

Technical Document

# Niagara Analytics Framework Web API Guide

August 24, 2018



# Niagara Analytics Framework Web API Guide

## **Tridium, Inc.**

3951 Westerre Parkway, Suite 350  
Richmond, Virginia 23233  
U.S.A.

## **Confidentiality**

The information contained in this document is confidential information of Tridium, Inc., a Delaware corporation ("Tridium"). Such information and the software described herein, is furnished under a license agreement and may be used only in accordance with that agreement.

The information contained in this document is provided solely for use by Tridium employees, licensees, and system owners; and, except as permitted under the below copyright notice, is not to be released to, or reproduced for, anyone else.

While every effort has been made to assure the accuracy of this document, Tridium is not responsible for damages of any kind, including without limitation consequential damages, arising from the application of the information contained herein. Information and specifications published here are current as of the date of this publication and are subject to change without notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia.

## **Trademark notice**

BACnet and ASHRAE are registered trademarks of American Society of Heating, Refrigerating and Air-Conditioning Engineers. Microsoft, Excel, Internet Explorer, Windows, Windows Vista, Windows Server, and SQL Server are registered trademarks of Microsoft Corporation. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Mozilla and Firefox are trademarks of the Mozilla Foundation. Echelon, LON, LonMark, LonTalk, and LonWorks are registered trademarks of Echelon Corporation. Tridium, JACE, Niagara Framework, NiagaraAX Framework, and Sedona Framework are registered trademarks, and Workbench, WorkPlaceAX, and AXSupervisor, are trademarks of Tridium Inc. All other product names and services mentioned in this publication that are known to be trademarks, registered trademarks, or service marks are the property of their respective owners.

## **Copyright and patent notice**

This document may be copied by parties who are authorized to distribute Tridium products in connection with distribution of those products, subject to the contracts that authorize such distribution. It may not otherwise, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Tridium, Inc.

Copyright © 2018 Tridium, Inc. All rights reserved.

The product(s) described herein may be covered by one or more U.S. or foreign patents of Tridium.

# Contents

<b>About this guide .....</b>	<b>5</b>
Document change log .....	5
Related documentation .....	5
<b>Chapter 1 Web API protocol.....</b>	<b>7</b>
Web API properties .....	7
Analytics value binding .....	8
HTTP .....	8
JSON requests.....	8
Data types .....	9
Errors .....	9
JSON error messaging .....	10
Encoding infinity and NaN .....	10
Time ranges .....	11
Rollups and aggregation.....	14
Intervals.....	15
Setting up user authentication .....	16
<b>Chapter 2 Messages .....</b>	<b>19</b>
GetNode .....	19
GetChildNodes .....	21
GetValue.....	25
GetTrend .....	28
GetRollup .....	33
Invoke.....	36
Query .....	37
Subscribe.....	40
PollSubscription .....	45
Unsubscribe .....	48
<b>Index.....</b>	<b>51</b>



## About this guide

This guide provides reference documentation for the Niagara Analytics Framework Web API protocol.

### Audience

The information in this document is for product designers and programmers who are responsible for configuring complex building systems.

### Document content

Chapters document the Analytics value binding, HTTP, JSON requests, data types, errors, time ranges, roll-up and aggregation functions, intervals, and messages.

### Product documentation

This document is part of the Niagara technical documentation library. Released versions of Niagara software include a complete collection of technical information that is provided in both online help and PDF formats.

## Document change log

This topic summarizes the history of this document.

### August 24, 2018 version 2.1 release

- Added a sentence to the [Chapter 1 Web API protocol, page 7](#) introduction about the Web API component in the `analytics` palette.
- Added [Web API properties, page 7](#) topic.

### March 13, 2017 (version 2.0U2)

Updated interval topic and added user authentication topic.

### January 21, 2016 (initial release)

The material in this guide was originally contained in the *Component Guide* (for Niagara Analytics Framework version 1.0), which was created October 27, 2014.

## Related documentation

These documents provide additional information about These documents provide additional information about Niagara Analytics Framework.

- *NiagaraAX Analytics Framework Guide* explains concepts and provides procedures..
- *NiagaraAX Analytics Framework Reference* documents each component and plugin.



# Chapter 1 Web API protocol

## Topics covered in this chapter

- ◆ Web API properties
- ◆ Analytics value binding
- ◆ HTTP
- ◆ JSON requests
- ◆ Data types
- ◆ Errors
- ◆ Time ranges
- ◆ Rollups and aggregation
- ◆ Intervals
- ◆ Setting up user authentication

The Web API protocol expands Niagara Analytics Framework's potential by providing programmatic access to the collected data. Using APIs, a third-party company can create custom dashboards and export data processed by the Niagara Analytics Framework to an external storage location locally or in the cloud.

It also allows third party companies to create custom dashboards or pull data processed by the framework engine to an external storage present locally or in the cloud.

The Niagara Analytics Web API is an HTTP-based protocol that uses JSON messaging to access data including Niagara Analytics Framework data definitions. The **analytics** palette contains the Web API component, which you can drag to the **AnalyticService** in the Nav tree. This component makes all the API protocols easily available.

## Web API properties

These properties support the Web API component.

Figure 1 Web API properties

The screenshot shows a 'Property Sheet' for a 'Web API (Na Servlet)' component. It lists several properties with their current values and controls:

- Status**: {ok}
- Fault Cause**: (empty text field)
- Enabled**: true (with a green circle icon and a dropdown arrow)
- Servlet Name**: na
- Debug User**: (empty text field)
- Debug Requests**: true (with a green circle icon and a dropdown arrow)

In addition to the standard properties (Status, Fault Cause and Enabled), these properties support the Web API component.

Property	Value	Description
Servlet Name	defaults to na	Identifies the name of the servlet.
Debug User	text	Specifies the user debugging the requests. The system starts debugging the requests if the logged-in user is the same as this user.
Debug Requests	true (default) or false	Enables and disables the debugging of the request.

## Analytics value binding

You can add a value binding to a bound label and use it to display a real-time value associated with a framework variable. When used with a history variable (no real-time value, but with history data), the system displays the latest record's value next to the value binding.

This API allows external applications to get the current value of a Niagara Analytics Framework point.

## HTTP

You make requests using HTTP GET and POST.

- Use HTTP GET to debug issues using a browser address bar.
- Use HTTP POST for application runtime.

The servlet name is `na`. URLs will follow the form:

`http://www.example.com/na`

When using HTTP GET, you encode JSON into the URL using the query parameter `json`. For example

`http://www.example.com/na?json={}`

When using HTTP POST, the body of the HTTP message is the JSON message. The `ContentType` for the JSON messages should be `text/plain` so that when you are debugging, the browser will not try to use an external process for viewing the json content.

## JSON requests

Wrap all requests and responses in an envelope object. All envelopes can contain a list of requests or responses. A client submits a list of requests.

### Request example

```
{
  "minify" : "false",
  "requests" :
    [
      {},
      {}
    ]
}
```

### Request elements

Request element	Value	Description
minify (optional)	true or false (default)	Defines whether or not the response should be minified.
requests	list of messages to process	The server responds with a list of responses.

### Response example

```
{
  "responses" :
    [
      {},
      {}
    ]
}
```



}

## Data types

Data types classify data based on how the system uses each type of information.

The following are the possible values for type elements.

Type	Description
Boolean	True or False
Duration	A number representing milliseconds
Enum	Represented by the enum tag.
Numeric	May or may not have a decimal.
Status	<p>The highest priority status as a string. The possible values are:</p> <ul style="list-style-type: none"> <li>• Disabled: the associated value is considered to be invalid and should not be used.</li> <li>• Fault: the associated value is considered to be invalid and should not be used.</li> <li>• Down: the associated value is considered to be invalid and should not be used.</li> <li>• Stale: the associated value is considered to be invalid and should not be used.</li> <li>• Overridden</li> <li>• Null: the associated value is considered to be invalid and should not be used.</li> <li>• unackedAlarm</li> <li>• ok</li> </ul>
string	
time	This string represents the ISO-8601-encoded timestamp in the format: <code>yyyymmddThh:mm:ss.mmm[+/-]hh:mm</code> . You can replace the time zone offset with Z for UTC.
<typespec>	Defines the typespec, such as <code>control:NumericOverride</code> . Excluding action parameters and results, this definition is present only for Niagara simples. The value represents the string encoding.

## Errors

Always check the HTTP response code. If the response code is not 202 (HTTP OK), there is no JSON to parse.

The API has several pre-defined error ids.

Error code	Description
clientError	Reports something wrong with the client request.
invalidOrd	Reports an invalid address.

Error code	Description
permissionError	Reports permissions-related issues.
serverError	Reports an unexpected error on the server.
unknownSubscription	Reports invalid subscription names.

## JSON error messaging

Errors happen at different levels. Therefore, a client should check for an error object in the response envelope, in each message response, and in each sub-response of a subscription.

### Response envelope error

```
{
  "error" :
  "requests" :
    {
      "id" : "Error identifier",
      "display" : "Text description",
      "stackTrace" : "Text representing the server exception",
    }
}
```

### Message error

```
{
  "responses" :
  [
    {
      "message" : "GetValue",
      "node" : "slot/Drivers/Blah/Blah",
      "data" : "hs:power",
      "Error" :
      {
        "id" : "invalidOrd",
        "display" : "Cannot resolve slot:/Drivers/Blah/Blah"
        "stackTrace" : "..."
      }
    }
  ]
}
```

## Encoding infinity and NaN

Since JSON cannot encode “Not A Number” and infinity, these numeric values are encoded as strings. The string starts with the escape character, which is the JSON string: `\u001B`.

- NaN = `\u001BNaN`
- Positive infinity = `\u001BInfinity`
- Negative infinity = `\u001BInfinity`

## Time ranges

The Niagara Analytics Framework supports three types of time ranges; constants, absolute ranges and scripts.

### Constants

These pre-defined values can be used to quickly process requests that involve standard time ranges:

- all
- today
- yesterday
- thisWeek
- lastWeek
- thisMonth
- lastMonth
- thisYear
- lastYear
- weekToDate
- monthToDate
- yearToDate

### Absolute ranges

An absolute time range consists of two ISO 8601 ranges separated by a semi-column. The individual format of each is:

`yyyy-mm-ddThh:mm:ss.mmm[+/-]hh:mm`

where

Absolute element	Description
<code>yyyy-mm-dd</code>	Defines the date on which to begin the time range.
<code>T</code>	Identifies what follows as defining hours, minutes and seconds.
<code>hh:mm:ss</code>	Defines hours minutes and seconds.
<code>mmm</code>	
<code>[+/-]hh:mm</code>	Specifies an offset number of hours and minutes to add to or subtract from the previously defined date and time.

The offset can be replaced with Z for UTC (Coordinated Universal Time, which is the current local time). The above is the format that BAbsTime uses for encoding/decoding.

For example:

`2011-01-01T00:00:00-05:00;2012-12-31T23:59:59-05:00`

starts the time range on December 31, 2010 at 7 pm. and ends it on December 31, 2011 at 6:59 pm.

`2011-01-01T00:00:00Z;2012-12-31T23:59:59Z`

starts the time range on January 1, 2011 offset into the future by the UTC; and ends the time range on December 31, 2012 at 11:59 pm offset into the future by the UTC.

## Scripts

As with an absolute time range, a time range script allows you to apply any conceivable time range to a request, for example, a day's worth of data for Sunday of last week, or data from the first 13 minutes of the last hour, etc. The advantage of a script over an absolute time range is that a script is relative to the current time. With a script, when you visit the same chart or table next month, the system automatically updates the time range to show the latest information. If you specify an absolute time range, you have to manually modify the dates each time you access a chart in order to view the latest information.

A script adheres to the following syntax:

```
from<unit><unitOffset>To<duration>
```

where

Script element	Option	Description
from	required syntax, is case sensitive	This keyword indicates that what immediately follows configures the start of the time range.
<unit>	Seconds, Minutes, Hours, Days, Weeks, Months, Years (case sensitive)	Sets the alignment of the data origin to a current period. For example, <code>fromYear</code> sets the alignment to the start of the current year; <code>fromMinute</code> sets the alignment to the start of the current minute.
<unitOffset>	<- or +P or P><#Y><#M><#D>T<#H><#M><#S>	Specifies an offset amount to subtract from (-) or add to (+) the <unit> to establish the start of the time range. For details, see <a href="#">unitOffset, page 12</a> .
To	required syntax; is case sensitive.	This keyword separates the two parts of the time range and indicates that what follows configures the end of the time range.
<duration>	<Now> or <durationOffset>	Establishes the end date and time for the time range based on the start date and time. For details, see <a href="#">durationOffset, page 13</a> .

## unitOffset

Specifies an offset amount to apply to the <unit> to establish the start of the time range. The syntax of a unit offset is as follows:

```
<sign>P<#Y><#M><#D>T<#H><#M><#S>
```

unitOffset element	Value	Description
<sign>	plus (+) or minus (-)	Minus (-) subtracts from <unit>, which moves the beginning of the time range into the past.  Plus (+) adds to <unit>, which moves the beginning of the time range into the future.  P and +P are equivalent.
P	required case-sensitive letter	Begins the definition of the offset to apply to <unit> in terms of years, months and days.
<#Y><#M><#D>	optional year, month and day definitions	Defines a number of year(s), month(s), day(s) (case sensitive) to subtract from or add to <unit>.

unitOffset element	Value	Description
T	case-sensitive letter (required if you are using <code>&lt;#H&gt;&lt;#M&gt;&lt;#S&gt;</code> and optional if you are using only <code>&lt;#Y&gt;&lt;#M&gt;&lt;#D&gt;</code> )	Begins the definition of the offset in terms of hours, minutes and seconds.
<code>&lt;#H&gt;&lt;#M&gt;&lt;#S&gt;</code>	optional hours, minutes and seconds definitions	Defines an amount of time in hours, minutes and/or seconds (case sensitive) to subtract from or add to the <code>&lt;unit&gt;</code> .

### durationOffset

Defines the duration of the time range in relationship to the start of the range. The syntax of a unit offset is:

`<Now>`

or

`<P><#Y><#M><#D>T<#H><#M><#S>`

The system calculates the duration by adding these values to the start of the time range. `<durationOffset>` cannot be negative.

durationOffset element	Value	Description
<code>&lt;Now&gt;</code>	case-sensitive keyword	Defines the current date and time as the <code>&lt;duration&gt;</code> (end of the time range). If you use <code>&lt;Now&gt;</code> , do not specify a <code>&lt;durationOffset&gt;</code> .
P	required case-sensitive letter	Begins the definition of the offset in terms of years, months and days. This designator is required even if you are using only <code>&lt;#H&gt;&lt;#M&gt;&lt;#S&gt;</code> .
<code>&lt;#Y&gt;&lt;#M&gt;&lt;#D&gt;</code>	optional year, month and day definitions	Defines a number of year(s), month(s), day(s) (case sensitive) to add to the start time range. These are optional if you are using only <code>&lt;#H&gt;&lt;#M&gt;&lt;#S&gt;</code> .
T	case-sensitive letter (required if you are using <code>&lt;#H&gt;&lt;#M&gt;&lt;#S&gt;</code> and optional if you are using only <code>&lt;#Y&gt;&lt;#M&gt;&lt;#D&gt;</code> )	Is required if you are using <code>&lt;#H&gt;&lt;#M&gt;&lt;#S&gt;</code> and optional if you are using only <code>&lt;#Y&gt;&lt;#M&gt;&lt;#D&gt;</code> .
<code>&lt;#H&gt;&lt;#M&gt;&lt;#S&gt;</code>	optional hours, minutes and seconds definitions	Defines an amount of time in hours minutes and seconds (case sensitive) to add to the start of the time range.

**NOTE:** Case is important. Always use uppercase when building an offset. For example, `-P1Y2M3DT4H5M6S` works; `-p1y2m3dt4h5m6s` does not work. Always include the P in front of `<#Y><#M><#D>`; and include the T in front of `<#H><#M><#S>`. You are not required to supply offset elements that are zero. For example, `-P0Y0M5D` is the same as `-P5D`.

## Examples

Depending on the current date, this first example sets up a time range greater than one year. The actual range depends on the current date and time.

- Start time: one year before the start of the current year (this year)
- End time: now

The required syntax to specify this time range would be:

```
fromYear-P1YToNow
```

This second example defines a time range of 10 minutes beginning 30 minutes ago.

- Start time: 30 minutes before the current minute
- End time: 10 minutes after the start time

The required syntax to specify this time range would be:

```
fromMinute-PT30MToPT10M
```

This third example defines a time range of a total of eight hours starting two days from now (in the future).

- Start time: two days after the start of the current day (today)
- End time: eight hours after the start time

The required syntax to specify this time range would be:

```
fromDayP2DToPT8H
```

This fourth example provides a time range for the second quarter of last year (April through June).

- Start time: nine months before the start of the year
- End time: three months after the start time

The required syntax to specify this time range would be:

```
fromYear-P9MToP3M
```

This fifth example sets up a one-year time range beginning now.

- Start time: zero seconds based on the current date and time
- End time: one year in the future

The required syntax to specify this time range would be:

```
fromSecondP0YToP1Y
```

When defining the offset, it is not necessary to specify a zero. In other words, the following are equivalent:

- P1Y0M3DT0H0M0S
- P1Y0M3DT0S
- P1Y3D

Each defines a start or end time range of one year and three days.

## Rollups and aggregation

Rollups and aggregation share the same set of operators, but have different meanings. Rollups are how multiple values (rows) from the same data source are combined. Aggregation is the combination of data values from multiple data sources.

## Functions

Function	Description
And	Logical "and".
Avg	The sum of all values in the data set divided by the count.
count	The number of values in the data set.
first	The first value retrieved by the framework
last	The last value retrieved by framework
max	The largest value in the data set.
min	The smallest value in the data set.
Or	Logical "or".
Mode	The statistical mode is the most frequently occurring number in the combination
Median	The value in the middle of a sorted numeric data set. This number separates the higher half from the lower half of the set.
Range	The statistical range is the difference between the largest and smallest values in the numeric data set.
Sum	All values added together.

## Intervals

Intervals represent the duration between two periods of time. This enables trends, where each row represents an interval, such as daily or monthly. The framework aligns timestamps to the beginning of the assigned interval. So, for an interval of `oneDay`, it aligns a time range that begins at 3 p.m. to 12 a.m. This may be subtle, but it is important to understand the data you are viewing.

### Constants

- none
- second
- minute
- fiveMinutes
- tenMinutes
- fifteenMinutes
- twentyMinutes
- thirtyMinutes
- hour
- twoHours
- threeHours
- fourHours
- sixHours
- twelveHours
- day

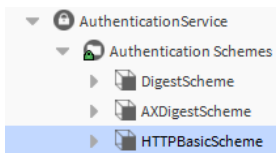
- twoDays
- week
- month
- year

## Setting up user authentication

Access to a station database requires user authentication, which is managed by the station's **AuthenticationService**. As with physical access, programmatic access requires authentication using the HTTPBasicScheme (HTTP Basic Authentication Scheme). Consider using a separate user for each type of access (physical access, programmatic access, etc.). This practice provides additional security as each user requires only the minimum number of access rights necessary to accomplish a specific task. Using roles and tagged categories allows for highly-configurable permissions for accessing various station components.

**Prerequisites:** You have administrative rights. The station is open in Workbench.

**Step 1** Open the **baja** palette.



**Step 2** Add the **HTTPBasicScheme** under the **Services→Authentication Service→Authentication Schemes** node in the nav tree.

**Step 3** Create a role to assign to users based on the type of access.

For example, users who are permitted to create and view web charts may be assigned the "NA\_charts" role. Permissions for this role might allow a user to read from the database but not invoke actions or write records to it.

Users who are permitted to query the station database with API calls may be assigned a "NA\_API" role. Permissions for this role might allow a user to read from the database and invoke actions, but not write records to it.

**Step 4** Expand **Services→UserService** in the nav tree and double-click the user name you intend to use to access the station database.

The **Edit User** window opens.



Property Sheet

api (User)

Full Name

Enabled ☒ true

Expiration ☒ Never Expires ☐ Expires On 07-Feb-2017 11:59 PM EST

Lock Out ☒ false

Language

Email

Authenticator

Password

Confirm

ForceResetAtNextLogin ☒ false

Expiration ☒ Never Expires ☐ Expires On 07-Feb-2017 11:59 PM EST

Facets

Time Format (default)

Unit Conversion None

Nav File null

Prototype Name

Network User ☒ false

Cell Phone Number

Authentication Scheme Name HTTPBasicScheme

Roles ☒ NA\_API ☐ admin

**Step 5** Select HTTPBasicScheme from the **Authentication Scheme Name** drop-down list, assign the role you created, and click **OK**.



# Chapter 2 Messages

## Topics covered in this chapter

- ◆ GetNode
- ◆ GetChildNodes
- ◆ GetValue
- ◆ GetTrend
- ◆ GetRollup
- ◆ Invoke
- ◆ Query
- ◆ Subscribe
- ◆ PollSubscription
- ◆ Unsubscribe

The API supports a number of request messages.

- To make examples fit on a printed page, this document breaks and continues long lines of code, particularly those that contain an ORD, on the next line. In reality, you do not need to break up your lines of code in this manner.
- Attribute names used in Web API requests are case sensitive. Executing a request with an incorrect attribute name returns a default result instead of the expected result.

## GetNode

GetNode provides the details about a single node in the Niagara config space or cached hierarchy tree ("hier:/"). This does not need to be called for full tree traversal. Starting at any node, you can use GetChildNodes exclusively.

### Request example

```
{
  "requests": [{
    "message": "GetNode",
    "node" : "slot:/TridiumEMEA/MainPower",
  }]
}
```

### Request elements

Request element	Value	Description
message	GetNode	Identifies the type of request being submitted.
node (required)	ORD or hierarchy	Defines the ORD to the desired node.

### Response example

```
{ "responses" : [
{
  "message" : "GetNode",
  "node" : "slot:/TridiumEMEA/MainPower",
  "name" : "MainPower",
  "icon" : "/ord?module://icons:/x16/folder.png",
  "hasChildren" : true,
  "data" : [
    {
```



```

    "data" : "slot:TridiumEMEA_Main",
    "name" : "TridiumEMEA_Main",
    "type" : "numeric",
    "icon" : "/ord?module://icons/x16/schedule.png",
    "hasTrend" : false,
    "unit" : {
      "name" : "kilowatt",
      "symbol" : "kW"
    }
  },
],
}
}]

```

## Response elements

In addition to the following, the system echoes back all request elements.

Response element	Value	Description
actions	list	Lists the actions available on the node:  <code>name</code> returns the action slot name.  <code>display</code> returns the display name.  <code>parameter</code> is only present if the action takes a parameter. This could represent a single object or a tree of complex objects. If the parameter represents a complex tree, each property is represented as a child object whose name is the slotName. The properties are the same as those described for the data element.  <code>returnType</code> identifies the data type.
attributes	text	Returns a map of attribute names and their associated values.
data	ORD	Returns the data definition ORD.
Name, name, or Display	read-only	Returns the display name of the object.
falseText	text	Returns descriptive word(s) when the result is <code>false</code> , such as <code>off</code> or <code>closed</code> .
hasChildren	read only	Indicates if the node has child nodes.
hasTrend	read only	Indicates if the node has histories.
Interval or interval	optional drop-down list or ORD parameter ( <code>interval=option</code> ); defaults to <code>Minute</code>	Refers to the <code>BIInterval</code> component, which the framework uses to identify the time between values in a trend (time series). When specified, a rollup is required, which causes the system to combine all values that fall into a single interval.  Options range from <code>None</code> to a <code>Year</code> .  Above the drop-down list, the <b>Use This Value</b> check box turns on and off the check box next to <b>Interval</b> in the <b>Settings</b> window (you access this window by clicking the Edit button (  ), followed by clicking the Settings button (  ) on the chart). The availability of this box provides an easy way for a user to enable and disable the use of the intervals in chart calculations.

Response element	Value	Description
max	number	Defines the largest number possible.
min	number	Defines the smallest number.
Range or range	string	Identifies an enum range as a list of string values.
trueText	text	Returns descriptive word(s) when the result is <code>true</code> , such as <code>on</code> , or <code>open</code> .
Type or type	boolean, numeric, enum, string, formula, data definition	Defines or returns the object type, which depends on where you are in the system.
unit	unit of measure	Returns the unit name and display symbol.

## GetChildNodes

GetChildNodes provides the details about the child nodes of a node in the Niagara config space or cached hierarchy tree (for example: "hier:/TridiumBuildings").

### Request example

```
{
  "requests" : [{,
    "message" : "GetChildNodes",
    "node" : "slot:/TridiumEMEA/MainPower",
  }]
}
```

### Request elements

Request element	Value	Description
message	text	GetChildNodes
node	Required	Defines the PRD to the desired node.

### Response example

```
{ "responses" : [
{
  "message" : "GetChildNodes",
  "node" : "slot:/Logic/TridiumEMEA",
  "nodes" : [
    {
      "node" : "slot:/Logic/TridiumEMEA/MainPower",
      "name" : "MainPower",
      "icon" : "ord?module:icons:/x16/folder.png",
      "hasChildren" : "true",
      "data" : [
        {
          "data" : "slot:TridiumEMEA_Main",
          "name" : "TridiumEMEA_Main",
          "type" : "numeric",
          "icon" : "/ord?module://icons/x16/schedule.png",
          "hasTrend" : "false",
          "unit" : {
            "name" : "kilowatt",
```

```

        "symbol" : "kW",
      }
    },
    {
      "data" : "slot:Area",
      "name" : "Area",
      "type" : "numeric",
      "icon" : "/ord?mobile://icons/x16/control/numericPoint.png",
      "hasTrend" : false
    }
  ]
},
{
  "node" : "slot:/Logic/TridiumEMEA/HVAC",
  "name" : "HVAC",
  "icon" : "/ord?module://icons/x16/folder.png",
  "hasChildren" : true
},
{
  "node" : "slot:/Logic/TridiumEMEA/TridiumEMEA_OutsideAirTemp",
  "name" : "TridiumEMEA_OutsideAirTemp",
  "icon" : "/ord?module://icons/x16/schedule.png",
  "hasChildren" : true,
  "data" : [
    {
      "data" : "slot:status",
      "name" : "Status",
      "type" : "status",
      "icon" : "/ord?module://icons/x16/object.png",
      "hasTrend" : false
    },
    ...
    {
      "data" : "slot:scanLimit",
      "name" : "Scan Limit",
      "type" : "numeric",
      "icon" : "/ord?module://icons/x16/object.png",
      "hasTrend" : false
    },
    {
      "data" : "slot:facets",
      "name" : "Facets",
      "type" : "baja:Facets",
      "icon" : "/ord?module://icons/x16/object.png",
      "hasTrend" : false
    },
    {
      "data" : "slot:lastModified",
      "name" : "Last Modified",
      "type" : "time",
      "icon" : "/ord?module://icons/x16/object.png",
      "hasTrend" : false
    },
    {
      "data" : "slot:outSource",
      "name" : "Out Source",
      "type" : "string",
      "icon" : "/ord?module://icons/x16/object.png",
      "hasTrend" : false
    }
  ],
}

```

```

{
  "data" : "slot:out",
  "name" : "Out",
  "type" : "numeric",
  "icon" : "/ord?module://icons/x16/statusNumeric.png",
  "hasTrend" : false,
  "unit" : {
    "name" : "fahrenheit",
    "symbol" : "F"
  }
},
{
  "data" : "slot:in",
  "name" : "In",
  "type" : "numeric",
  "icon" : "/ord?module://icons/x16/statusNumeric.png",
  "hasTrend" : false,
  "unit" : {
    "name" : "fahrenheit",
    "symbol" : "F"
  }
},
{
  "data" : "slot:nextTime",
  "name" : "Next Time",
  "type" : "time",
  "icon" : "/ord?module://icons/x16/object.png",
  "hasTrend" : false
},
{
  "data" : "slot:nextValue",
  "name" : "Next Value",
  "type" : "numeric",
  "icon" : "/ord?module://icons/x16/statusNumeric.png",
  "hasTrend" : false,
  "unit" : {
    "name" : "fahrenheit",
    "symbol" : "F"
  }
},
{
  "data" : "slot:wsAnnotation",
  "name" : "wsAnnotation",
  "type" : "baja:WsAnnotation",
  "icon" : "/ord?module://icons/x16/object.png",
  "hasTrend" : false
},
{
  "data" : "slot:outsideAirTemp",

  "name" : "outsideAirTemp",
  "type" : "numeric",
  "icon" : "/ord?module://icons/x16/history.png",
  "hasTrend" : false,
  "unit" : {
    "name" : "kilowatt",
    "symbol" : "kW"
  }
}

```

```

    }
  ],
  "actions" : [
    {
      "action" : "cleanup",
      "display" : "Cleanup"
    }
  ]
},
{
  "node" : "slot:/Logic/TridiumEMEA/Schedules",
  "name" : "Schedules",
  "icon" : "/ord?module://icons/x16/folder.png",
  "hasChildren" : true,
  "data" : [
    {
      "data" : "slot:Occupancy",
      "name" : "Occupancy",
      "type" : "enum",
      "icon" : "/ord?module://icons/x16/schedule.png",
      "hasTrend" : false,
      "range" : [
        "Occupied",
        "Unoccupied",
        "Bypass",
        "Standby",
        "Occ Null"
      ]
    },
    {
      "data" : "slot:Not",
      "name" : "Not",
      "type" : "boolean",
      "icon" : "/ord?module://icons/x16/control/logic/not.png",
      "hasTrend" : false,
      "trueText" : "true",
      "falseText" : "false"
    },
    {
      "data" : "slot:Holidays",
      "name" : "Holidays",
      "type" : "boolean",
      "icon" : "/ord?module://icons/x16/schedule.png",
      "hasTrend" : false,
      "trueText" : "true",
      "falseText" : "false"
    },
    ...
    {
      "data" : "slot:OccupiedHeating",
      "name" : "OccupiedHeating",
      "type" : "boolean",
      "icon" : "/ord?module://icons/x16/schedule.png",
      "hasTrend" : false,
      "trueText" : "true",
      "falseText" : "false"
    },
    {
      "data" : "slot:UnoccupiedCooling",

```



```

        "name" : "UnoccupiedCooling",
        "type" : "boolean",
        "icon" : "/ord?module://icons/x16/schedule.png",
        "hasTrend" : false,
        "trueText" : "true",
        "falseText" : "false"
    },
    {
        "data" : "slot:OccupiedCooling",
        "name" : "OccupiedCooling",
        "type" : "boolean",
        "icon" : "/ord?module://icons/x16/schedule.png",
        "hasTrend" : false,
        "trueText" : "true",
        "falseText" : "false"
    }
]
}
]
}
[]

```

### Response elements

In addition to the following, the system echoes back all request elements.

Response element	Value	Description
nodes	various	Lists the nodes. The information provided is identical to that returned for GetNode.

## GetValue

GetValue provides the current value of a data definition on a node in the data model tree.

### Request example

```

{"requests":
  [ {
    "message" : "GetValue",
    "mode" : "hierarchy:/TridiumBuildings/UK",
    "data" : "hs:power",
  }
],
}

```

### Request elements



Request element	Value	Description
message	text	GetValue
Node or node (required)	ORD	Defines the ORD to the desired slot.
Data (property) or data (ORD parameter)	tag or algorithm name	Specifies the tag used to retrieve data. This tag can be from the Haystack dictionary, Niagara dictionary or any other custom tag dictionary. Instead of a tag, this value can specify an algorithm. This is how output from one algorithm becomes the input data source to another algorithm. The prefix for algorithms is <code>alg:.</code>

Request element	Value	Description
Aggregation or aggregation	Count, First, Last, Avg, Sum, Min, Max, And, Or. Defaults to the setting of this property in the data definition.	Defines the mathematical function to use to aggregate the data. To use this property, <b>Use This Aggregation</b> must be set to true.
Rollup (property) or rollup (ORD parameter)	optional drop-down list (property, defaults to First) or ORD parameter (rollup= <b>option</b> )	<p>Defines the mathematical function to be used to combine data from a single source.</p> <p>And returns the logical "and" of Boolean values.</p> <p>Avg returns the statistical mean, which is determined by calculating the sum of all values and dividing by the number of values.</p> <p>Count returns the total number or quantity of values in a combination. If you request this value on a binding in a PX view, the system counts the number of values based on the properties defined by the data source block and the algorithm's property sheet.</p> <p>First returns the first value in the combination. This generates the fastest result.</p> <p>Last returns the last value in the combination.</p> <p>Max returns the highest value in the combination.</p> <p>Median returns the value in the middle of a sorted combination—the number that separates the higher half from the lower half.</p> <p>Min returns the lowest value in the combination.</p> <p>Mode returns the statistically most frequently occurring number in the combination.</p> <p>Or returns the logical "or" of Boolean values.</p> <p>Range returns the statistical difference between the largest and smallest values in the combination.</p> <p>Sum adds together all values in the combination resulting in a single value.</p>

### Response example

```
{
  "responses" : [
    {
      "message" : "GetValue",
      "node" : "hierarchy:/TridiumBuildings/UK",
      "data" : "hs:power",
      "value" : "150.2304365555372",
      "status" : "ok"
    }
  ]
}
```

## Response elements

Response Elements	Value	Description
message	text	Identifies the type of request being returned.
Data (property) or data (ORD parameter)	tag or algorithm name	Specifies the tag used to retrieve data. This tag can be from the Haystack dictionary, Niagara dictionary or any other custom tag dictionary. Instead of a tag, this value can specify an algorithm. This is how output from one algorithm becomes the input data source to another algorithm. The prefix for algorithms is <code>alg:</code> .
Display, display, or Name	text	Defines or returns the object name.
falseText	text	Returns descriptive word(s) when the result is <code>false</code> , such as <code>off</code> or <code>closed</code> .
Interval or interval	optional drop-down list or ORD parameter ( <code>interval=option</code> ); defaults to <code>Minute</code>	<p>Refers to the <code>BInterval</code> component, which the framework uses to identify the time between values in a trend (time series). When specified, a rollup is required, which causes the system to combine all values that fall into a single interval.</p> <p>Options range from <code>None</code> to a <code>Year</code>.</p> <p>Above the drop-down list, the <b>Use This Value</b> check box turns on and off the check box next to <b>Interval</b> in the <b>Settings</b> window (you access this window by clicking the Edit button (  ), followed by clicking the Settings button (  ) on the chart). The availability of this box provides an easy way for a user to enable and disable the use of the intervals in chart calculations.</p>
Max	number	Returns a numeric value
Min	number	Returns a numeric value
Name, name, or Display	read-only	Returns the display name of the object.
Node or node (required)	ORD	Defines the ORD to the desired slot.
Range or range	string	Identifies an enum range as a list of string values.
Rollup (property) or rollup (ORD parameter)	optional drop-down list (property, defaults to <code>First</code> ) or ORD parameter ( <code>rollup=option</code> )	<p>Defines the mathematical function to be used to combine data from a single source.</p> <p>And returns the logical “and” of Boolean values.</p> <p><code>Avg</code> returns the statistical mean, which is determined by calculating the sum of all values and dividing by the number of values.</p> <p><code>Count</code> returns the total number or quantity of values in a combination. If you request this value on a binding in a PX view, the system counts the number of values based on the properties defined by the data source block and the algorithm’s property sheet.</p> <p><code>First</code> returns the first value in the combination. This generates the fastest result.</p>

Response Elements	Value	Description
		<p><b>Last</b> returns the last value in the combination.</p> <p><b>Max</b> returns the highest value in the combination.</p> <p><b>Median</b> returns the value in the middle of a sorted combination—the number that separates the higher half from the lower half.</p> <p><b>Min</b> returns the lowest value in the combination.</p> <p><b>Mode</b> returns the statistically most frequently occurring number in the combination.</p> <p><b>Or</b> returns the logical “or” of Boolean values.</p> <p><b>Range</b> returns the statistical difference between the largest and smallest values in the combination.</p> <p><b>Sum</b> adds together all values in the combination resulting in a single value.</p>
trueText	text	Returns descriptive word(s) when the result is <code>true</code> , such as <code>on</code> , <code>or</code> <code>open</code> .
Type or type	boolean, numeric, enum, string, formula, data definition	Defines or returns the object type, which depends on where you are in the system.
unit	unit of measure	Returns the unit name and display symbol.
Value	read-only	Returns the current raw value of the device point.

## GetTrend

GetTrend provides the time series for the data definition on a node.



### Request example

```
{
  "requests": [{
    "message": "GetTrend",
    "node": "hierarchy:/TridiumBuildings",
    "data": "hs:power",
    "timerange": "yesterday",
    "interval": "hour"
  }]
}
```

### Request elements

Request element	Value	Description
message	text	GetTrend
Node or node (required)	ORD	Defines the ORD to the desired slot.

Request element	Value	Description
Data (property) or data (ORD parameter)	tag or algorithm name	Specifies the tag used to retrieve data. This tag can be from the Haystack dictionary, Niagara dictionary or any other custom tag dictionary. Instead of a tag, this value can specify an algorithm. This is how output from one algorithm becomes the input data source to another algorithm. The prefix for algorithms is <code>alg:</code> .
Rollup (property) or rollup (ORD parameter)	optional drop-down list (property, defaults to <code>First</code> ) or ORD parameter ( <code>rollup=option</code> )	<p>Defines the mathematical function to be used to combine data from a single source.</p> <p><code>And</code> returns the logical “and” of Boolean values.</p> <p><code>Avg</code> returns the statistical mean, which is determined by calculating the sum of all values and dividing by the number of values.</p> <p><code>Count</code> returns the total number or quantity of values in a combination. If you request this value on a binding in a PX view, the system counts the number of values based on the properties defined by the data source block and the algorithm’s property sheet.</p> <p><code>First</code> returns the first value in the combination. This generates the fastest result.</p> <p><code>Last</code> returns the last value in the combination.</p> <p><code>Max</code> returns the highest value in the combination.</p> <p><code>Median</code> returns the value in the middle of a sorted combination—the number that separates the higher half from the lower half.</p> <p><code>Min</code> returns the lowest value in the combination.</p> <p><code>Mode</code> returns the statistically most frequently occurring number in the combination.</p> <p><code>Or</code> returns the logical “or” of Boolean values.</p> <p><code>Range</code> returns the statistical difference between the largest and smallest values in the combination.</p> <p><code>Sum</code> adds together all values in the combination resulting in a single value.</p>
Aggregation or aggregation	<code>Count, First, Last, Avg, Sum, Min, Max, And, Or</code> . Defaults to the setting of this property in the data definition.	Defines the mathematical function to use to aggregate the data. To use this property, <b>Use This Aggregation</b> must be set to <code>true</code> .

Request element	Value	Description
Interval or interval	optional drop-down list or ORD parameter ( <code>interval=option</code> ); defaults to <code>Minute</code>	<p>Refers to the <code>BIInterval</code> component, which the framework uses to identify the time between values in a trend (time series). When specified, a rollup is required, which causes the system to combine all values that fall into a single interval.</p> <p>Options range from <code>None</code> to a <code>Year</code>.</p> <p>Above the drop-down list, the <b>Use This Value</b> check box turns on and off the check box next to <b>Interval</b> in the <b>Settings</b> window (you access this window by clicking the Edit button (  ), followed by clicking the Settings button (  ) on the chart). The availability of this box provides an easy way for a user to enable and disable the use of the intervals in chart calculations.</p>
Time Range or timeRange	drop-down list or ORD parameter ( <code>timerange=option</code> ); the default is <code>Today</code> (current value)	<p>Defines the time period over which to combine the data in a rollup.</p> <p>This property is required for rollup requests (<code>analyticRollup</code>), trends (<code>analyticTrend</code>), and rollup bindings. It is optional elsewhere.</p> <p>It is not used on components whose <b>Use Request Time Range</b> property is <code>true</code> and the request specifies a time range.</p> <p>Options range from <code>From</code> to <code>All</code>. Time Range defaults to <code>Today</code>, which causes the framework to return a point's current value.</p>

### Response example



```
{
  "responses" : [
    {
      "message" : "GetTrend",
      "node" : "hierarchy:/TridiumBuildings",
      "data" : "hs:power",
      "timerange" : "yesterday",
      "interval" : "hour",
      "rows" : [
        [
          "2016-05-17T00:00:00.000-04:00",
          "1611.7379056215286",
          "ok"
        ],
        [
          "2016-05-17T01:00:00.000-04:00",
          "1644.921025633812",
          "ok"
        ],
        [
          "2016-05-17T02:00:00.000-04:00",
          "1686.7962741851807",
          "ok"
        ],
        [
          "2016-05-17T03:00:00.000-04:00",
          "1695.273845076561",
          "ok"
        ]
      ]
    }
  ]
}
```

```

    ],
    ...
    [
        "2016-05-17T23:00:00.000-04:00",
        "1688.156940460205",
        "ok"
    ]
  ]
}
]]

```

## Response elements

Response Elements	Value	Description
message	text	Identifies the type of request being returned.
Aggregation or aggregation	Count, First, Last, Avg, Sum, Min, Max, And, Or. Defaults to the setting of this property in the data definition.	Defines the mathematical function to use to aggregate the data. To use this property, <b>Use This Aggregation</b> must be set to <code>true</code> .
Data (property) or data (ORD parameter)	tag or algorithm name	Specifies the tag used to retrieve data. This tag can be from the Haystack dictionary, Niagara dictionary or any other custom tag dictionary. Instead of a tag, this value can specify an algorithm. This is how output from one algorithm becomes the input data source to another algorithm. The prefix for algorithms is <code>alg:</code> .
falseText	text	Returns descriptive word(s) when the result is <code>false</code> , such as <code>off</code> or <code>closed</code> .
Interval or interval	optional drop-down list or ORD parameter ( <code>interval=option</code> ); defaults to <code>Minute</code>	<p>Refers to the <code>Interval</code> component, which the framework uses to identify the time between values in a trend (time series). When specified, a rollup is required, which causes the system to combine all values that fall into a single interval.</p> <p>Options range from <code>None</code> to a <code>Year</code>.</p> <p>Above the drop-down list, the <b>Use This Value</b> check box turns on and off the check box next to <b>Interval</b> in the <b>Settings</b> window (you access this window by clicking the Edit button (  ), followed by clicking the Settings button (  ) on the chart). The availability of this box provides an easy way for a user to enable and disable the use of the intervals in chart calculations.</p>
Max	number	Returns a numeric value
Min	number	Returns a numeric value
Node or node (required)	ORD	Defines the ORD to the desired slot.
Range or range	string	Identifies an enum range as a list of string values.
Rollup (property) or rollup (ORD parameter)	optional drop-down list (property, defaults to <code>First</code> )	Defines the mathematical function to be used to combine data from a single source.

Response Elements	Value	Description
	or ORD parameter (rollup= <b>option</b> )	<p><b>And</b> returns the logical “and” of Boolean values.</p> <p><b>Avg</b> returns the statistical mean, which is determined by calculating the sum of all values and dividing by the number of values.</p> <p><b>Count</b> returns the total number or quantity of values in a combination. If you request this value on a binding in a PX view, the system counts the number of values based on the properties defined by the data source block and the algorithm’s property sheet.</p> <p><b>First</b> returns the first value in the combination. This generates the fastest result.</p> <p><b>Last</b> returns the last value in the combination.</p> <p><b>Max</b> returns the highest value in the combination.</p> <p><b>Median</b> returns the value in the middle of a sorted combination—the number that separates the higher half from the lower half.</p> <p><b>Min</b> returns the lowest value in the combination.</p> <p><b>Mode</b> returns the statistically most frequently occurring number in the combination.</p> <p><b>Or</b> returns the logical “or” of Boolean values.</p> <p><b>Range</b> returns the statistical difference between the largest and smallest values in the combination.</p> <p><b>Sum</b> adds together all values in the combination resulting in a single value.</p>
rows	list of row lists	The columns of each row list are : Timestamp (ISO-8601–encoded timestamp), Value and Status.
Time Range or timeRange	drop-down list or ORD parameter (timerange= <b>option</b> ); the default is <b>Today</b> (current value)	<p>Defines the time period over which to combine the data in a rollup.</p> <p>This property is required for rollup requests (analyticRollup), trends (analyticTrend), and rollup bindings. It is optional elsewhere.</p> <p>It is not used on components whose <b>Use Request Time Range</b> property is <b>true</b> and the request specifies a time range.</p> <p>Options range from <b>From</b> to <b>All</b>. Time Range defaults to <b>Today</b>, which causes the framework to return a point’s current value.</p>
trueText	text	Returns descriptive word(s) when the result is <b>true</b> , such as <b>on</b> , <b>or</b> <b>open</b> .
Type or type	boolean, numeric, enum, string, formula, data definition	Defines or returns the object type, which depends on where you are in the system.
unit	unit of measure	Returns the unit name and display symbol.



## GetRollup



GetRollup combines trend data into a single value.

### Request example

```
{ "requests":
  [ {
    "message": "GetRollup",
    "node": "hierarchy:/TridiumBuildings",
    "data" : "hs:power",
    "rollup": "avg",
    "aggregation": "sum"
  }
  ]
}
```

### Request elements



Request element	Value	Description
message	text	GetRollup
Node or node (required)	ORD	Defines the ORD to the desired slot.
Data (property) or data (ORD parameter)	tag or algorithm name	Specifies the tag used to retrieve data. This tag can be from the Haystack dictionary, Niagara dictionary or any other custom tag dictionary. Instead of a tag, this value can specify an algorithm. This is how output from one algorithm becomes the input data source to another algorithm. The prefix for algorithms is <code>alg:</code> .
Aggregation or aggregation	Count, First, Last, Avg, Sum, Min, Max, And, Or. Defaults to the setting of this property in the data definition.	Defines the mathematical function to use to aggregate the data. To use this property, <b>Use This Aggregation</b> must be set to <code>true</code> .
Rollup (property) or rollup (ORD parameter)	optional drop-down list (property, defaults to <code>First</code> ) or ORD parameter ( <code>rollup=option</code> )	<p>Defines the mathematical function to be used to combine data from a single source.</p> <p><code>And</code> returns the logical "and" of Boolean values.</p> <p><code>Avg</code> returns the statistical mean, which is determined by calculating the sum of all values and dividing by the number of values.</p> <p><code>Count</code> returns the total number or quantity of values in a combination. If you request this value on a binding in a PX view, the system counts the number of values based on the properties defined by the data source block and the algorithm's property sheet.</p> <p><code>First</code> returns the first value in the combination. This generates the fastest result.</p> <p><code>Last</code> returns the last value in the combination.</p> <p><code>Max</code> returns the highest value in the combination.</p>

Request element	Value	Description
		<p><b>Median</b> returns the value in the middle of a sorted combination—the number that separates the higher half from the lower half.</p> <p><b>Min</b> returns the lowest value in the combination.</p> <p><b>Mode</b> returns the statistically most frequently occurring number in the combination.</p> <p><b>Or</b> returns the logical “or” of Boolean values.</p> <p><b>Range</b> returns the statistical difference between the largest and smallest values in the combination.</p> <p><b>Sum</b> adds together all values in the combination resulting in a single value.</p>
Interval or interval	optional drop-down list or ORD parameter ( <code>interval=option</code> ); defaults to <code>Minute</code>	<p>Refers to the <code>BIInterval</code> component, which the framework uses to identify the time between values in a trend (time series). When specified, a rollup is required, which causes the system to combine all values that fall into a single interval.</p> <p>Options range from <code>None</code> to a <code>Year</code>.</p> <p>Above the drop-down list, the <b>Use This Value</b> check box turns on and off the check box next to <b>Interval</b> in the <b>Settings</b> window (you access this window by clicking the Edit button , followed by clicking the Settings button  on the chart). The availability of this box provides an easy way for a user to enable and disable the use of the intervals in chart calculations.</p>
Time Range or timeRange	drop-down list or ORD parameter ( <code>timerange=option</code> ); the default is <code>Today</code> (current value)	<p>Defines the time period over which to combine the data in a rollup.</p> <p>This property is required for rollup requests (<code>analyticRollup</code>), trends (<code>analyticTrend</code>), and rollup bindings. It is optional elsewhere.</p> <p>It is not used on components whose <b>Use Request Time Range</b> property is <code>true</code> and the request specifies a time range.</p> <p>Options range from <code>From</code> to <code>All</code>. Time Range defaults to <code>Today</code>, which causes the framework to return a point’s current value.</p>

### Response example

```
{ "responses" : [
{
  "message" : "GetRollup",
  "node" : "hierarchy:/TridiumBuildings",
  "data" : "hs:power",
  "rollup" : "avg",
  "aggregation" : "sum",
  "value" : "462.12754219643614",
  "status" : "ok"
}
] }
```

## Response elements

Response Elements	Value	Description
message	text	Identifies the type of request being returned.
Data (property) or data (ORD parameter)	tag or algorithm name	Specifies the tag used to retrieve data. This tag can be from the Haystack dictionary, Niagara dictionary or any other custom tag dictionary. Instead of a tag, this value can specify an algorithm. This is how output from one algorithm becomes the input data source to another algorithm. The prefix for algorithms is <code>alg:</code> .
Display, display, or Name	text	Defines or returns the object name.
falseText	text	Returns descriptive word(s) when the result is <code>false</code> , such as <code>off</code> or <code>closed</code> .
Interval or interval	optional drop-down list or ORD parameter ( <code>interval=option</code> ); defaults to <code>Minute</code>	<p>Refers to the <code>BInterval</code> component, which the framework uses to identify the time between values in a trend (time series). When specified, a rollup is required, which causes the system to combine all values that fall into a single interval.</p> <p>Options range from <code>None</code> to a <code>Year</code>.</p> <p>Above the drop-down list, the <b>Use This Value</b> check box turns on and off the check box next to <b>Interval</b> in the <b>Settings</b> window (you access this window by clicking the Edit button (  ), followed by clicking the Settings button (  ) on the chart). The availability of this box provides an easy way for a user to enable and disable the use of the intervals in chart calculations.</p>
Max	number	Returns a numeric value
Min	number	Returns a numeric value
Name, name, or Display	read-only	Returns the display name of the object.
Node or node (required)	ORD	Defines the ORD to the desired slot.
Range or range	string	Identifies an enum range as a list of string values.
Rollup (property) or rollup (ORD parameter)	optional drop-down list (property, defaults to <code>First</code> ) or ORD parameter ( <code>rollup=option</code> )	<p>Defines the mathematical function to be used to combine data from a single source.</p> <p>And returns the logical “and” of Boolean values.</p> <p><code>Avg</code> returns the statistical mean, which is determined by calculating the sum of all values and dividing by the number of values.</p> <p><code>Count</code> returns the total number or quantity of values in a combination. If you request this value on a binding in a PX view, the system counts the number of values based on the properties defined by the data source block and the algorithm’s property sheet.</p> <p><code>First</code> returns the first value in the combination. This generates the fastest result.</p>

Response Elements	Value	Description
		<p><b>Last</b> returns the last value in the combination.</p> <p><b>Max</b> returns the highest value in the combination.</p> <p><b>Median</b> returns the value in the middle of a sorted combination—the number that separates the higher half from the lower half.</p> <p><b>Min</b> returns the lowest value in the combination.</p> <p><b>Mode</b> returns the statistically most frequently occurring number in the combination.</p> <p><b>Or</b> returns the logical “or” of Boolean values.</p> <p><b>Range</b> returns the statistical difference between the largest and smallest values in the combination.</p> <p><b>Sum</b> adds together all values in the combination resulting in a single value.</p>
Time Range or timeRange	drop-down list or ORD parameter (timerange= <b>option</b> ); the default is <b>Today</b> (current value)	<p>Defines the time period over which to combine the data in a rollup.</p> <p>This property is required for rollup requests (analyticRollup), trends (analyticTrend), and rollup bindings. It is optional elsewhere.</p> <p>It is not used on components whose <b>Use Request Time Range</b> property is <b>true</b> and the request specifies a time range.</p> <p>Options range from <b>From</b> to <b>All</b>. Time Range defaults to <b>Today</b>, which causes the framework to return a point’s current value.</p>
trueText	text	Returns descriptive word(s) when the result is <b>true</b> , such as <b>on</b> , <b>or</b> open.
Type or type	boolean, numeric, enum, string, formula, data definition	Defines or returns the object type, which depends on where you are in the system.
unit	unit of measure	Returns the unit name and display symbol.
Value	read-only	Returns the current raw value of the device point.

## Invoke

Initiates an action on a node.

### Request examples

ORD for Driver Point:

```
{
  "requests": [{
    "message": "Invoke",
    "node": "slot:/NumericWritable",
    "action": "override",
```

```

    "parameter": {
      "duration": 3000,
      "value": 48
    }
  }
}

```

### Request elements

Request elements	Value	Description
message	text	Invoke
Node or node (required)	ORD	Defines the ORD to the desired slot.
action (required)	text	Defines the action to take.
Parameter (optional)	parameter value	This element is needed if the action takes a parameter and you want to change the default value(s). If the parameter is a simple type, you need only provide the simple value. If the parameter is a complex type, provide a map where you name each value with the slot name of the property.

### Response examples

```

{ "responses" : [
  {
    "message" : "Invoke",
    "node" : "slot:/Logic/Alarm",
    "action" : "override",
    "parameter" : {
      "duration" : 300000,
      "value" : 76
    }
  }
]
}

```

### Response elements

Response Elements	Value	Description
message	text	Identifies the type of request being returned.
Node or node (required)	ORD	Defines the ORD to the desired slot.
action (required)	text	Defines the action to take.
Parameter (optional)	parameter value	This element is needed if the action takes a parameter and you want to change the default value(s). If the parameter is a simple type, you need only provide the simple value. If the parameter is a complex type, provide a map where you name each value with the slot name of the property.

## Query

Resolves an ORD and returns the results as a table.

## Request example

Some real examples:

```
{ "requests": [ {
  "message": "Query",
  "node": "slot:/",
  "query": "bql:select * from nAnalytics:AnalyticsNumericInput"
} ] }
```

## Request elements

Request element	Value	Description
message	text	Query
Node or node (required)	ORD	Defines the ORD to the desired slot.
query	ORD	Any ORD.

## Response example

```
{ "responses" : [
{
  "message" : "Query",
  "node" : "slot:/",
  "query" : "bql:select * from nAnalytics:AnalyticsNumericInput",
  "columns" : [
    {
      "name" : "Slot Path",
      "type" : "string"
    },
    {
      "name" : "To String",
      "type" : "string"
    },
    {
      "name" : "Resolved Value Ord",
      "type" : "baja:Ord"
    },
    {
      "name" : "Fault Cause",
      "type" : "string"
    },
    {
      "name" : "Facets",
      "type" : "baja:Facets"
    },
    {
      "name" : "Poll Mode",
      "type" : "enum"
    },
    {
      "name" : "History Is Cov",
      "type" : "boolean"
    }
  ]
}
]
```

```

        "name" : "History",
        "type" : "string"
    },
    {
        "name" : "Visible",
        "type" : "boolean"
    },
    {
        "name" : "Priority",
        "type" : "numeric"
    },
    {
        "name" : "Value",
        "type" : "numeric"
    },
    {
        "name" : "Degrade Mode",
        "type" : "enum"
    },
    {
        "name" : "Definition",
        "type" : "string"
    },
    {
        "name" : "Poll On Demand",
        "type" : "boolean"
    },
    {
        "name" : "History Id",
        "type" : "history:HistoryId"
    },
    {
        "name" : "Value Ord",
        "type" : "string"
    },
    {
        "name" : "Poll Queue",
        "type" : "string"
    }
    {
        "name" : "seriesName",
        "type" : "string"
    }
],
"rows" : [
    [
        "slot:/nAnalytics/templates/CVAHU_Template/SpaceTemp",
        "0.0 °F {ok}",
        "null",
        "",
        "units=u:fahrenheit;°F;(K);*0.5555555555555556+255.37222222222223;
          |precision=i:1|min=d:55.0|max=d:85.0",
        "Poll Value Ord",
        false,
        "/%attr.Facility%/%parent.name%_%name%",
        false,
        17.0,
        0.0,
        "Disable",
    ]
]

```

```

    "definition:Analytics/HVAC/Temperature/Air/Zone$20Air$20Temp",
    true,
    "",
    "slot:/Drivers/NiagaraNetwork/%attr.Facility%
      /points/%parent.name%/%name%",
    "Poll History"
  ]
}

```

## Response elements

In addition to the following, the system echoes back all request elements in the response.

Response element	Value	Description
Columns	List of column definitions	If the ORD resolves to a BITable, returns this list of column definitions.
Rows	List of row lists	If the ORD resolves to a BITable, each row list returns a value for each column, otherwise each row has exactly one value.

## Subscribe

Subscribe creates and adds **GetValue** messages to a subscription. A subscription returns an initial value for each **GetValue**, and after that only for changes of value.

Subscriptions are private to the client session. **Subscription** names need only be unique to the client.

## Request example

```

{
  "message" : "Subscribe",
  "name" : "SubscriptionName",
  "values" :
  [
    {
      "message" : "GetValue",
      "uid" : "identifier uniquely identifying
        this specific GetValue",
      "node" : "slot:/nAnalyticTree/East",
      "data" : "definition:Some/Data/Definition",
      "tags" : ["hvac", "it"],
      "tagMode" : "all",
      "aggregation" : "max",
      "rollup" : "average",
      "interval" : "oneMinute",
    }
  ]
}

```





## Request elements

Request element	Value	Description
message	Subscribe	Identifies the type of request being submitted.
name (of subscription)	text	Defines the subscription name. If the client does not already have a subscription by the given name, the system creates one.  Subscriptions are private to each session and only have to be unique in that context.
Values	multiple	Defines one or more <code>GetValue</code> message requests. See <a href="#">Values, page 41</a> .

## Values

Request element	Value	Description
message	text	Identifies the type of request being returned.
uid	client string	Provides a unique identifier (uid) for the <code>GetValue</code> string. There could be several <code>GetValues</code> for the same node and data, each with different options, such as tags. For this reason, a unique identifier is required.
Node or node (required)	ORD	Defines the ORD to the desired slot.
Definition or data (required)	input ORD for data definition	Identifies the data definition (properties) associated with an input point (nAnalytics variable). No other input point directly contained by the same node can have the same data definition.  BFormats are allowed here.
Tags or tags (optional)	tag text	Defines or returns the tags used for the operation. The framework searches the data model tree for nodes that declare or inherit tags based on the <b>Tag Mode</b> setting.
Tag Mode or tagMode	Any (default) or All	Constrains the search to find nodes with one or more of the defined tags ( <i>Any</i> ) or requires that all tags be present to include the node ( <i>All</i> ).
Aggregation or aggregation	Count, First, Last, Avg, Sum, Min, Max, And, Or. Defaults to the setting of this property in the data definition.	Defines the mathematical function to use to aggregate the data. To use this property, <b>Use This Aggregation</b> must be set to <code>true</code> .

Request element	Value	Description
Rollup (property) or rollup (ORD parameter)	optional drop-down list (property, defaults to <code>First</code> ) or ORD parameter ( <code>rollup=option</code> )	<p>Defines the mathematical function to be used to combine data from a single source.</p> <p><code>And</code> returns the logical “and” of Boolean values.</p> <p><code>Avg</code> returns the statistical mean, which is determined by calculating the sum of all values and dividing by the number of values.</p> <p><code>Count</code> returns the total number or quantity of values in a combination. If you request this value on a binding in a PX view, the system counts the number of values based on the properties defined by the data source block and the algorithm’s property sheet.</p> <p><code>First</code> returns the first value in the combination. This generates the fastest result.</p> <p><code>Last</code> returns the last value in the combination.</p> <p><code>Max</code> returns the highest value in the combination.</p> <p><code>Median</code> returns the value in the middle of a sorted combination—the number that separates the higher half from the lower half.</p> <p><code>Min</code> returns the lowest value in the combination.</p> <p><code>Mode</code> returns the statistically most frequently occurring number in the combination.</p> <p><code>Or</code> returns the logical “or” of Boolean values.</p> <p><code>Range</code> returns the statistical difference between the largest and smallest values in the combination.</p> <p><code>Sum</code> adds together all values in the combination resulting in a single value.</p>
Interval or interval	optional drop-down list or ORD parameter ( <code>interval=option</code> ); defaults to <code>Minute</code>	<p>Refers to the <code>BIInterval</code> component, which the framework uses to identify the time between values in a trend (time series). When specified, a rollup is required, which causes the system to combine all values that fall into a single interval.</p> <p>Options range from <code>None</code> to a <code>Year</code>.</p> <p>Above the drop-down list, the <b>Use This Value</b> check box turns on and off the check box next to <b>Interval</b> in the <b>Settings</b> window (you access this window by clicking the Edit button , followed by clicking the Settings button  on the chart). The availability of this box provides an easy way for a user to enable and disable the use of the intervals in chart calculations.</p>

### Response example

```
{
  "message" : "Subscribe",
  "name" : "SubscriptionName",
  "values" :
  [
    {
```

```

    "message" : "GetValue",
    "uid" : "identifier uniquely identifying
              this specific GetValue",
    "node" : "slot:/nAnalyticTree/East",
    "data" : "definition:Some/Data/Definition",
    "tags" : ["hvac", "it"],
    "tagMode" : "all",
    "aggregation" : "max",
    "rollup" : "average",
    "interval" : "oneMinute",
    "type" : "numeric",
    "value" : "20.20",
    "display" : "20.20{ok}",
  }
}

```

## Response elements



The response is the same as for `PollSubscription` (except for the message name). Only some response values may be subscribed. The server may break up a very large response into smaller sets of responses.

Response element	Value	Description
message	Subscribe	Identifies the type of request being submitted.
name (of subscription)	text	Defines the subscription name. If the client does not already have a subscription by the given name, the system creates one.  Subscriptions are private to each session and only have to be unique in that context.
Values	multiple	Defines one or more <code>GetValue</code> message requests. See <a href="#">Values, page 43</a> .

## Values

Property	Value	Description
message	text	Identifies the type of request being returned.
uid	client string	Provides a unique identifier (uid) for the <code>GetValue</code> string. There could be several <code>GetValues</code> for the same node and data, each with different options, such as tags. For this reason, a unique identifier is required.
Node or node (required)	ORD	Defines the ORD to the desired slot.
Definition or data (required)	input ORD for data definition	Identifies the data definition (properties) associated with an input point (nAnalytics variable). No other input point directly contained by the same node can have the same data definition.  BFormats are allowed here.
Tags or tags (optional)	tag text	Defines or returns the tags used for the operation. The framework searches the data model tree for nodes that declare or inherit tags based on the <code>Tag Mode</code> setting.

Property	Value	Description
Tag Mode or tagMode	Any (default) or All	Constrains the search to find nodes with one or more of the defined tags (Any) or requires that all tags be present to include the node (All).
Aggregation or aggregation	Count, First, Last, Avg, Sum, Min, Max, And, Or. Defaults to the setting of this property in the data definition.	Defines the mathematical function to use to aggregate the data. To use this property, <b>Use This Aggregation</b> must be set to true.

Property	Value	Description
Rollup (property) or rollup (ORD parameter)	optional drop-down list (property, defaults to <code>First</code> ) or ORD parameter ( <code>rollup=option</code> )	<p>Defines the mathematical function to be used to combine data from a single source.</p> <p><code>And</code> returns the logical “and” of Boolean values.</p> <p><code>Avg</code> returns the statistical mean, which is determined by calculating the sum of all values and dividing by the number of values.</p> <p><code>Count</code> returns the total number or quantity of values in a combination. If you request this value on a binding in a PX view, the system counts the number of values based on the properties defined by the data source block and the algorithm’s property sheet.</p> <p><code>First</code> returns the first value in the combination. This generates the fastest result.</p> <p><code>Last</code> returns the last value in the combination.</p> <p><code>Max</code> returns the highest value in the combination.</p> <p><code>Median</code> returns the value in the middle of a sorted combination—the number that separates the higher half from the lower half.</p> <p><code>Min</code> returns the lowest value in the combination.</p> <p><code>Mode</code> returns the statistically most frequently occurring number in the combination.</p> <p><code>Or</code> returns the logical “or” of Boolean values.</p> <p><code>Range</code> returns the statistical difference between the largest and smallest values in the combination.</p> <p><code>Sum</code> adds together all values in the combination resulting in a single value.</p>
Interval or interval	optional drop-down list or ORD parameter ( <code>interval=option</code> ); defaults to <code>Minute</code>	<p>Refers to the <code>BIInterval</code> component, which the framework uses to identify the time between values in a trend (time series). When specified, a rollup is required, which causes the system to combine all values that fall into a single interval.</p> <p>Options range from <code>None</code> to a <code>Year</code>.</p> <p>Above the drop-down list, the <b>Use This Value</b> check box turns on and off the check box next to <b>Interval</b> in the <b>Settings</b> window (you access this window by clicking the Edit button (  ), followed by clicking the Settings button (  ) on the chart). The availability of this box provides an easy way for a user to enable and disable the use of the intervals in chart calculations.</p>

## PollSubscription

Poll subscription returns values from a subscription initially when subscribed, and then when changes occur. There is no guarantee of when the system returns values. The system must poll subscriptions at least once a minute. Otherwise, the subscription terminates on the server and later attempts to poll it result in an error.

### Request example

```
{
```

```

    "message" : "PollSubscription",
    "name" : "SubscriptionName",
  }

```

### Request elements

Request element	Value	Description
message	PollSubscription	Identifies the type of request being submitted.
name (of subscription)	text	<p>Defines the subscription name. If the client does not already have a subscription by the given name, the system creates one.</p> <p>Subscriptions are private to each session and only have to be unique in that context.</p>

### Response example

```

{
  "message" : "PollSubscription",
  "name" : "SubscriptionName",
  "values" :
  [
    {
      "message" : "GetValue",
      "uid" : "identifier uniquely identifying this specific
        GetValue",
      "node" : "slot:/nAnalyticTree/East",
      "data" : "definition:Some/Data/Definition",
      "tags" : ["hvac", "it"],
      "tagMode" : "all",
      "aggregation" : "max",
      "rollup" : "average",
      "interval" : "oneMinute",
    }
  ]
}
{
  "message" : "Query",
  "node" : "slot:/",
  "query" : "bql:select * from baja:Component
}

```

## Response elements

Response element	Value	Description
message	PollSubscription	Identifies the type of request being submitted.
name (of subscription)	text	Defines the subscription name. If the client does not already have a subscription by the given name, the system creates one.  Subscriptions are private to each session and only have to be unique in that context.
Values	multiple	Defines one or more <b>GetValue</b> message requests. See <a href="#">Values, page 47</a> .

## Values

Property	Value	Description
message	text	Identifies the type of request being returned.
uid	client string	Provides a unique identifier (uid) for the <b>GetValue</b> string. There could be several <b>GetValues</b> for the same node and data, each with different options, such as tags. For this reason, a unique identifier is required.
Node or node (required)	ORD	Defines the ORD to the desired slot.
Definition or data (required)	input ORD for data definition	Identifies the data definition (properties) associated with an input point (nAnalytics variable). No other input point directly contained by the same node can have the same data definition.  BFormats are allowed here.
Tags or tags (optional)	tag text	Defines or returns the tags used for the operation. The framework searches the data model tree for nodes that declare or inherit tags based on the <b>Tag Mode</b> setting.
Tag Mode or tagMode	Any (default) or All	Constrains the search to find nodes with one or more of the defined tags ( <b>Any</b> ) or requires that all tags be present to include the node ( <b>All</b> ).
Aggregation or aggregation	Count, First, Last, Avg, Sum, Min, Max, And, Or. Defaults to the setting of this property in the data definition.	Defines the mathematical function to use to aggregate the data. To use this property, <b>Use This Aggregation</b> must be set to <b>true</b> .

Property	Value	Description
Rollup (property) or rollup (ORD parameter)	optional drop-down list (property, defaults to <code>First</code> ) or ORD parameter ( <code>rollup=option</code> )	<p>Defines the mathematical function to be used to combine data from a single source.</p> <p><code>And</code> returns the logical “and” of Boolean values.</p> <p><code>Avg</code> returns the statistical mean, which is determined by calculating the sum of all values and dividing by the number of values.</p> <p><code>Count</code> returns the total number or quantity of values in a combination. If you request this value on a binding in a PX view, the system counts the number of values based on the properties defined by the data source block and the algorithm’s property sheet.</p> <p><code>First</code> returns the first value in the combination. This generates the fastest result.</p> <p><code>Last</code> returns the last value in the combination.</p> <p><code>Max</code> returns the highest value in the combination.</p> <p><code>Median</code> returns the value in the middle of a sorted combination—the number that separates the higher half from the lower half.</p> <p><code>Min</code> returns the lowest value in the combination.</p> <p><code>Mode</code> returns the statistically most frequently occurring number in the combination.</p> <p><code>Or</code> returns the logical “or” of Boolean values.</p> <p><code>Range</code> returns the statistical difference between the largest and smallest values in the combination.</p> <p><code>Sum</code> adds together all values in the combination resulting in a single value.</p>

## Unsubscribe

Unsubscribe removes values from a subscription. To terminate a subscription, let it go unpolled for more than a minute and let the server handle the termination.

### Request example

```
{
  "message" : "Unsubscribe",
  "name" : "SubscriptionName",
  "uids" :
  [
    [
      "message" : "GetValue",
      "firstUid",
      "secondUid"
    ]
  ]
}
```



## Request elements

Request elements	Value	Description
name (of subscription)	text	Defines the subscription name. If the client does not already have a subscription by the given name, the system creates one.  Subscriptions are private to each session and only have to be unique in that context.
Uids	firstUid, secondUid, etc.	One or more unique IDs.

## Response example

```
{  
  "message" : "Unsubscribe",  
  "name" : "SubscriptionName",  
  "uids" :  
    [  
      [  
        "firstUid",  
        "secondUid"  
      ]  
    ]  
}
```

## Response elements

The system echoes back the request.



# Index

## A

aggregation .....	14
Analytics value binding .....	8
authentication	
setting up .....	16

## D

data types .....	9
document change log .....	5

## E

error ids .....	9
-----------------	---

## G

GetChildNodes .....	21
GetNode .....	19
GetRollup .....	33
GetTrend .....	28
GetValue .....	25
guide .....	5

## H

HTTP	
GET .....	8
POST .....	8

## I

infinity	
NaN .....	10
interval	
constants .....	15
intervals .....	15
Invoke .....	36

## J

JSON	
error messaging .....	10
requests .....	8

## M

messages .....	19
----------------	----

## P

PollSubscription .....	45
------------------------	----

## Q

Query .....	37
-------------	----

## R

related documentation .....	5
res[pmse .....	10
response code .....	9
rollups .....	14

## S

Subscribe .....	40
-----------------	----

## T

time range	
absolute ranges .....	11
constants .....	11
scripts .....	12
time ranges .....	11

## U

Unsubscribe .....	48
-------------------	----

## V

value binding .....	8
---------------------	---

## W

web API properties .....	7
web API protocol .....	7