

NAPM Web Services

NAPM Web Services allow you to interact with an appliance or director without using the Web Console or Desktop Console. Any data you might request via the console can be obtained directly using Web Services.

You access Web Services via Web Services URLs. For example, you can manually construct a URL for a particular type of information, go to that URL in a web browser, and copy the data that is returned into a spreadsheet for analysis. Or, you could automate a frequent activity by creating a script that programmatically constructs a URL based on certain criteria and processes the resulting data.

Note the following:

- This feature is supported on 2200 (and higher) appliance models only.
- Because NAPM Web Services use HTTPS, you must put a copy of the appliance's public certificate on your local computer before using a Web Services URL to access NAPM data. You might be able to do this by accessing Web Services through your browser (using the URL in "Supported Services", for example) and downloading the certificate from the browser's exceptions dialog or advanced options.
- If you log in to the Web Console manually, you do not need to include login credentials (username/password) in URLs for the remainder of the active session.
- You cannot use NAPM Web Services to access the "95th Percentile Throughput [kbits/sec]" metric.

The following topics describe various ways of obtaining Web Services URLs:

- "URLs from the Desktop Console" –how to automatically generate a URL to request the data in a displayed table or chart
- "URL Cookbook" –customizable examples of URLs you can use for common requests
- "URL Construction" –how to manually construct a URL for any supported request

You can specify that Web Services results be returned in either comma-separated value (CSV) or XML format. These formats are described "Result Formats".

URLs from the Desktop Console

NAPM can automatically generate the Web Services URL that will request the data displayed in a specific table, chart, or alert log. This approach can be faster than constructing a URL from scratch and is sometimes more convenient than customizing an existing URL. To view these URLs, make sure that the “Display web services URL menu in tables and charts” preference is enabled. You can find this preference under Tools > Preferences > Web Service.

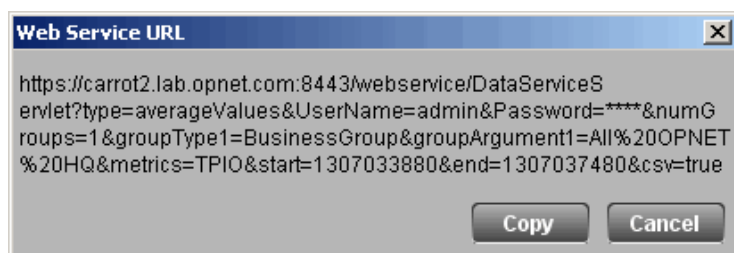
Viewing URLs in the Desktop Console

Use the following procedure to generate a Web Services URL in the Desktop Console.

Procedure 66 Generating a URL in the Desktop Console

1. Open a table, chart, or alert log that contains the information of interest.
2. Select the information of interest:
 - In a table, click on a metric cell.
 - In a multi-group chart, click on a business group in the legend.
3. Right-click on the table or chart.
 - A pop-up menu appears.
4. Select the Web Service URL item.
 - In some cases, this item has a submenu that further defines the desired URL.
 - In some cases (such as Alert Logs), the submenu items vary depending on which table row you clicked on.
 - The Web Service URL is displayed in a dialog box.

Figure 279 Web Service URL Dialog Box



5. Click Copy to copy the URL to your clipboard for use in another application, or click Cancel to close the dialog box.

End of Procedure 66

Result Formats

The data returned in response to a Web Services request can be in either of two formats. Depending on the structure of the request, the results will be returned in either comma-separated value (CSV) or XML format. The following examples show the same data as it would appear in each format.

CSV Format

The comma-separated values format lets you import data into a spreadsheet for further analysis. The first line in a CSV result contains column labels.

CSV Example

```
Group,Time (seconds),Throughput (Inbound) [kb/sec],Throughput (Outbound) [kb/sec]
Default-Internet>Local Traffic>TCP,1291658040,5927.266666666667,5927.266666666667
Default-Internet>Local Traffic>TCP,1291658100,5481.866666666667,5481.866666666667
Default-Internet>Local Traffic>TCP,1291658160,6219.96,6219.96
...
```

XML Format

The XML format returns data suitable for parsing using the XML tools in various programming languages and applications.

XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <timeSeriesValue>
    <group argument="TCP" argumentType="13" hasArgument="true"
      hasData="true" label="IP Protocol" type="IPProtocol">
      <parent>
        <group argument="" argumentType="0" hasArgument="false"
          hasData="true" label="Local Traffic" type="LocalTraffic">
          <parent>
            <group argument="Default-Internet" argumentType="3"
              hasArgument="true" hasData="true"
              label="Business Group" type="BusinessGroup">
              </group>
            </parent>
          </group>
        </parent>
      </group>
    </timeSeriesValue>
    <interval end="1291661640" start="1291658040"/>
    <granularity value="60"/>
    <metricValue unit="kb/sec">
      <metric id="TPI" label="Throughput (Inbound)"/>
      <interval end="1291661640" start="1291658040"/>
      <values>5927.266666666667,5481.866666666667,6219.96,...</values>
    </metricValue>
    <metricValue unit="kb/sec">
      <metric id="TPO" label="Throughput (Outbound)"/>
      <interval end="1291661640" start="1291658040"/>
    </metricValue>
  </result>
</xml>
```

```

        <values>5927.266666666667,5481.866666666667,6219.96,...</values>
    </metricValue>
</timeSeriesValue>
</result>

```

URL Cookbook

This topic includes examples of Web Services URLs for several common types of requests. You can customize these examples to meet many of your data needs. For a detailed description of how to create any Web Services URL, see “URL Construction”.

To use these examples with your appliance, you will need to

- replace “host” with the domain name or IP address of your appliance
- replace “httpsPort” with the httpsPort for the appliance. The default value for httpsPort is 8443, so use this value if the appliance is using the default port.
- replace “pwd” with your admin password
- replace “1299181380” with the start time of interest (in UNIX time)
- replace “1299184980” with the end time of interest (in UNIX time)

Note— There are many online utilities that convert date/time to a UNIX timestamp, such as http://www.onlineconversion.com/unix_time.htm.

Example 1

Let’s say you have an IP address (172.16.1.64) for which you’d like to query the Throughput (Inbound and Outbound) metric for the period from 12:07 to 13:07 on 3 March 2011. For the average over this interval, use the URL

```
https://host:httpsPort/webservice/DataServiceServlet?type=averageValues&UserName=admin&Password=pwd&numGroups=1&groupType=IPAddress&groupArgument1=172.16.1.64&metrics=TPIO&start=1299182820&end=1299186420&csv=true
```

For time series data, use

```
https://host:httpsPort/webservice/DataServiceServlet?type=timeValues&UserName=admin&Password=pwd&numGroups=1&groupType=IPAddress&groupArgument1=172.16.1.64&metrics=TPIO&start=1299182820&end=1299186420&granularity=60&csv=true
```

Example 2

To get the top 20 IP addresses by Throughput (Inbound and Outbound) over the same interval:

```
https://host:httpsPort/webservice/DataServiceServlet?type=topValues&UserName=admin&Password=pwd&numGroups=1&groupType=IPAddress&metrics=TPIO&start=1299181380&end=1299184980&topMetric=TPIO&topCount=20&csv=true
```

Example 3

Let’s repeat examples 1 and 2, but for the Default-Internet business group (BG) instead of IP addresses.

Average:

```
https://host:httpsPort/webservice/DataServiceServlet?type=averageValues&UserName=admin&Password=pwd&numGroups=1&groupType1=BusinessGroup&groupArgument1=Default-Internet&metrics=TPIO&start=1299182820&end=1299186420&csv=true
```

Time series:

```
https://host:httpsPort/webservice/DataServiceServlet?type=timeValues&UserName=admin&Password=pwd&numGroups=1&groupType1=BusinessGroup&groupArgument1=Default-Internet&metrics=TPIO&start=1299182820&end=1299186420&granularity=60&csv=true
```

Top 20 BGs:

```
https://host:httpsPort/webservice/DataServiceServlet?type=topValues&UserName=admin&Password=pwd&numGroups=1&groupType1=BusinessGroup&metrics=TPIO&start=1299182820&end=1299186420&topMetric=TPIO&topCount=20&csv=true
```

Changing Group Types

As you can see from these examples, the only changes necessary to switch from IP addresses to business groups are the values for the `groupType1` and `groupArgument1` parameters — “IPAddress” is replaced by “BusinessGroup” and “172.16.1.64” is replaced by “Default-Internet”, respectively. The common top-level groupTypes you can specify include

- TotalTraffic
- IPAddress
- Prefix24
- DestAS
- BusinessGroup
- IPConversation
 - **Note** — If you want to specify an IP conversation, use a vertical bar (|) and not a colon (:) as the delimiter. For example, specify a range as 1.2.3.4|5.6.7.8 rather than 1.2.3.4:5.6.7.8.
- DefinedApp

You can see a complete list of top-level groupTypes by using the following URL and looking at all rows where the ParentId is 0:

```
https://host:httpsPort/webservice/DataServiceServlet?UserName=admin&Password=pwd&type=groups&csv=true
```

If you want data from a particular drilldown (for example, Business Group > Applications > App), things get a little more complicated. The single pair of `groupType`, `groupArgument` parameters becomes a daisy-chain list, one pair for each level in the drilldown. The “Construction Example (CSV)” demonstrates this kind of drilldown.

Changing Metrics

The preceding examples query for the Throughput (Inbound and Outbound) metric, represented by the metric code TPIO. For other metrics, replace TPIO with a comma-delimited list of metric codes. You can use the following URL to a list of all metrics and their metric codes:

```
https://host:httpsPort/webservice/DataServiceServlet?UserName=admin&Password=pwd&type=metrics&csv=true&see
```

Changing Services

The preceding examples used three different services—`averageValues`, `timeValues`, and `topValues`. Other services are available, as described in “Supported Services”.

URL Construction

This topic explains how to create any Web Services URL. The general format of a Web Services URL is

```
https://<host>:<httpsPort>/webservice/DataServiceServlet?UserName=<user>&Password=<password>&type=<service><arguments>&csv=<format>
```

where:

- `<host>` is the IP address or domain name of the NAPM appliance or director.
- `<httpsPort>` is the port to use for accessing NAPM Web Services on the host (usually 8443).
- `<user>` is the user name for accessing NAPM Web Services.
- `<password>` is the password for accessing NAPM Web Services.
- `<service>` is the requested service. See “Supported Services” for possible values of this parameter.
- `<arguments>` is zero or more parameters representing additional specifications required by some services, as described in “Supported Services”.
- `<format>` is the format for returned data, either `true` for comma-separated values (CSV) or `false` for XML.

Note: The `<user>`, `<password>`, `<service>`, `<arguments>`, and `<format>` parameters can be given in any order. Make sure the first parameter is preceded by “?” and all others by “&”.

Supported Services

Each web service provides access to a particular type of information. The desired service is specified by adding a “type” parameter to the request URL. For example, including `&type=metrics` in an appliance’s Web Services URL returns a list of metrics supported by the appliance. Some services require arguments to further specify the desired results; if needed, these appear as additional URL parameters with the format `&<parameter>=<value>`.

Table 55 summarizes the services that are supported. Follow the service name link for usage details. For a current description of an appliance's supported services and the parameters needed by each, go to

<https://<host>:<port>/webservice/DataServiceServlet?UserName=<user>&Password=<password>>

Table 55 Supported Services

Service	Description	Parameters
"applications"	Returns a list of the applications supported by the appliance.	
"averageValues"	Returns the average values of the specified metrics for a group drilldown.	start end metrics numGroups ¹ groupType ⁱ ₁ groupArgument ⁱ ₁
"businessGroupLinks"	Returns a list of the custom group links supported by the appliance.	
"businessGroups"	Returns a list of the business groups supported by the appliance.	
"containers"	(Director only) Returns a list of business groups available via the director.	
"distributedEngines"	(Director only) Returns a list of appliances in the director's domain.	
"events"	Returns a list of events that satisfy the search criteria.	start end task
"granularities"	Returns a list of the granularities, in seconds, that can be used for time series traffic queries.	
"groupArguments"	Returns the arguments (if any) for a given group.	argumentType
"groups"	Returns a list of the group drill-downs on the appliance.	
"interfaces"	Returns a list of the interfaces supported by the appliance.	
"metrics"	Returns a list of the metrics supported by the appliance.	
"metricsForGroup"	Returns a list of the valid metrics for a group drilldown.	numGroups ¹ ps ₁ groupType ⁱ groupArgument ⁱ ₁
"mifgs"	Returns a list of the MIF groups supported by the appliance.	
"pages"	Returns a list of pages on the appliance that satisfy the search criteria.	searchText maxLimit
"tasks"	Returns a list of user-defined alerts on the appliance.	
"timeValues"	Returns the time series values for a group drilldown.	start end metrics granularity numGroups ¹ groupType ⁱ ₁ groupArgument ⁱ ₁

Table 55 Supported Services (Continued)

Service	Description	Parameters
"topValues"	Returns the top values for a group drilldown.	start end metrics topMetric topCount numGroups ¹ groupType ¹ groupArgument ⁱ
"tosSets"	Returns all ToS set groups supported by the appliance.	
"users"	Returns a list of users on the appliance that satisfy the search criteria.	searchText maxLimit
"vlans"	Returns a list of VLANs supported by the appliance.	

1. The numGroups, groupType, and groupArgument arguments are used together to specify groups. See "Specifying a Group" for details about how to use these arguments.

applications

The applications service returns a list of the applications supported by the appliance. For example, the parameter

```
&type=applications
```

produces a result (as CSV) similar to

```
Id,Type,Name,Description,Status
62,1,ORACLE-TTC-SSL-UDP,UDP Ports: 2484,W
30,1,SAP-ROUTER,TCP Ports: 3299,W
31,1,SAP-OSS,TCP Ports: 3397,W
...
```

averageValues

The averageValues service returns the average values of the specified metrics for a group drilldown.

Parameters:

- start—start time of the interval, in UNIX seconds
- end—end time of the interval, in UNIX seconds
- metrics—comma-separated list of metric IDs, obtained from the "metrics" service (for example, &metrics=TPI,TPO)
- showOnlyValidData—returns an empty token when the data is not valid or does not exist
- group parameters for the desired drilldown (see "Specifying a Group")

For example, the parameters

```
&type=averageValues&start=1294312380&end=1294315980&metrics=TPI,TPO&numGroups=1&groupType1=TotalT
raffic
```

produce a result (as CSV) similar to

```
Group,Throughput (Inbound) [kb/sec],Throughput (Outbound) [kb/sec]
Total Traffic,15268.629744444444,277242.97418
```


businessGroupLinks

The `businessGroupLinks` service returns a list of the custom group links supported by the appliance. For example, the parameter

```
&type=businessGroupLinks
```

produces a result (as CSV) similar to

```
Id,Name,Description,Status
1,New Business Group Link 1,,W
```

businessGroups

The `businessGroups` service returns a list of the business groups supported by the appliance. For example, the parameter

```
&type=businessGroups
```

produces a result (as CSV) similar to

```
Id,Name,Description,Status,IpMembers,AsMembers,BgMembers,Distribution,CollectOnDirector
2,Default-Internet,Default
Internet,W,"1.0.0.0-9.255.255.255,11.0.0.0-172.15.255.255,172.32.0.0-192.167.255.255,192.169.0.0-
255.255.255.255",,,,
4,Default-172.x.x.x,Default      172.16.0.0-172.31.255.255,W,172.16.0.0-172.31.255.255,,,,
9,BG-4,adding bgs for sla
dashboard,W,"192.70.81.157,192.70.81.160,192.70.81.201,192.70.82.105,192.70.82.109
...
```

containers

(Director only) The `containers` service returns a list of business groups available from the director. For example, the parameter

```
&type=containers
```

produces a result (as CSV) similar to

```
Id,ParentId,Type,Name,Description,Status
1,0,108,New Director Group 1,,W
3,1,111,New Director Group 1-cougar4-Total Traffic,member,W
4,1,111,New Director Group 1-cougar5-Total Traffic,member,W
2,0,109,New Director Group 2,,W
5,2,112,New Director Group 2-cougar4-ISP AS 5,member,W
6,2,112,New Director Group 2-cougar5-ISP AS 5,member,W
...
```

distributedEngines

(Director only) The `distributedEngines` service returns a list of appliances in the director's domain. For example, the parameter

```
&type=distributedEngines
```

produces a result (as CSV) similar to

```
Id,Type,Name,Available
9,0,cooper33,true
15,0,cooper31,true
17,0,cooper7,false
```

```
18,0,cooper18,false
19,0,cooper15,true
...
```

events

The `events` service returns a list of events that satisfy the search criteria.

Parameters:

- `start` — start time of the interval, in UNIX seconds
- `end` — end time of the interval, in UNIX seconds
- `task` — label of the task of interest, obtained from the “tasks” service (optional; without this parameter, all events in the specified interval are returned)

For example, the parameters

```
&type=events&start=<time>&end=<time>&task=RT APP ALL RTT IN
```

might produce a result (as CSV) similar to

```
Id,TaskName,TrafficGroup,Start,End,Severity
103872,RT APP ALL RTT IN,NETBIOS-SSN-TCP,1294164120,1294164180,2
103873,RT APP ALL RTT IN,FTP-CONTROL,1294164120,1294164180,2
103871,RT APP ALL RTT IN,SUN-RPC-MOUNT,1294164060,1294164120,2
...
```

granularities

The `granularities` service returns a list of the granularities, in seconds, that can be used for time series traffic queries. For example, the parameter

```
&type=granularities
```

produces a result (as CSV) similar to

```
Granularity (seconds)
60
300
3600
86400
```

groupArguments

The `groupArguments` service returns the arguments (if any) for a given group.

Parameter:

- `argumentType` — ID of the argument type for the group of interest, obtained from the “groups” service

For example, the parameters

```
&type=groupArguments&argumentType=13
```

produce the result (as CSV)

```
Type,Label,Description,IsPrimitive,Options
13,Protocol,The protocol name as a string such as TCP,true,null
```

groups

The `groups` service returns a list of the group drill-downs on the appliance. For example, the parameter

```
&type=groups
```

produces a result (as CSV) similar to

```
NodeId,ParentId,Type,Label,HasArgument,ArgumentType,Argument,HasData
1,0,TotalTraffic,Total Traffic,false,1,,true
2,1,IPProtocol,IP Protocol,true,13,,true
3,2,Applications,Applications,false,0,,false
...
```

Use `NodeId` and `ParentId` to determine what groups you can drill down to. For example, the `TotalTraffic` group has a `NodeId` of 1. You can drill down from `TotalTraffic` to any other group that has a `ParentId` of 1 (such as `IPProtocol`).

If `HasArgument` is `true` for a group, you can use the “groupArguments” service to get information about the argument.

interfaces

The `interfaces` service returns a list of the interfaces supported by the appliance. For example, the parameter

```
&type=interfaces
```

produces a result (as CSV) similar to

```
Id,Name,Description,Status
1,New Router 1:New Interface 1,,W
```

metrics

The `metrics` service returns a list of the metrics supported by the appliance. For example, the parameter

```
&type=metrics
```

produces a result (as CSV) similar to

```
Id,Label
PGSIZEO,Page Size
FILLI,Connection Request Rate (TCP Servers)
VPESDO,End System Delay (Outbound)
...
```

metricsForGroup

The `metricsForGroup` service returns a list of the valid metrics for a group drilldown.

Parameters:

- group parameters for the desired drilldown (see “Specifying a Group”)

For example, the parameters

```
&type=metricsForGroup&numGroups=2&groupType1=TotalTraffic&groupType2=IPProtocol&groupArgument2=Ethernet
```

return the metrics available for a Total Traffic > Ethernet drilldown, with a result (as CSV) similar to

```
Id,Label
FILI,Connection Request Rate (TCP Servers)
RDTI,Retransmission Delay (Inbound)
UEIO,User Response Time (Clients)
...
```

mifgs

The `mifgs` service returns a list of the MIF groups supported by the appliance. For example, the parameter

```
&type=mifgs
```

produces a result (as CSV) similar to

```
Id,Name,Description,Status
1,Monitoring Group 1,,W
```

pages

The `pages` service returns a list of pages on the appliance that satisfy the search criteria.

Parameters:

- `searchText`—text string to search for.
- `maxLimit`—maximum number of results to return. This value must be between 1 and 10000, inclusive.

For example, the parameters

```
&type=pages&maxLimit=10&searchText=http://www.google.com
```

produce a result (as CSV) similar to

```
Id,URL
33,http://www.google.com/
1605,http://www.google.com/s...
2137,http://www.google.com/news
4592,http://www.google.com/maps
...
```

tasks

The `tasks` service returns a list of user-defined alerts on the appliance. For example, the parameter

```
&type=tasks
```

produces a result (as CSV) similar to

```
Id,Type,Label,Description,Status
10152,static-any,IP Throughput,,W
10151,static-any,BG Throughput,,W
...
```

timeValues

The `timeValues` service returns the time series values for a group drilldown.

Parameters:

- `start`—start time of the interval, in UNIX seconds

- `end` — end time of the interval, in UNIX seconds
- `metrics` — comma-separated list of metric IDs, obtained from the “metrics” service (for example, `&metrics=TPI,TPO`)
- `granularity` — desired data granularity, in seconds. Must be one of the values returned by the “granularities” service.
- `showOnlyValidData` — returns an empty token when the data is not valid or does not exist
- `skipZeroOrInvalidRows` — removes rows that have no data from the CSV file
- group parameters for the desired drilldown (see “Specifying a Group”)

For example, the parameters

```
&type=timeValues&start=1294312380&end=1294315980&metrics=TPI,TPO&granularity=60&numGroups=1&groupType1=TotalTraffic
```

produce a result (as CSV) similar to

```
Group,Time (seconds),Throughput (Inbound) [kb/sec],Throughput (Outbound) [kb/sec]
Total Traffic,1294312380,11194.6552,274995.557066666666
Total Traffic,1294312440,27256.6293333333334,276856.450933333337
Total Traffic,1294312500,15141.3394666666666,277310.890533333335
...
```

topValues

The `topValues` service returns the top values for a group drilldown.

Parameters:

- `start` — start time of the interval, in UNIX seconds
- `end` — end time of the interval, in UNIX seconds
- `metrics` — comma-separated list of metric IDs, obtained from the “metrics” service (for example, `&metrics=TPI,TPO`)
- `topMetric` — ID of the top metric, obtained from the “metrics” service
- `topCount` — maximum number of top groups to return. This value must be between 1 and 1000, inclusive.
- `showOnlyValidData` — returns an empty token when the data is not valid or does not exist
- group parameters for the desired drilldown (see “Specifying a Group”)

For example, the parameters

```
&type=topValues&start=1294312380&end=1294315980&metrics=TPI,TPO&topMetric=TPI&topCount=10&numGroups=1&groupType1=TotalTraffic
```

produce a result (as CSV) similar to

```
Group,Throughput (Inbound) [kb/sec],Throughput (Outbound) [kb/sec]
Total Traffic,15268.6297444444444,277242.97418
```

tosSets

The `tosSets` service returns a list of all the ToS set groups that are supported by the appliance. For example, the parameter

```
&type=tosSets
```

produces a result (as CSV) similar to

```
Id,Name, Description, Status
99,ToS Set 1,,W
```

users

The `users` service returns a list of users on the appliance that satisfy the search criteria.

Parameters:

- `searchText`—text string to search for.
- `maxLimit`—maximum number of results to return. This value must be between 1 and 10000, inclusive.

For example, the parameters

```
&type=users&maxLimit=10&searchText=http://www.google.com
```

produce a result (as CSV) similar to

```
Id,Name
110,sguarini
332,hwen
73,dtaylor@w8aas.net
```

vlan

The `vlan` service returns a list of VLANs supported by the appliance. For example, the parameter

```
&type=vlan
```

produces a result (as CSV) similar to

```
Id,Name, Description, Status
99,Test Vlan,,W
```

Specifying a Group

The `metricsForGroup`, `topValues`, `averageValues`, and `timeValues` services all require that you specify one or more groups. The following arguments are used:

- `numGroups`—Number of groups in the drilldown.
- `groupTypei`—Group type of the *i*th group, obtained from the “groups” service.
- `groupArgumenti`—Group argument of the *i*th group, if any. You can use the “groups” service to determine if an argument is needed and the “groupArguments” service to identify the argument.

For example, to drill down to Total Traffic > IP Protocol, you would include the following group arguments:

```
&numGroups=2&groupType1=TotalTraffic&groupType2=IPProtocol&groupArgument2=Ethernet
```

Construction Example (CSV)

Here’s how to construct a URL to drill down into the Default-Internet BG > Local Traffic > TCP group and retrieve time series values for the Throughput (Inbound) and Throughput (Outbound) metrics.

Service:

From the description of supported services, find the service that returns time series values. This gives the first parameter:

```
&type=timeValues
```

Group arguments:

The `timeValues` service requires several arguments. To determine the group arguments (`numGroups`, `groupTypei`, and `groupArgumenti`), begin by using the `groups` service to get information about the groups involved:

```
https://<host>:<port>/webservice/DataServiceServlet?UserName=<user>&Password=<password>&csv=true&type=groups
```

returns

```
NodeId,ParentId,Type,Label,HasArgument,ArgumentType,Argument,HasData
...
98,0,BusinessGroup,Business Group,true,3,,true
...
113,98,LocalTraffic,Local Traffic,false,0,,true
114,113,IPProtocol,IP Protocol,true,13,,true
...
```

This gives the Type strings and shows that the Business Group and IP Protocol groups each have an argument (`ArgumentType` is `true`), but Local Traffic does not. Use the `groupArguments` service to identify the arguments. For Business Group:

```
https://<host>:<port>/webservice/DataServiceServlet?UserName=<user>&Password=<password>&csv=true&type=groupArguments&argumentType=3
```

returns

```
Type,Label,Description,IsPrimitive,Options
3,Business Group,Business Group,false,WebAv-BG-Cli 200.0.02.201-250;WebAv-BG-Cli
160.0.02.101-200;...;Default-Internet;...
```

which shows that the argument for Business Group is the BG name. Pick the desired BG (Default - Internet) from the listed options. Similarly, the information returned by

```
https://<host>:<port>/webservice/DataServiceServlet?UserName=<user>&Password=<password>&csv=true&type=groupArguments&argumentType=13
```

shows that the argument for IP Protocol is the protocol name. From the preceding information, assemble the necessary three levels of group parameters:

```
&numGroups=3&groupType1=BusinessGroup&groupArgument1=Default-Internet&groupType2=LocalTraffic&groupType3=IPProtocol&groupArgument3=TCP
```

Interval arguments:

The `timeValues` service also requires an interval. Use a time utility to convert the desired start and end date/time to UNIX seconds. This gives parameters such as

```
&start=1291658040&end=1291661640
```

Metric argument:

Use the `metrics` service to get the IDs for the Throughput (Inbound) and Throughput (Outbound) metrics. (Use the same approach as when getting groups, but change `&type=groups` to `&type=metrics`.) This gives the parameter

```
&metrics=TPI,TPO
```

Granularity argument:

Finally, use the `granularities` service to discover the possible values and select one for the parameter. For example:

```
&granularity=60
```

The complete URL:

Putting these parameters together gives

```
https://<host>:<port>/webservice/DataServiceServlet?UserName=<user>&Password=<password>&csv=true&type=timeValues&numGroups=3&groupType1=BusinessGroup&groupArgument1=Default-Internet&groupType2=LocalTraffic&groupType3=IPProtocol&groupArgument3=TCP&start=1291658040&end=1291661640&metrics=TPI,TPO&granularity=60
```

Accessing this URL produces the desired results, something similar to this:

```
Group,Time (seconds),Throughput (Inbound) [kb/sec],Throughput (Outbound) [kb/sec]
Default-Internet>Local Traffic>TCP,1291658040,5927.266666666667,5927.266666666667
Default-Internet>Local Traffic>TCP,1291658100,5481.866666666667,5481.866666666667
Default-Internet>Local Traffic>TCP,1291658160,6219.96,6219.96
...
```

Construction Example (XML)

For most services, the URL for XML results is the same as for CSV results, except that the `&csv=` parameter is omitted or set to `false`. However, requesting XML results for services that require group information is a little more complicated. These services are

- `averageValues`
- `metricsForGroup`
- `timeValues`
- `topValues`

For these services, you must post an XML document containing the necessary parameters to the service URL. For example, to request XML results for the drilldown used in “Construction Example (CSV)”, you would post a document containing this text

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <group argument="TCP" argumentType="13" hasArgument="true"
    hasData="true" label="IP Protocol" type="IPProtocol">
    <parent>
      <group argument="" argumentType="0" hasArgument="false"
        hasData="true" label="Local Traffic" type="LocalTraffic">
        <parent>
          <group argument="Default-Internet" argumentType="3"
            hasArgument="true" hasData="true"
            label="Business Group" type="BusinessGroup">
          </group>
        </parent>
      </group>
    </parent>
  </group>
</configuration>
```



```

        </group>
      </parent>
    </group>
    <interval end="1291661640" start="1291658040"/>
    <queryMetrics>
      <metric id="TPI" label="Throughput (Inbound)"/>
      <metric id="TPO" label="Throughput (Outbound)"/>
    </queryMetrics>
    <granularity value="60"/>
  </configuration>

```

to the URL

`https://<host>:<port>/webservice/DataServiceServlet?UserName=<user>&Password=<password>&type=time`
Values

In response, you would get a result similar to this:

```

<?xml version="1.0" encoding="UTF-8"?>
<result>
  <timeSeriesValue>
    <group argument="TCP" argumentType="13" hasArgument="true"
      hasData="true" label="IP Protocol" type="IPProtocol">
      <parent>
        <group argument="" argumentType="0" hasArgument="false"
          hasData="true" label="Local Traffic" type="LocalTraffic">
          <parent>
            <group argument="Default-Internet" argumentType="3"
              hasArgument="true" hasData="true"
              label="Business Group" type="BusinessGroup">
            </group>
          </parent>
        </group>
      </parent>
    </group>
    <interval end="1291661640" start="1291658040"/>
    <granularity value="60"/>
    <metricValue unit="kb/sec">
      <metric id="TPI" label="Throughput (Inbound)"/>
      <interval end="1291661640" start="1291658040"/>
      <values>5927.266666666667,5481.866666666667,6219.96,...</values>
    </metricValue>
    <metricValue unit="kb/sec">
      <metric id="TPO" label="Throughput (Outbound)"/>
      <interval end="1291661640" start="1291658040"/>
      <values>5927.266666666667,5481.866666666667,6219.96,...</values>
    </metricValue>
  </timeSeriesValue>
</result>

```

XML Request Templates

The following templates show the keywords used by each of the four services that require an XML document. Note that

- additional pairs of <parent><group> keywords can be nested as needed to specify the desired drill-down.
- the <queryMetrics> section can contain any number of <metric> keywords.
- values for keyword arguments can be determined using the techniques described in “Construction Example (CSV)”.

averageValues Service

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <group argument= argumentType= hasArgument= hasData= label= type= >
    <parent>
      <group argument= argumentType= hasArgument= hasData= label= type= >
        </group>
      </parent>
    </group>
    <interval end= start= />
    <queryMetrics>
      <metric id= label= />
    </queryMetrics>
  </configuration>
```

metricsForGroup Service

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <group argument= argumentType= hasArgument= hasData= label= type= >
    <parent>
      <group argument= argumentType= hasArgument= hasData= label= type= >
        </group>
      </parent>
    </group>
  </configuration>
```

timeValues Service

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <group argument= argumentType= hasArgument= hasData= label= type= >
    <parent>
      <group argument= argumentType= hasArgument= hasData= label= type= >
        </group>
      </parent>
    </group>
    <interval end= start= />
    <queryMetrics>
      <metric id= label= />
    </queryMetrics>
    <granularity value= />
  </configuration>
```

topValues Service

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <group argument= argumentType= hasArgument= hasData= label= type= >
    <parent>
      <group argument= argumentType= hasArgument= hasData= label= type= >
        </group>
      </parent>
    </group>
    <interval end= start= />
    <queryMetrics>
      <metric id= label= />
    </queryMetrics>
    <topMetric>
      <metric id= label= />
    </topMetric>
  </configuration>
```

```
    </topMetric>  
    <topCount value= />  
</configuration>
```