

Measurement Archive REST Interface v1.0

June 18, 2014

[Overview](#)

[JSON Types](#)

[Base JSON Types](#)

[BaseEventTypeDescriptor](#)

[BaseMetadata](#)

[BaseP2PMetadata](#)

[BaseSummaryDescriptor](#)

[BaseTimeSeriesDatum<TYPE>](#)

[BaseTimeSeriesConstructor<TYPE>](#)

[Metadata JSON Types](#)

[BaseIPPacketMetadata](#)

[BaseTimeMetadata](#)

[BasePacketSampleMetadata](#)

[BaseTCPMetadata](#)

[BaseThroughputMetadata](#)

[BasePacketTraceMetadata](#)

[Datum JSON Types](#)

[FailureDescriptor](#)

[HistogramItem](#)

[PercentageConstructor](#)

[PacketTraceItem](#)

[StatisticalSummary](#)

[SubintervalDatum<TYPE>](#)

[Event Types](#)

[Messaging](#)

[URL Parameter Sets](#)

[Time Parameters](#)

[Metadata Search Parameters](#)

[Queries](#)

[General Queries](#)

[Listing/Searching Metadata](#)

[Querying a Single Metadata Object](#)

[Querying Event Type Descriptors](#)

[Query Base Data](#)

[Querying Summarization Descriptors](#)

[Querying Summary Data](#)

[Query Examples](#)

[Querying Failures](#)

- [Querying a One-way Delay Histogram](#)
- [Querying a Round Trip Time \(RTT\) Histogram](#)
- [Querying a TTL Histogram](#)
- [Querying Packet Duplicates](#)
- [Querying a Packet Loss Rate Summary](#)
- [Querying a Packet Trace](#)
- [Querying Packets Lost](#)
- [Querying Packets Sent](#)
- [Querying a One-Way Delay Statistical Summary](#)
- [Querying a Round Trip Time \(RTT\) Summary](#)
- [Querying Throughput](#)
- [Querying Throughput Subintervals](#)
- [Querying Time Error Estimates](#)

[Creates and Updates](#)

- [Registering New Metadata](#)
- [Writing to Multiple Event Types](#)
- [Writing to a Single Event Type](#)

[Security Considerations](#)

- [Read Access](#)
- [Write Access](#)

[Appendix A: Searching by Hostname](#)

[Appendix B: Percentage Type](#)

Overview

This document describes the REST interface for accessing various data types in the measurement archive.

JSON Types

Base JSON Types

BaseEventPropertyDescriptor

Parent Type(s): None

Description: An object that describes an event type

Keys:

Name	Type	Description
event-type	string	String indicating the type of data
base-uri	URI	The URI where the data of this event type can be queried
summaries	Array of BaseSummaryDescriptor objects	A list of supported summaries for this event type.
time-updated	UNIX timestamp	Unix timestamp indicating when the event type last received new data. A value of null indicates that it has no data.

Example:

```
{
  event-type: "histogram-owdelay",
  base-uri: /perfsonar/archive/ABC123/histogram-owdelay/base,
  time-updated: 1397242422,
  summaries: [
    {
      uri: /perfsonar/archive/ABC123/histogram-owdelay/aggregations/3600,
      summary-type: "aggregation",
      summary-window: 3600,
      time-updated: 1397242422,
    },
    {
      uri: /perfsonar/archive/ABC123/histogram-owdelay/statistics/3600,
      summary-type: "statistics",
      summary-window: 3600,
      time-updated: 1397242422,
    }
  ]
}
```

```
}  
]
```

BaseMetadata

Parent Type(s): None

Description: This base type for all metadata objects. All metadata objects may add additional keys but **MUST** contain the keys specified in this definition.

Keys:

Name	Type	Description
uri	uri	The URI to retrieve this object. Must be set unless being used in a request to create a new object.
metadata-key	string	A unique string that identifies this metadata. Must be set unless being used in a request to create a new object.
subject-type	string	The type of subject represented in this metadata. This document only defines 'point-to-point'.
event-types	array of BaseEventTypeDescriptors	List of event types stored by the test this metadata represents

Example:

```
{  
  "uri": "/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B",  
  "metadata-key": "DE1DADF052BA11E3AE3667D36188709B",  
  "subject-type": "point-to-point",  
  "event-types": [  
    {  
      event-type: "packet-count-lost",  
      base-uri:  
"/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B/packet-count-lost",  
      time-updated: 1397242422,  
    }  
  ]  
}
```

BaseP2PMetadata

Parent Type(s): BaseMetadata

Description: The base type for all point-to-point measurements. The 'subject-type' must be 'point-to-point' if this type of metadata is used.

Keys:

Name	Type	Description
source	ip-string	The IPv4 or IPv6 address used as the source of the measurement (See Appendix A: Searching by Hostname)
destination	ip-string	The IPv4 or IPv6 address used as the destination of the measurement (See Appendix A: Searching by Hostname)
measurement-agent	ip-string	The IPv4 or IPv6 address of the host that initiated the measurement. For example, the host where the bwctl process runs which may be the source, destination or a third-party host. (See Appendix A: Searching by Hostname)
input-source	string	The source exactly as it is passed to the tool. This may be the same IP address as 'source' or it may be a hostname.
input-destination	string	The destination exactly as it is passed to the tool. This may be the same IP address as 'destination' or it may be a hostname.
tool-name	string	Identifies the tool used to perform the measurement. In the general case, clients should query on even type over this field as the underlying tool may change over time. Examples

		are 'owamp/powstream' and 'bwctl/iperf3', 'nuttcp'
Example: <pre> { "uri": "/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B", "metadata-key": "DE1DADF052BA11E3AE3667D36188709B", "subject-type": "point-to-point", "tool-name": "owamp/powstream", "source": "198.124.238.49", "destination": "198.129.252.142", "measurement-agent": "198.124.238.49", "input-source": "bnl-owamp.es.net", "input-destination": "lbl-owamp.es.net", "event-types": [{ event-type: "packet-count-lost", base-uri: "/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B/packet-count-lost", time-updated: 1397242422, }] } </pre>		

BaseSummaryDescriptor

Parent Type(s): None Description: An object that describes an event type Keys:		
Name	Type	Description
uri	URI	The URI where the data of this event type can be queried
summary-type	string	The type of summarization. Valid values are: <ul style="list-style-type: none"> • aggregation - These values collect multiple base values and combine them in a context specific way over the given summary-window • average - the statistical average of numerical values in the summary window

		<ul style="list-style-type: none"> • statistics - These values provide standard statistical measures about the base values over the given summary window. Summaries of this type must contain time series with values of type <code>StatisticalSummaryType</code>
summary-window	int	The window (in seconds) over which the data is summarized. 0 if the base type.
time-updated	UNIX timestamp	Unix timestamp indicating when the summary last received new data. A value of null indicates that it has no data.

Example:

```
{
  uri: /perfsonar/archive/ABC123/histogram-owdelay/aggregations/3600,
  summary-type: "aggregation",
  summary-window: 3600,
  time-updated: 1397242422,
}
```

BaseTimeSeriesDatum<TYPE>

Parent Type(s): None

Description: The fundamental data object of items currently in the archive. It associates a particular time with a value. This will usually be collected with other `BaseTimeSeriesDatum` objects in an array.

Keys:

Name	Type	Description
ts	UNIX timestamp	The time associated with the value such as the time a measurement occurred.
val	<TYPE>	Any value of type <TYPE> where type is context specific. For example in this document a reference to <code>TimeNewSeries<String></code> means a <code>TimeSeries</code> objects where the value field is always of a <code>String</code> type.

Example:

```
{
```

```
ts: 1384712820,  
val: 600  
}
```

BaseTimeSeriesConstructor<TYPE>

Parent Type(s): None

Description: The fundamental object used to register new data to the archive.

Keys:

Name	Type	Description
event-type	string	The event-type of the data being written
val	<TYPE>	The value to store

Example:

```
{  
  event-type: "packet-count-sent",  
  val: 1000000000  
}
```

Metadata JSON Types

BaseIPPacketMetadata

Parent Type(s): None

Description: An object that describes a standard parameters for IP packets. Always inherited, never instantiated. None of the fields are required unless otherwise specified in the context-specific type.

Keys:

Name	Type	Description
ip-tos	int	The value of the ip type-of-service bit if set
ip-ttl	int	The value of the ip time-to-live field if set
ip-transport-protocol	string	The transport layer protocol such as 'tcp' or 'udp'.

ip-packet-size	integer	The size (bytes) of the packets sent
----------------	---------	--------------------------------------

Example:

```
{
  "uri": "/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B",
  "metadata-key": "DE1DADF052BA11E3AE3667D36188709B",
  "subject-type": "example-type",
  "tool-name": "example-tool",
  "ip-tos": 32,
  "event-types": [
    {
      event-type: "throughput",
      base-uri: "/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B/throughput",
      time-updated: 1397242422,
    }
  ]
}
```

BaseTimeMetadata

Parent Type(s): None		
Description: An object that describes something that runs at specified time intervals.		
Keys:		
Name	Type	Description
time-duration	int	Number of seconds that a test session runs.
time-interval	int	Number of seconds between test sessions. 0 if continuous. Null or undefined means no interval such as an on-demand test.
time-interval-randomization	int	Amount of randomization in interval. 0 if none.
time-probe-interval	int	mean average time between probes (seconds). Example: powstream's -i option.

Example:

```
{
```

```

"uri": "/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B",
"metadata-key": "DE1DADF052BA11E3AE3667D36188709B",
"subject-type": "example-type",
"tool-name": "example-tool",
"time-duration": 30,
"time-interval": 14400,
"time-randomization": 10,
"event-types": [
  {
    event-type: "throughput",
    base-uri: "/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B/throughput",
    time-updated: 1397242422,
  }
]
}

```

BasePacketSampleMetadata

Parent Type(s): BaseP2PMetadata, BaseIPPacketMetadata, BaseTimeMetadata

Description: An object that describes a tool that collects a large sample of packets.

Examples include powstream, owping and ping

Keys:

Name	Type	Description
sample-bucket-width	double	The bucket width of the provided data which indicates the level of accuracy to which it is rounded in seconds (e.g. .1 means rounded to nearest 1/10th of a second)
sample-size	integer	The size of the sample. For ping this could be the packet count.

Example:

```

{
  "uri": "/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B",
  "metadata-key": "DE1DADF052BA11E3AE3667D36188709B",
  "subject-type": "point-to-point",
  "tool-name": "owamp/powstream",
  "source": "198.124.238.49",
  "destination": "198.129.252.142",
  "measurement-agent": "198.124.238.49",
  "input-source": "bnl-owamp.es.net",
}

```

```

    "input-destination": "lbl-owamp.es.net",
    "time-duration": 60,
    "time-probe-interval": 0.1,
    "time-interval": 0,
    "time-interval-randomization": 0,
    "sample-bucket-width": 0.0001,
    "event-types": [
      {
        event-type: "histogram-owdelay",
        base-uri:
"/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B/histogram-owdelay/base",
        time-updated: 1397242422,
      }
    ]
  }
}

```

BaseTCPMetadata

Parent Type(s): None

Description: An object that describes a TCP connection

Keys:

Name	Type	Description
tcp-mss	int	The maximum segment size if explicitly set
tcp-no-delay	boolean	If true, indicates TCP no delay, disabling Nagle's Algorithm
tcp-window-size	int	The size in bytes of the TCP window

BaseThroughputMetadata

Parent Type(s): BaseP2PMetadata, BaseIPPacketMetadata, BaseTimeMetadata, BaseTCPMetadata

Description: An object that describes a throughput tool test specification. Examples include iperf, iperf3 and nuttcp

Keys:

Name	Type	Description
------	------	-------------

bw-buffer-size	int	The length of the network buffer in bytes
bw-parallel-streams	int	The number of parallel streams
bw-target-bandwidth	int	The target bandwidth for the transfer.
bw-zero-copy	boolean	Use a 'zero copy' method of sending data
bw-ignore-first-seconds	int	Ignores the first n seconds

Example:

```
{
  "uri": "/perfsonar/archive/D694500252F111E388E97F226288709B",
  "metadata-key": "D694500252F111E388E97F226288709B",
  "subject-type": "point-to-point",
  "tool-name": "bwctl/iperf3",
  "source": "198.124.238.38",
  "destination": "198.129.254.30",
  "measurement-agent": "198.124.238.38",
  "input-source": "bnl-pt1.es.net",
  "input-destination": "lbl-pt1.es.net",
  "time-interval": 21600,
  "time-interval-randomization": 10,
  "time-duration": 20,
  "ip-transport-protocol": "tcp",
  "ip-tos": 32,
  "event-types": [
    {
      event-type: "packet-count-sent",
      base-uri:
"/perfsonar/archive/D694500252F111E388E97F226288709B/packet-count-sent/base",
      time-updated: 1397242422,
    },
    {
      event-type: "throughput",
      base-uri:
"/perfsonar/archive/D694500252F111E388E97F226288709B/throughput/base",
      time-updated: 1397242422,
      summaries: []
    }
  ]
}
```

Parent Type(s): BaseP2PMetadata, BaseIPPacketMetadata, BaseTimeMetadata

Description: An object that describes a packet trace test specification. Examples include traceroute and tracepath.

Keys:

Name	Type	Description
trace-first-ttl	integer	The first hop reported in the results. Counting starts at 1.
trace-max-ttl	integer	Maximum number of hops this traceroute will probe before ending the test if the destination is not found
trace-num-queries	int	The number of queries to each TTL value

Example:

```
{
  "uri": "/perfsonar/archive/97CA69B8539811E3BCED777A6188709B",
  "metadata-key": "97CA69B8539811E3BCED777A6188709B",
  "subject-type": "point-to-point",
  "tool-name": "bwctl/iperf3",
  "source": "198.124.238.38",
  "destination": "198.129.254.30",
  "measurement-agent": "198.124.238.38",
  "input-source": "bnl-pt1.es.net",
  "input-destination": "lbl-pt1.es.net",
  "time-interval": 600,
  "time-interval-randomization": 0,
  "ip-transport-protocol": "icmp",
  "ip-tos": 32,
  "trace-num-queries": 3,
  "trace-first-ttl": 1,
  "trace-max-ttl": 255,
  "ip-packet-size": 60,
  "event-types": [
    {
      event-type: "packet-trace",
      base-uri:
"/perfsonar/archive/97CA69B8539811E3BCED777A6188709B/packet-trace",
      time-updated: 1397242422,
    }
  ]
}
```

Datum JSON Types

FailureDescriptor

Parent Type(s): None

Description: An object that describes the failure of a measurement

Keys:

Name	Type	Description
error	string	A human-readable description of the error

Example:

```
{
  "error": "Control connection timed-out to remote host"
}
```

HistogramItem

Parent Type(s): None

Description: An object that represents an entry in a histogram. This will usually be collected with other Histogram objects in an array.

Keys:

Name	Type	Description
<bucket-label>	integer	An entry in the histogram. The key is the bucket label which may be any string. Think of it as a category label. The value is always a number indicating the number of items that fall in that category.

Example:

```
{
  "1": 100,
  "2": 10,
  "3": 40
}
```

PercentageConstructor

Parent Type(s): None

Description: An object used to create a percentage object. This constructor allows data to be more easily aggregated over time by providing raw values used in the calculation

Keys:

Name	Type	Description
numerator	integer	The number to be used at the top of the fraction in the percentage calculation
denominator	integer	The number used at the bottom of the fraction in the percentage calculation

Example:

```
{
  numerator: 50,
  denominator: 100,
}
```

PacketTraceItem

Parent Type(s): None		
Description: An object that represents the result of a packet trace such as one performed by traceroute or tracepath. These will generally be collected in an array.		
Keys:		
Name	Type	Description
ttl	string	The TTL set in the probe as an integer. It can also be viewed as the hop number.
query	integer	Many tools send multiple probes to the same ttl. This is an integer indicating which attempt this probe was starting at 1.
success	boolean	Indicates whether the probe succeeded.
error-message	string	If <i>success</i> is 0, then provides human-readable description of error. Must exist but can be null if <i>success</i> is 1 or no error message can be generated.
ip	string	The IP address returned by the probe as string. Should exist but may be null if not measured.
rtt	double	The round-trip time of the probe as a real number in milliseconds. It should exist but be null if not measured by tool.

mtu	integer	The maximum transmission unit measured by the probe as an integer in bytes. Should be null but exist of not measured.
------------	---------	---

Example:

```
{
  "ttl": 1,
  "query": 1,
  "success": 1,
  "error-message": null,
  "ip": "198.124.238.37",
  "rtt": 0.191,
  "mtu": 9000
}
```

StatisticalSummary

Parent Type(s): None Description: An object that contains values of a basic statistical summary Keys:		
Name	Type	Description
minimum	double	The minimum value observed
maximum	double	The maximum value observed
median	double	The median value observed
mean	double	The average value observed
standard-deviation	double	The standard deviation observed
variance	double	The variance observed
percentile-25	double	The 25th percentile
percentile-75	double	The 75th percentile
percentile-95	double	The 95th percentile

Example:

```
{
  "minimum": 1.0,
  "maximum": 10.0,
  "median": 5.0,
  "mean": 5.5,
```



```
“standard-deviation”: 3.02765,  
}
```

SubintervalDatum<TYPE>

Parent Type(s): None

Description: The fundamental data object of subinterval

Keys:

Name	Type	Description
start	float	Start time relative to beginning of test.
duration	float	The duration of the interval
val	<TYPE>	Any value of type <TYPE> where type is context specific. For example in this document a reference to SubintervalDatum<String> means a TimeSeries objects where the value field is always of a String type.

Example:

```
{  
  start: 0,  
  duration: 1.0,  
  val: 125000000  
}
```

Event Types

Name	Type	Description
failures	FailureDescriptor	A record of test failures. A failure is any measurement that was scheduled to run but circumstances led to a state where it is unable to record a measurement. For example, being unable to connect to the remote endpoint.
histogram-owdelay	HistogramItem	A histogram describing the observed one-way delays over a time period. Buckets always in milliseconds.

histogram-rtt	HistogramItem	A histogram describing the observed packet round-trip times over a over a time period. Buckets always in milliseconds.
histogram-ttl	HistogramItem	A histogram describing the observed number of hops (time-to-live) of packets over a time period from source to destination.
histogram-ttl-reverse	HistogramItem	A histogram describing the observed number of hops (time-to-live) of packets over a time period from destination to source.
packet-duplicates	Integer	The number of duplicate packets observed in a sample for a single direction.
packet-duplicates-bidir	Integer	The number of duplicate packets observed for a complete packet round trip (from source to destination and then back to source)
packet-loss-rate	Percentage (see Appendix B)	The number of packets lost in one direction divided by the number of packets sent over a given summarization window.
packet-loss-rate-bidir	Percentage (see Appendix B)	The number of packets lost in both directions divided by the number of packets sent over a given summarization window.
packet-trace	PacketTraceItem	The observed packet trace such as that returned by traceroute or tracepath
packet-count-lost	Integer	The number of packets dropped in one direction. This is a raw count of packets and can be combined with <i>packet-count-sent</i> event type data to determine the rate of unidirectional loss.
packet-count-lost-bidir	Integer	The number of packets dropped in both directions. This is a raw count of packets and can be combined with <i>packet-count-sent</i> event type data to determine the rate of biidirectional loss.
packet-count-sent	Integer	The number of packets sent in a sample. This is a raw count of packets and can be combined with <i>packet-count-lost</i> event type data to determine the rate of loss.

packet-reorders	Integer	The number of packets received out of order for a unidirectional transfer
packet-reorders-bidir	Integer	The number of packets received out of order for a bidirectional transfer
packet-retransmits	Integer	The number of packets retransmitted for a transfer using reliable transport protocol such as TCP
path-mtu	Integer	The maximum transmission unit of a path
streams-packet-retransmits	Array of Integers	For tests running multiple streams, the packet-retransmits for each individual stream. Each stream is represented as a position in an array. If other stream-* stats collected, a stream must maintain the same position in each event type
streams-throughput	Array of Integers	For tests running multiple streams, the throughput for each individual stream. Each stream is represented as a position in an array. If other stream-* stats collected, a stream should maintain the same position in each event type
streams-throughput-subintervals	Array of SubintervalDatum<Integer> Arrays	For tests running multiple streams, the throughput subintervals for each individual stream. Each stream is represented as a position in an array. If other stream-* stats collected, a stream should maintain the same position in each event type
throughput	Integer	The observed amount of data sent over a period of time. Throughput must be in bits per second(bps).
throughput-subintervals	SubintervalDatum<Integer> Arrays	The throughput for individual subintervals of a throughput test
time-error-estimates	Double	An estimate of the clock error in a sample in milliseconds.

Messaging

URL Parameter Sets

Time Parameters

Parameter	Description
time	Match exactly the time given. If provided all other time URL parameters are ignored. In UNIX timestamp format.
time-start	Match only data that was measured after the given time (inclusive). If time-end nor time-range is defined, then it will return all results from the start time to the current time. In UNIX timestamp format.
time-end	Match only data that was measured before the given time (inclusive). If time-start nor time-range is provided, then will return all data stored in the archive up to and including the end-time. In UNIX timestamp format.
time-range	A time range in seconds. If time-start nor end-time is defined, then it is the number of seconds in the past from the current time. If only time-start is defined then it is the number of seconds after time-start to search. If only time-end is provided it is the number of seconds before end time to search. If both time-start and time-end are defined, this value is ignored.

Metadata Search Parameters

Name	Description
<metadata-parameter>	Any parameter defined in a metadata type is a valid search parameter with the exception of event-types (special parameters are defined below for searching event types). This includes the parameters defined in BaseMetadata and all inheriting data types

dns-match-rule	<p>If searching by hostname (See Appendix A: Searching by Hostname), indicates whether to use IPv6 DNS records (AAAA) and/or IPv4 records (A). The default value is “v4v6”. Valid values are below:</p> <ul style="list-style-type: none"> • prefer-v6: Prefers AAAA record if exists but will match against A record if no AAAA record found • prefer-v4: Prefers A record, but will match against AAAA records if one exists. • only-v6: Only matches against results of AAAA lookup. If no AAAA record, then no results will be returned. • only-v4: Only matches against A record. If no A record, then no results will be returned. • v4v6: Returns matches against both the A and AAAA record.
event-type	Matches metadata with the given event type.
summary-type	Matches metadata object containing summarizations of the given type
summary-window	Matches metadata objects containing summaries with the given summarization window.

Queries

General Queries

Listing/Searching Metadata

Description: List all the available metadata or search through existing metadata using parameters

URI: /perfsonar/archive

HTTP METHOD: GET

Request Type: application/json

URL Parameters: [Metadata Search Parameters](#)

Response Type: application/json

Return Object Type: Array of BaseMetadata objects

Return Object Example: *NOTE: event types replaced with “...” for sake of brevity*

```
[
{
  "uri": "/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B",
  "metadata-key": "DE1DADF052BA11E3AE3667D36188709B",
  "subject-type": "point-to-point",
  "tool-name": "owamp/powstream",
```

```

    "source": "198.124.238.49",
    "destination": "198.129.252.142",
    "measurement-agent": "198.124.238.49",
    "input-source": "bnl-owamp.es.net",
    "input-destination": "lbl-owamp.es.net",
    "time-duration": 60,
    "time-interval": 0,
    "time-interval-randomization": 0,
    "sample-bucket-width": 0.0001,
    "event-types": [...]
  },
  {
    "uri": "/perfsonar/archive/211684A452BD11E3A63AC4D56188709B",
    "metadata-key": "211684A452BD11E3A63AC4D56188709B",
    "subject-type": "point-to-point",
    "tool-name": "owamp/powstream",
    "source": "198.129.252.142",
    "destination": "198.124.238.49",
    "measurement-agent": "198.124.238.49",
    "input-source": "lbl-owamp.es.net",
    "input-destination": "bnl-owamp.es.net",
    "time-duration": 60,
    "time-interval": 0,
    "time-interval-randomization": 0,
    "sample-bucket-width": 0.0001,
    "event-types": [...]
  }
]

```

Querying a Single Metadata Object

Description: Returns the details of a specific metadata key

URI: /perfsonar/archive/<metadata-key>

HTTP METHOD: GET

Request Type: application/json

URL Parameters: None

Response Type: application/json

Return Object Type: BaseMetadata

Return Object Example:

```

{
  "uri": "/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B",
  "metadata-key": "DE1DADF052BA11E3AE3667D36188709B",
  "subject-type": "point-to-point",
  "tool-name": "owamp/powstream",
  "source": "198.124.238.49",

```

```

"destination": "198.129.252.142",
"measurement-agent": "198.124.238.49",
"input-source": "bnl-owamp.es.net",
"input-destination": "lbl-owamp.es.net",
"time-duration": 60,
"time-probe-interval": 0.1,
"time-interval": 0,
"interval-randomization": 0,
"sample-bucket-width": 0.0001,
"event-types": [
  {
    event-type: "packet-count-lost",
    base-uri:
"/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B/packet-count-lost/base",
  },
  {
    event-type: "packet-count-sent",
    base-uri:
"/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B/packet-count-sent/base",
  },
  {
    event-type: "packet-duplicates",
    base-uri:
"/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B/packet-duplicates/base",
  },
  {
    event-type: "time-error-estimates",
    base-uri:
"/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B/time-error-estimates/base",
  },
  {
    event-type: "histogram-owdelay",
    base-uri:
"/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B/histogram-owdelay/base",
    summaries:[
      {
        uri:
"/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B/histogram-owdelay/statistics/
0",
        summary-type: "statistics"
        summary-window: 60
      }
    ]
  },
  {
    event-type: "histogram-ttl",

```

```

      base-uri:
"/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B/histogram-ttl/base",
    },
    {
      event-type: "summary-owdelay",
      base-uri:
"/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B/summary-owdelay/base"
    },
    {
      event-type: "packet-loss-rate",
      base-uri:
"/perfsonar/archive/DE1DADF052BA11E3AE3667D36188709B/packet-loss-rate/base",
    }
  ]
}

```

Querying Event Type Descriptors

Description: Get the event type descriptor including available summaries

URI: /perfsonar/archive/<metadata-key>/<event-type>

HTTP METHOD: GET

Request Type: application/json

URL Parameters: None

Response Type: application/json

Return Object Type: Array of BaseEventTypeDescriptor objects

Return Object Example:

```

[
  {
    event-type: "histogram-owdelay",
    base-uri: /perfsonar/archive/ABC123/histogram-owdelay/base,
    summaries: [
      {
        uri: /perfsonar/archive/ABC123/histogram-owdelay/aggregations/3600,
        summary-type: "aggregation",
        summary-window: 3600,
      }
    ]
  }
]

```

Query Base Data

Description: Get a list of data with a summary-type of "base"

URI: /perfsonar/archive/<metadata-key>/<event-type>/base

HTTP METHOD: GET

Request Type: application/json

URL Parameters: [Time Parameters](#)

Response Type: application/json

Return Object Type: Array of BaseTimeSeriesDatum<Integer> objects

Return Object Example:

```
[
  {
    ts: 1384712820,
    val: {
      "error": "Unable to connect to remote host."
    }
  },
  {
    ts: 1384712880,
    val: {
      "error": "Unable to find timeslot to schedule test"
    }
  }
]
```

Querying Summarization Descriptors

Description: List the available summaries for a given non-base summary type. Valid summary types are plural forms of the types: aggregations statistics, and subintervals

URI: /perfsonar/archive/<metadata-key>/<event-type>/<summary-type>

HTTP METHOD: GET

Request Type: application/json

URL Parameters:

summary-window	Matches metadata objects containing summaries with the given summarization window.
----------------	--

Response Type: application/json

Return Object Type: Array of BaseEventTypeDescriptor objects

Return Object Example:

```
[
  {
    uri: /perfsonar/archive/ABC123/histogram-owdelay/aggregations/3600,
    summary-type: "aggregation",
    summary-window: 3600,
  },
  {
    uri: /perfsonar/archive/ABC123/histogram-owdelay/aggregations/86400,
```

```
summary-type: "aggregation",
summary-window: 86400
}
]
```

Querying Summary Data

Description: Get summarized data
URI: /perfsonar/archive/<metadata-key>/<event-type>/<summary-type>/<summary-window>
HTTP METHOD: GET
Request Type: application/json
URL Parameters: [Time Parameters](#)
Response Type: application/json
Return Object Type: Array of BaseTimeSeriesDatum<Integer> objects
Return Object Example:

```
[
  {
    ts: 1384712820,
    val: {
      "10": 300,
      "11": 300,
    }
  },
  {
    ts: 1384712880,
    val: {
      "10": 299,
      "11": "301"
    }
  }
]
```

Query Examples

Querying Failures

Description: Get a list of measurement failures
URI: /perfsonar/archive/<metadata-key>/failures/base
HTTP METHOD: GET
Request Type: application/json
URL Parameters: [Time Parameters](#)
Response Type: application/json
Return Object Type: Array of BaseTimeSeriesDatum<Integer> objects

Return Object Example:

```
[
  {
    ts: 1384712820,
    val: {
      "error": "Unable to connect to remote host."
    }
  },
  {
    ts: 1384712880,
    val: {
      "error": "Unable to find timeslot to schedule test"
    }
  }
]
```

Querying a One-way Delay Histogram

Description: Get one-way delay histograms for the given summary window

URI:

/perfsonar/archive/<metadata-key>/histogram-owdelay/base

/perfsonar/archive/<metadata-key>/histogram-owdelay/aggregations/<summary-window>

HTTP METHOD: GET

Request Type: application/json

URL Parameters: [Time Parameters](#)

Response Type: application/json

Return Object Type: Array of BaseTimeSeriesDatum<Array<HistogramItem>> objects

Return Object Example:

```
[
  {
    ts: 1384712820,
    val: {
      "10": 300,
      "11": 300,
    },
  },
  {
    ts: 1384712880,
    val: {
      "10": 299,
      "11": "301"
    }
  }
]
```

Querying a Round Trip Time (RTT) Histogram

Description: Get rtt histograms for the given summary window

URI:

/perfsonar/archive/<metadata-key>/histogram-rtt/base

/perfsonar/archive/<metadata-key>/histogram-rtt/aggregations/<summary-window>

HTTP METHOD: GET

Request Type: application/json

URL Parameters: [Time Parameters](#)

Response Type: application/json

Return Object Type: Array of BaseTimeSeriesDatum<Array<HistogramItem>> objects

Return Object Example:

```
[
  {
    ts: 1384712820,
    val: {
      "10": 300,
      "11": 300,
    }
  },
  {
    ts: 1384712880,
    val: {
      "10": 299,
      "11": "301"
    }
  }
]
```

Querying a TTL Histogram

Description: Get TTL histograms for the given summary window

URI:

/perfsonar/archive/<metadata-key>/histogram-ttl/base

/perfsonar/archive/<metadata-key>/histogram-ttl/aggregations/<summary-window>

HTTP METHOD: GET

Request Type: application/json

URL Parameters: [Time Parameters](#)

Response Type: application/json

Return Object Type: Array of BaseTimeSeriesDatum<Array<HistogramItem>> objects

Return Object Example:

```
[
  {
```

```
    ts: 1384712820,
    val: {
      "10": 300,
      "11": 300,
    }
  },
  {
    ts: 1384712880,
    val: {
      "10": 299,
      "11": "301"
    }
  }
]
```

Querying Packet Duplicates

Description: Get a count of the packets duplicated in each measurement sample.

URI: /perfsonar/archive/<metadata-key>/packet-duplicates/base

HTTP METHOD: GET

Request Type: application/json

URL Parameters: [Time Parameters](#)

Response Type: application/json

Return Object Type: Array of BaseTimeSeriesDatum<Integer> objects

Return Object Example:

```
[
  {
    ts: 1384712820,
    val: 0
  },
  {
    ts: 1384712880,
    val: 2
  }
]
```

Querying a Packet Loss Rate Summary

Description: The percentage of packet loss in the given summary window over time.

URI:

/perfsonar/archive/<metadata-key>/packet-loss-rate/base

HTTP METHOD: GET

Request Type: application/json

URL Parameters: [Time Parameters](#)

Response Type: application/json

Return Object Type: Array of BaseTimeSeriesDatum<Double> objects

Return Object Example:

```
[
  {
    ts: 1384712820,
    val: 0.01
  },
  {
    ts: 1384712880,
    val: 0
  }
]
```

Querying a Packet Trace

Description: Get a list of measurement failures

URI: /personar/archive/<metadata-key>/packet-trace/base

HTTP METHOD: GET

Request Type: application/json

URL Parameters: [Time Parameters](#)

Response Type: application/json

Return Object Type: Array of BaseTimeSeriesDatum<PacketTraceItem> objects

Return Object Example:

```
[
  {
    ts: 1384712820,
    val: [
      {
        "ttl": 1,
        "query": 1,
        "success": 1,
        "error-message": null,
        "ip": "198.124.238.37",
        "rtt": 0.191,
        "mtu": 9000
      },
      {
        "ttl": 1,
        "query": 2,
        "success": 1,
        "error-message": null,
        "ip": "198.124.238.37",
        "rtt": 0.165,
        "mtu": 9000
      }
    ]
  }
]
```

```
{
  "ttl": 2,
  "query": 1,
  "success": 1,
  "error-message": null,
  "ip": "134.55.221.134",
  "rtt": 2.427,
  "mtu": 9000
},
{
  "ttl": 2,
  "query": 2,
  "success": 1,
  "error-message": null,
  "ip": "134.55.221.134",
  "rtt": 3.545,
  "mtu": 9000
},
{
  "ttl": 3,
  "query": 1,
  "success": 1,
  "error-message": null,
  "ip": "134.55.49.30",
  "rtt": 28.462,
  "mtu": 9000
},
{
  "ttl": 3,
  "query": 2,
  "success": 1,
  "error-message": null,
  "ip": "134.55.49.30",
  "rtt": 28.880,
  "mtu": 9000
},
{
  "ttl": 4,
  "query": 1,
  "success": 1,
  "error-message": null,
  "ip": "198.129.254.30",
  "rtt": 74.950,
  "mtu": 9000
},
{
  "ttl": 4,
```

```
    "query": 2,  
    "success": 1,  
    "error-message": null,  
    "ip": "198.129.254.30",  
    "rtt": 74.976,  
    "mtu": 9000  
  }  
]  
}  
]
```

Querying Packets Lost

Description: Get a count of the packets lost in each measurement sample. Note that this is not the loss percentage but a raw count of the packets dropped.

URI: /perfsonar/archive/<metadata-key>/packet-count-lost/base

HTTP METHOD: GET

Request Type: application/json

URL Parameters: [Time Parameters](#)

Response Type: application/json

Return Object Type: Array of BaseTimeSeriesDatum<Integer> objects

Return Object Example:

```
[  
  {  
    ts: 1384712820,  
    val: 0  
  },  
  {  
    ts: 1384712880,  
    val: 10  
  }  
]
```

Querying Packets Sent

Description: Get a count of the packets sent in each measurement sample.

URI: /perfsonar/archive/<metadata-key>/packet-count-sent/base

HTTP METHOD: GET

Request Type: application/json

URL Parameters: [Time Parameters](#)

Response Type: application/json

Return Object Type: Array of BaseTimeSeriesDatum<Integer> objects

Return Object Example:


```
[
  {
    ts: 1384712820,
    val: 600
  },
  {
    ts: 1384712880,
    val: 600
  }
]
```

Querying a One-Way Delay Statistical Summary

Description: Get summary statistics on one-way delay over the summary-window given. All values are in milliseconds.

URI:

/perfsonar/archive/<metadata-key>/histogram-owdelay/statistics/<summary-window>

HTTP METHOD: GET

Request Type: application/json

URL Parameters: [Time Parameters](#)

Response Type: application/json

Return Object Type: Array of BaseTimeSeriesDatum<StatisticalSummary> objects

Return Object Example:

```
[
  {
    ts: 1384712820,
    val: {
      "minimum": 1.0,
      "maximum": 10.0,
      "median": 5.0,
      "mean": 5.5,
      "standard-deviation": 3.02765,
    }
  },
  {
    ts: 1384712880,
    val: {
      "minimum": 10.0,
      "maximum": 10.0,
      "median": 10.0,
      "mean": 10,
      "standard-deviation": 0.0,
    }
  }
]
```

Querying a Round Trip Time (RTT) Summary

Description: Get summary statistics on rtt over the summary-window given. All values are in milliseconds.

URI:

/perfsonar/archive/<metadata-key>/summary-rtt/base

/perfsonar/archive/<metadata-key>/summary-rtt/aggregations/<summary-window>

HTTP METHOD: GET

Request Type: application/json

URL Parameters: [Time Parameters](#)

Response Type: application/json

Return Object Type: Array of BaseTimeSeriesDatum<StatisticalSummary> objects

Return Object Example:

```
[
  {
    ts: 1384712820,
    val: {
      "minimum": 1.0,
      "maximum": 10.0,
      "median": 5.0,
      "mean": 5.5,
      "standard-deviation": 3.02765,
    }
  },
  {
    ts: 1384712880,
    val: {
      "minimum": 10.0,
      "maximum": 10.0,
      "median": 10.0,
      "mean": 10,
      "standard-deviation": 0.0,
    }
  }
]
```

Querying Throughput

Description: Get the throughput measurements recorded in the archive for a given metadata key. All results are in bits per second (bps).

URI: /perfsonar/archive/<metadata-key>/throughput/base

HTTP METHOD: GET

Request Type: application/json

URL Parameters: [Time Parameters](#)

Response Type: application/json

Return Object Type: Array of BaseTimeSeriesDatum<Integer> objects

Return Object Example:

```
[
  {
    ts: 1384712820,
    val: 1000000000
  },
  {
    ts: 1384712880,
    val: 9000000000
  }
]
```

Querying Throughput Subintervals

Description: Get the subintervals reported for each throughput measurement

URI: /perfsonar/archive/<metadata-key>/throughput-subintervals/base

HTTP METHOD: GET

Request Type: application/json

URL Parameters: [Time Parameters](#)

Response Type: application/json

Return Object Type: Array of SubintervalDatum<Integer> objects

Return Object Example:

```
[
  {
    ts: 1384712820,
    val: [
      {
        start: 0,
        duration: 1.0,
        val: 8500000000
      },
      {
        start: 1.0,
        duration: 1.0,
        val: 9500000000
      }
    ]
  },
  {
    ts: 1384712880,
```

```

    val: [
      {
        start: 0,
        duration: 1.2,
        val: 850000000
      },
      {
        start: 1.2,
        duration: 1.1,
        val: 950000000
      }
    ]
  }
]

```

Querying Time Error Estimates

Description: Get the time error estimate of each measurement sample.

URI: /personar/archive/<metadata-key>/time-error-estimates/base

HTTP METHOD: GET

Request Type: application/json

URL Parameters: [Time Parameters](#)

Response Type: application/json

Return Object Type: Array of BaseTimeSeriesDatum<Double> objects

Return Object Example:

```

[
  {
    ts: 1384712820,
    val: 0.012345
  },
  {
    ts: 1384712880,
    val: 0.09876
  }
]

```

Creates and Updates

Registering New Metadata

Description: Create a new metadata object and return a URI for the newly created object. As a convenience, if all the parameters of the submitted object including event-types exactly match an existing metadata object (and the creator has permissions to update the existing metadata) then that existing object will be returned. This will make life easier for registering

clients that don't save state, are restarted, or are updated to store new data from an existing tool.

URI: /perfsonar/archive

HTTP METHOD: POST

Request Type: application/json

Request Object Type: BaseMetadata (with uri and metadata-key undefined)

Request Object Example:

```
{
  "tool-name": "bwctl/iperf3",
  "source": "198.124.238.38",
  "destination": "198.129.254.30",
  "measurement-agent": "198.124.238.38",
  "input-source": "bnl-pt1.es.net",
  "input-destination": "lbl-pt1.es.net",
  "time-interval": 21600,
  "time-interval-randomization": 10,
  "time-duration": 20,
  "ip-transport-protocol": "tcp",
  "ip-tos": 32,
  "event-types": [
    {
      event-type: "packet-count-sent",
    },
    {
      event-type: "throughput",
    }
  ]
}
```

Response Type: application/json

Return Object Type: BaseMetadata (with uri and metadata-key defined)

Return Object Example:

```
{
  "uri": "/perfsonar/archive/D694500252F111E388E97F226288709B",
  "metadata-key": "D694500252F111E388E97F226288709B",
  "tool-name": "bwctl/iperf3",
  "source": "198.124.238.38",
  "destination": "198.129.254.30",
  "measurement-agent": "198.124.238.38",
  "input-source": "bnl-pt1.es.net",
  "input-destination": "lbl-pt1.es.net",
  "time-interval": 21600,
  "time-interval-randomization": 10,
  "time-duration": 20,
  "ip-transport-protocol": "tcp",
  "ip-tos": 32,
```

```

"event-types": [
  {
    event-type: "packet-count-sent",
    base-uri:
"/perfsonar/archive/D694500252F111E388E97F226288709B/packet-count-sent",
    time-updated: null,
  },
  {
    event-type: "throughput",
    base-uri: "/perfsonar/archive/D694500252F111E388E97F226288709B/throughput",
    time-update: null,
    summaries: [
      {
        uri: "/perfsonar/archive/D694500252F111E388E97F226288709B/throughput",
        summary-type: "subintervals"
        summary-window: 1,
        time-updated: null,
      }
    ]
  }
]
}

```

Writing to Multiple Event Types

Description: Write one or more data elements to a metadata object. If one write fails, then the entire request fails.

URI: /perfsonar/archive/<metadata-key>

HTTP METHOD: POST

Request Type: application/json

Request Object Type: Object containing 'data' field with array of TimeSeriesDatum<BaseTimeSeriesConstructor> objects

Request Object Example:

```

{
  data: [
    {
      ts: 1384712820,
      val: [
        {
          event-type: "throughput",
          val: 1000000000,
        },
      ],
    }
  ],
}

```

```

    {
      event-type: "packet-count-sent",
      val: 2500,
    },
  ]
}

```

Writing to a Single Event Type

Description: Write a single data elements to a metadata object

URI:

/perfsonar/archive/<metadata-key>/base

/perfsonar/archive/<metadata-key>/<summary-type>/<summary-window>

HTTP METHOD: POST

Request Type: application/json

Request Object Type: TimeSeriesDatum object

Request Object Example:

```

{
  ts: 1384712820,
  val: 1000000000
}

```

Response Type: application/json

Return Object Type: None

Security Considerations

The interface to the Measurement Archive should provide a basic level of security. The protocol relies on out-of-band mechanisms to handle security. In general the measurement archive should provide at a minimum the level of security defined in this document.

Read Access

The measurement archive is designed to be open and allow the sharing of data. It is not a requirement that any read access control be implemented to limit what users can pull from the archive. On the other hand, the protocol was not designed to prohibit some level of read access control so it can be considered optional.

Write Access

A measurement archive SHOULD implement some type of write access control. It should be able to limit who can create/update metadata and data. All POST and PUT methods should be sent over an HTTPS connection. While no authentication mechanism is currently preferred,

something such as IP-based filtering, HTTP BASIC, client SSL, or OpenID should be used to identify the requester. Once authenticated some basic access control should be implemented. At a bare minimum they should only allow the creator of a test specification to update and post data to that specification. More complex schemes where groups of users share write access to the same resources are possible, but an implementation should put in some basic access control so that those with the ability to update data is limited.

Appendix A: Searching by Hostname

As defined certain fields MUST contain an IPv4 or IPv6 address when returned as the result of a metadata search or query. When searching though, it may be desirable to search by a DNS hostname. As a result some rules must be established so that consistent results are returned.

A few basic rules to enable searching by hostname are listed below:

1. In the object definition found in this document, all fields that store addresses but can be searched by hostname are labeled with the special *ip-string* type.
2. An archive storing these fields MUST return them as IPv4 or IPv6 addresses whenever the object is returned as part of a search or direct query
3. The parameters provided by the user may be any valid IPv4, IPv6, DNS CNAME or primary DNS name.
4. If given a hostname it is the duty of the archive to perform the required lookup and map it to the correct address. This should be done according to the *dns-match-rule* parameter if provided. If not provided it will prefer the AAAA record but fallback to the A record if no AAAA record can be found. If a CNAME is provided, it should follow the CNAME until a record is found with the requested AAAA and/or A record.
5. The archive is not concerned with tracking changes in hostnames over time. A hostname can simply be thought of as a current alias to the desired address.

These rules should allow for convenient searching while also avoiding common pitfalls associated with trying to account for address changes over time.

Appendix B: Percentage Type

There is a special type called “percentage” that can be assigned to event-types. It is special in the sense that the object used to create it is not the same as the object returned by a query. A “percentage” is created with a [PercentageConstructor](#) object which specifies the numerator and denominator in the percentage calculation. The returned type is a double with the result of the division. this separation is done for summarization purposes. if you want to summarize a percentage value over a longer time window, it will be considerably more accurate to use the actual values used in the calculation when calculating the longer window.