERNET_VIRTUAL",
          "duplex": "autonegotiate",
          "flow-control": "none"
        },
        {
          "id": "PORT_2",
          "hardware": "ETHERNET_VIRTUAL",
          "duplex": "autonegotiate",
          "flow-control": "none"
        }
      ]
    },
    "network-status": {
      "address": {
```

```
      "value": "192.168.184.100/255.255.255.0",
      "dhcp": false
     },
     "gateway": {
      "value": "192.168.184.2",
      "dhcp": false
     },
     "ipv6": {
      "dhcp": false
     },
     "domain": {
      "dhcp": false
     },
     "dns": {
      "value": "8.8.8.8 8.8.4.4",
      "dhcp": false
     },
     "ntp": {
      "value": "time.google.com",
      "dhcp": false
     }
    },
    "firmware": {
     "version": "4.0-beta-rc34"
    },
    "license-key": {
     "sn": "VN-42A7A21C9A",
     "expiration": "No Operational License",
     "bandwidth": 0
    }
   }
}
```


Gather Network Information:

curl -X GET "http://192.168.184.100:8000/apposite-netropy-system:admin/network-status" -H "accept: application/json"

Response:


```
{
  "network-status": {
    "address": "192.168.184.100",
    "netmask": "255.255.255.0",
    "gateway": "192.168.184.1",
    "domain": "apposite-tech.com",
    "dns": [
      "8.8.8.8",
      "192.168.184.1"
    ],
    "ntp": [
      " us.pool.ntp.org"
    ]
  }
}
```


Gather License Information:

curl -X GET "http://192.168.184.100:8000/apposite-netropy-system:admin/license-key" -H "accept: application/json"

Response:


```
{
  "license-key": {
    "sn": "VN-42A7A21C9A",
    "expiration": "2019-06-01 23:59:59 UTC",
    "bandwidth": 1000000000
  }
}
```

3 PATH COMMANDS

Gather Path Information:

curl -X GET "http://192.168.184.100:8000/apposite-wan-emulator:topology/path" -H "accept: application/json"

Response:


```
{
  "path": [
    {
      "id": 1,
      "engine": 1,
      "index": 4,
      "label": "Path 1",
      "capacity": "10G",
      "source": {
        "outbound": {
          "mode": "single",
          "bandwidth": {
            "rate": "1",
            "metric": "Gbps"
          }
        }
      },
      "destination": {
        "outbound": {
          "mode": "single",
          "bandwidth": {
            "rate": "1",
            "metric": "Gbps"
          }
        }
      },
      "wan": {
        "source-to-destination": {
          "delay": {
            "method": "constant",
            "reordering": false,
            "constant": {
              "latency": "35"
            }
          }
        },
        "destination-to-source": {
          "delay": {
            "method": "constant",
            "reordering": false,
            "constant": {
              "latency": "35"
```

```
        }
       }
      }
     },
     "apposite-netropy-system:capacity": "1G"
    }
   ]
}
```


Explanation of feedback:

- Path ID: This is the number of the path. You will use this number when creating endpoints.
- Engine: This path was created on the first engine of the emulator
- Label: Is the name of the path in this example the name is “Path 1”
- Source (Port 1) is set to 1Gb
- Destination (Port 2) is set to 1Gb
- WAN
    - Constant delay of 35ms in both directions
- The Netropy is rated for 1Gb speeds

Creating A Simple Path:

Example JSON file that creates a path with the following settings:

- Engine number: 1
- Path number: 2
- Named: Lab Path 2
- Bandwidth: 1Gbps

Example of path_create.json file:


```
{
 "engine": 1,
 "id": 2,
 "label": "Lab Path 2",
 "source": {
    "outbound": {
     "mode": "single",
     "bandwidth": {
      "rate": 1,
      "metric": "Gbps"
    }
   }
 },
   "destination": {
    "outbound": {
     "mode": "single",
     "bandwidth": {
      "rate": 1,
      "metric": "Gbps"
     }
    }
   }
}
```

Curl command:

Save this file as path_create.json and use curl to create the path:

curl -X POST "http://192.168.184.100:8000/apposite-wan-emulator:topology/path" -H "content-type: application/json" -d@path_create.json

Create A Path with Jitter and Bandwidth Throttling:

- Engine number: 1
- Path number: 4
- Named: Delay Test
- Bandwidth:
    - 200Mps on port 1
    - 50Mbps on port 2
- Delay Constant:
    - 50ms on source port
    - 70ms on destination port

Example of 250mbps.json:


```
{
 "engine": 1,
 "id": 4,
 "label": "Delay Test",
 "source": {
    "outbound": {
     "mode": "single",
     "bandwidth": {
      "rate": 200,
      "metric": "Mbps"
    }
   }
 },
  "destination": {
   "outbound": {
    "mode": "single",
    "bandwidth": {
     "rate": 50,
     "metric": "Mbps"
    }
   }
  },
  "wan": {
   "source-to-destination": {
    "delay": {
     "method": "constant",
     "reordering": false,
     "constant": {
      "latency": 50
     }
    }
   },
   "destination-to-source": {
```

```
      "delay": {
       "method": "constant",
       "reordering": false,
       "constant": {
        "latency": 70
       }
      }
     }
    }
}
```

Curl Command:

Save this as 250mbps.json and use the following curl command to submit:

curl -X POST "http://192.168.184.100:8000/apposite-wan-emulator:topology/path" -H "content-type: application/json" -d@250mbps.json

Deleting a Path:

This example deletes path 2 on the Netropy:

curl -X DELETE "http://192.168.184.100:8000/apposite-wan-emulator:topology/path/2" -H "accept: application/json"

Response:

```
{
 "path": {
  "id": 2,
  "engine": 1,
  "index": 9,
  "label": "Path 2",
  "capacity": "10G",
  "source": {
   "outbound": {
    "mode": "single",
    "bandwidth": {
     "rate": "1",
     "metric": "Gbps"
    }
   }
  },
  "destination": {
   "outbound": {
    "mode": "single",
    "bandwidth": {
     "rate": "1",
     "metric": "Gbps"
    }
   }
  },
  "wan": {
   "source-to-destination": {
    "delay": {
     "method": "constant",
     "reordering": false,
```

```
          "constant": {
            "latency": "35"
          }
        }
      },
      "destination-to-source": {
        "delay": {
          "method": "constant",
          "reordering": false,
          "constant": {
            "latency": "35"
          }
        }
      }
    },
    "apposite-netropy-system:capacity": "1G"
  }
}
```

Updating a Path:

- Updating Source to destination latency to: 35 ms
- Updating Destination to source latency to: 45 ms

Example update.json file:

```
{
  "wan":{
    "source-to-destination":{
      "delay":{
        "method":"constant",
        "reordering":false,
        "constant":{
          "latency":35
        }
      }
    },
    "destination-to-source":{
      "delay":{
        "method":"constant",
        "reordering":false,
        "constant":{
          "latency":45
        }
      }
    }
  }
}
```

Curl Command:

```
curl -X PATCH "http://192.168.184.100:8000/apposite-wan-emulator:topology/path/1" -H "content-type: application/json" -d@update.json
```

4 Endpoint Commands

Gather Endpoint Information:

curl -X GET "http://192.168.184.100:8000/apposite-wan-emulator:topology/endpoint" -H "accept: application/json"

Response:


```
{
  "endpoint": [
    {
      "id": "05e85393-3764-45d8-a9a6-cc07fcac5800",
      "engine": 1,
      "path": "1",
      "label": "Client-1",
      "port": 1,
      "address": [
        "10.10.1.1"
      ],
      "protocol": "tcp"
    },
    {
      "id": "963d3f7f-c4e2-4822-b339-70790f19616c",
      "engine": 1,
      "path": "1",
      "label": "Server-2",
      "port": 2,
      "address": [
        "10.10.2.2"
      ],
      "protocol": "tcp"
    }
}
```


This response shows that there are:

- 2 Endpoints
- On engine 1
- Named Client-1 with:
    - Connection to path 1
    - Port 1
    - With the IP address of: 10.10.1.1
- Named Server-2
    - Connected to path 1
    - Port 2
    - With the IP address of: 10.10.2.2

Creating an Endpoint with an IP Address:

Example JSON file that creates an endpoint with the following settings:

- Name: My Endpoint
- On engine 1
- Path 2
- Port 2
- IP Address: 10.10.10.99

Example of endpoint.json:

```
{
  "label": "My Endpoint",
  "engine": 1,
  "port": 2,
  "address": [
    "10.10.10.99"
  ],
  "path": "2",
  "protocol": "tcp"
}
```

Curl Command:

```
curl -X POST "http://192.168.184.100:8000/apposite-wan-emulator:topology/endpoint" -H "content-type: application/json" -d@server_endpoint
```

Response:

```
{
  "endpoint": [
    {
      "id": "b7dfe4c9-51d8-431e-a4f4-11dc6644b311",
      "engine": 1,
      "path": "2",
      "label": "My Endpoint",
      "port": 2,
      "address": [
        "10.10.10.99"
      ],
      "protocol": "tcp"
    }
  ]
}
```

Creating an Endpoint IP Subnet:

Example JSON file that creates an endpoint for a network of IPS:

- Name: Private 172 Network
- Engine: 1
- Path: 1
- Port: 1
- IP Addresses: 172.16.0.0/16

Example of 172network.json:

```
{
 "label": "Private 172 Network",
 "engine": 1,
 "port": 1,
 "address": [
  "172.16.0.0/16"
 ],
 "path": "1",
 "protocol": "tcp"
}
```

Curl Command:

curl -X POST "http://192.168.184.100:8000/apposite-wan-emulator:topology/endpoint" -H "content-type: application/json" -d@172network.json

Response:

```
{
 "label": "Private 172 Network",
 "engine": 1,
 "port": 1,
 "address": [
  "172.16.0.0/16"
 ],
 "path": "1",
 "protocol": "tcp"
}
```

Creating an Endpoint VLAN Network:

Example JSON file that creates an endpoint for VLAN 4012:

- Name: VLAN 4012 Network
- Engine: 1
- Path: 1
- Port: 1
- VLAN: 4012

Example of 4012vlan.json:

```
{
    "engine": 1,
    "path": "1",
    "label": "VLAN 4012 Network",
    "port": 1,
    "vlan": [
     4012
    ],
    "protocol": "tcp"
}
```

Curl Command:

curl -X POST "http://192.168.184.100:8000/apposite-wan-emulator:topology/endpoint" -H "content-type: application/json" -d@4012vlan.json

Response:


```
{
 "endpoint": [
  {
   "id": "4b6d2a90-8ea1-4356-9e49-5fe409036c4f",
   "engine": 1,
   "path": "1",
   "label": "VLAN 4012 Network",
   "port": 1,
   "vlan": [
    4012
   ],
   "protocol": "tcp"
  }
 ]
}
```


Creating an Endpoint for TCP Port:

Example JSON that creates an endpoint for TCP port 443:

- Name: Port 442 Endpoint
- Engine: 1
- Path: 3
- Port: 1
- Protocol: tcp
- Port: 443

Example of port443endpoint.jason:


```
{
  "engine": 1,
  "path": "3",
  "label": "Port 443 3ndpoint",
  "port": 1,
  "protocol": "tcp",
  "transports": [
   443
  ]
}
```


Curl Command:

curl -X POST "http://192.168.184.100:8000/apposite-wan-emulator:topology/endpoint" -H "content-type: application/json" -d@porte443ndpoint.json

Response:


```
{
 "endpoint": [
  {
   "id": "60c00d74-b4be-4578-8b52-edd68870cad4",
   "engine": 1,
   "path": "3",
   "label": "Port 443 Endpoint",
   "port": 1,
   "protocol": "tcp",
   "transports": [
    443
   ]
  }
 ]
}
```


Deleting an Endpoint:

You will need the “id” of the endpoint you want to delete. To obtain the id please see “Gathering endpoint information.”

```
curl -X DELETE "http://192.168.184.100:8000/apposite-wan-emulator:topology/endpoint/60c00d74-b4be-4578-8b52-edd68870cad4" -H "accept: application/json"
```

Response:


```
{
 "endpoint": {
  "id": "60c00d74-b4be-4578-8b52-edd68870cad4",
  "engine": 1,
  "path": "3",
  "label": "Port 443 Endpoint",
  "port": 1,
  "protocol": "tcp",
  "transports": [
   443
  ]
 }
}
```

5 GATHERING STATISTICS

Gathering Current Statistics:

curl -X POST -H "content-type: application/json" http://192.168.184.100:8000/stat:server/fetch -d@engineone.json

Example of engineone.json:

```
{
  "engine": 1,
  "index": 3
}
```

Response:


```
{
  "timestamp": 1556933638129,
  "engine": 1,
  "index": 3,
  "first": 58807,
  "start": 58994,
  "end": 58994,
  "record": [
    {
      "sequence": "58994",
      "source": {
        "frames": 0,
        "bytes": 0,
        "drops": 0,
        "outbound": {
          "frames": 0,
          "bytes": 0,
          "queue": {
            "frames": 0,
            "bytes": 0,
            "drops": 0
          },
          "background": {
            "frames": 0,
            "bytes": 0,
            "drops": 0
          }
        },
        "inbound": {
          "frames": 0,
          "bytes": 0,
          "queue": {
            "frames": 0,
            "bytes": 0,
```

```
        "drops": 0
       },
       "background": {
        "frames": 0,
        "bytes": 0,
        "drops": 0
       }
      }
     },
     "destination": {
      "frames": 0,
      "bytes": 0,
      "drops": 0,
      "inbound": {
       "frames": 0,
       "bytes": 0,
       "queue": {
        "frames": 0,
        "bytes": 0,
        "drops": 0
       },
       "background": {
        "frames": 0,
        "bytes": 0,
        "drops": 0
       }
      },
      "outbound": {
       "frames": 0,
       "bytes": 0,
       "queue": {
        "frames": 0,
        "bytes": 0,
        "drops": 0
       },
       "background": {
        "frames": 0,
        "bytes": 0,
        "drops": 0
       }
      }
     },
     "wan": {
      "source-to-destination": {
       "drops": 0,
       "duplicates": 0,
       "reorders": 0,
       "corruptions": 0
      },
      "destination-to-source": {
       "drops": 0,
       "duplicates": 0,
       "reorders": 0,
       "corruptions": 0
      }
     }
    }
   ]
}
```

Gathering Statistics Between Times:

curl -X POST -H "content-type: application/json" http://192.168.184.100:8000/stat:server/fetch -d@times.json

Example of times.json:

Times are unix or utime in seconds.

```
{
 "engine": 1,
 "index": 1,
 "from": 1558467273,
 "to": 1558467288
}
```

Response:

This will have the statistics for each second within the time you specified. Every second will look something like this:


```
{
 "engine": 1,
 "index": 1,
 "from": 1558467442,
 "to": 1558467443,
 "record": [
  {
   "sequence": "12392",
   "source": {
    "frames": 70964,
    "bytes": 107658824,
    "drops": 0,
    "outbound": {
     "frames": 70963,
     "bytes": 107657306,
     "queue": {
      "frames": 232,
      "bytes": 352176,
      "drops": 0
     },
     "background": {
      "frames": 0,
      "bytes": 0,
      "drops": 0
     }
    },
    "inbound": {
     "frames": 0,
     "bytes": 0,
     "queue": {
      "frames": 0,
      "bytes": 0,
      "drops": 0
     },
     "background": {
      "frames": 0,
      "bytes": 0,
      "drops": 0
```

```
      }
     }
    },
    "destination": {
     "frames": 168,
     "bytes": 11760,
     "drops": 0,
     "inbound": {
      "frames": 0,
      "bytes": 0,
      "queue": {
       "frames": 0,
       "bytes": 0,
       "drops": 0
      },
      "background": {
       "frames": 0,
       "bytes": 0,
       "drops": 0
      }
     },
     "outbound": {
      "frames": 168,
      "bytes": 11760,
      "queue": {
       "frames": 0,
       "bytes": 0,
       "drops": 0
      },
      "background": {
       "frames": 0,
       "bytes": 0,
       "drops": 0
      }
     }
    },
    "wan": {
     "source-to-destination": {
      "drops": 0,
      "duplicates": 0,
      "reorders": 0,
      "corruptions": 0
     },
     "destination-to-source": {
      "drops": 0,
      "duplicates": 0,
      "reorders": 0,
      "corruptions": 0
     }
    },
    "timestamp": 1558467442
   },
   {
    "sequence": 12393,
    "source": {
     "frames": 70343,
     "bytes": 106708946,
     "drops": 0,
     "outbound": {
      "frames": 70344,
      "bytes": 106710464,
```

```
    "queue": {
     "frames": 516,
     "bytes": 783200,
     "drops": 0
    },
    "background": {
     "frames": 0,
     "bytes": 0,
     "drops": 0
    }
   },
   "inbound": {
    "frames": 0,
    "bytes": 0,
    "queue": {
     "frames": 0,
     "bytes": 0,
     "drops": 0
    },
    "background": {
     "frames": 0,
     "bytes": 0,
     "drops": 0
    }
   }
  },
  "destination": {
   "frames": 177,
   "bytes": 12390,
   "drops": 0,
   "inbound": {
    "frames": 0,
    "bytes": 0,
    "queue": {
     "frames": 0,
     "bytes": 0,
     "drops": 0
    },
    "background": {
     "frames": 0,
     "bytes": 0,
     "drops": 0
    }
   },
   "outbound": {
    "frames": 177,
    "bytes": 12390,
    "queue": {
     "frames": 0,
     "bytes": 0,
     "drops": 0
    },
    "background": {
     "frames": 0,
     "bytes": 0,
     "drops": 0
    }
   }
  },
  "wan": {
   "source-to-destination": {
```

```
        "drops": 0,
        "duplicates": 0,
        "reorders": 0,
        "corruptions": 0
       },
       "destination-to-source": {
        "drops": 0,
        "duplicates": 0,
        "reorders": 0,
        "corruptions": 0
       }
      },
      "timestamp": 1558467443
     }
    ]
}
```


Gathering Statistics from a Specific Time to Present:

curl -X POST -H "content-type: application/json" http://192.168.184.100:8000/stat:server/fetch -d@from_time.json

Example of from_time.json:


```
{
  "engine": 1,
  "index": 1,
  "from": 1558467442
}
```