

# Accessing Data Via the NetInsight API



---

## About the NetInsight API

The Unica NetInsight API lets you retrieve data from Unica NetInsight reports for use in web pages, databases, applications, or other data-capable vehicles. For example, on your company's web site you may want to display a list of the day's most watched videos, hot products and search terms, or the most requested support articles among different user groups. Or you may want to write campaign-based visit information to a database for use with your third-party marketing software. In cases where you cannot access the Unica NetInsight database directly, the API is the best avenue for retrieving Unica NetInsight data programmatically. NetInsight's implementation conforms to the [industry standard XML-RPC specification](#).

The number of API requests you can make and the number of report rows you can transfer is determined by the terms of your NetInsight license.

## About the GetReportData method

In most instances the only Unica NetInsight API call method you will use is GetReportData. GetReportData returns data from a specified report, subject to any filters you specify in the request. The request must include the report key of the targeted report. As desired, you can include date and row filters as well as any of the dimension filters described in this document. Performance will be faster if your requests target data for which you have pre-generated reports.

GetReportData requests use the basic structure shown here:

```
Example: GetPageData request

<?xml version="1.0" encoding="UTF-8"?>
<methodCall>
  <methodName>GetReportData</methodName>
  <params>
    <param>
      <value>some_report_key</value>
    </param>
    <param>
      <value>
        <struct>
          <member>
            <name>timeperiod</name>
```

```

        <value>201003</value>
      </member>
    <member>
      <name>show</name>
      <value>50</value>
    </member>
  </struct>
</value>
</param>
</params>
</methodCall>

```

### Required parameters for GetReportData requests

While you will probably want to include a number of filter parameters in your GetReportData requests, there are only two parameters that are absolutely required: the method name and the report key

- **methodName:** This must be the first parameter within **<methodCall>**. Its value must be **GetReportData**.
- **The report key:** The report key identifies the NetInsight report targeted by the request. It must be the first **<param>** value within the **<params>** element.

A complete list of report keys for the pre-defined reports that ship with NetInsight appears later in this document. You can also lookup reports keys from within the NetInsight UI: In the Custom Report Wizard on the **Custom** tab, view the list of available reports. Each report's key is shown in parentheses alongside its title.

*Example: GetPageData request with required parameters*

```

<?xml version="1.0" encoding="UTF-8"?>
<methodCall>
  <methodName>GetReportData</methodName>
  <params>
    <param>
      <value>some_report_key</value>
    </param>
  </params>
</methodCall>

```

### Optional parameters for GetReportData requests

While you will probably want to include a number of filter parameters in your GetReportData requests, there are only two parameters that are absolutely required: the method name and the report key

- **TimePeriod:** This is an encoded string that specifies the time period to report on. It can be specified as year and month (YYYYMM) or as year only (YYYY). If no value is specified (and no dimension calendar filter is included in the request), the default is to return all rows. If TimePeriod is used in conjunction with a dimension date filter, the dimension date filter takes precedence.

- **StartRow:** This specifies the report row to begin reporting from. If unspecified, it defaults to the first row (1) in the report. In most cases the first row is where you want to begin, but you might use this if you want to list, for example, the 11th through the 20th most popular pages on your site.
- **Show:** From the start row, the number of rows you want to report on. If unspecified, defaults to 100.

The example shown here returns data from March 2010, ten rows starting at row eleven.

*Example: GetPageData request with required and optional parameters*

```
<?xml version="1.0" encoding="UTF-8"?>
<methodCall>
  <methodName>GetReportData</methodName>
  <params>
    <param>
      <value>some_report_key</value>
    </param>
    <param>
      <value>
        <struct>
          <member>
            <name>timeperiod</name>
            <value>201003</value>
          </member>
          <member>
            <name>startrow</name>
            <value>10</value>
          </member>
          <member>
            <name>show</name>
            <value>10</value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

## Using dimension filters in GetReportData calls

As desired, you can filter your GetReportData requests using any dimension filter available to you in Unica NetInsight. This section describes how to use these filters in your requests.

### Basic syntax

In GetReportData requests, individual report dimension filters are defined using **filterinstance** elements. These must themselves be wrapped in a **filterinstances** element. For example:

*Example: filterinstance and filterinstances elements*

```
<member>
  <filterinstances>
    <filterinstance ...attributes...>value</filterinstance>
    <filterinstance ...attributes...>value</filterinstance>
  </filterinstances>
</member>
```

Each **filterinstance** element requires a **type** attribute. The **type** attribute identifies the dimension the filter represents; therefore, the value of the filterinstance type must match the underlying name of the relevant dimension (a complete list of out-of-box NetInsight filters and their specifications appears later in this documents).

Individual filters within a **filterinstances** element are logically ORed together. For example, this filter finds records where the browser is either Chrome or Safari.

*Example: filterinstance types*

```
<filterinstances>
  <filterinstance type="browser"
comparison="equals">chrome</filterinstance>
  <filterinstance type="browser"
comparison="equals">safari</filterinstance>
</filterinstances>
```

### Including dimension filters in the XML remote procedure call

Your **filterinstances** element must be wrapped in a **member** element as shown here. The **member** element's name must be "filters" and within its value element the **filterinstances** element must be wrapped in a CDATA section.

*Example: filters must be wrapped in a member element and CDATA*

```
<member>
  <name>filters</name>
  <value><![CDATA[
    <filterinstances>
      <filterinstance type="date"
comparison="equals">today</filterinstance>
    </filterinstances>
  ]]></value>
</member>
```

### Dimension filter formats

The basic categories of dimension filters include string, numeric, date, and path. Within these categories there are a number of variations. NetInsight also supports filter groups--collections of individual filters that can be applied with a single call.

### Working with string filters

Each string filter must contain a comparison operator and a string value to match on. Valid comparison operators for strings are:

- equals: Finds records that match the specified value.
- contains: Finds records that contain the specified value.
- starts\_with: Finds records that start with the specified value.
- ends\_with: Finds records that end with the specified value.
- match\_pattern: Finds records that match the specified pattern. (**Note:** Pattern matching uses the pattern-matching syntax supported by your Unica NetInsight database. Consult your Unica NetInsight database administrator or technical account manager if you are not sure what database your NetInsight installation uses.)
- match\_regexp: Finds records that match the regular expression. (**Note:** Regular expression matching is not supported with all databases. Consult your Unica NetInsight database administrator or technical account manager to determine if your NetInsight database supports regular expression matching.)

For example, this string filter finds records where the entry page is equal to "index.php":

```
<filterinstance type="entrypage"
comparison="equals">index.php</filterinstance>
```

### String filter variations

Variations on string filters include:

The **stringexists** filter

The **stringexists** filter type supports one additional comparison operator: **exists**. Otherwise it is identical to regular **string** filters. The **stringexists** comparison operator finds records that contain any instance of the filter's target object. You do not specify a string to match. For example, this filter finds visits that contain one or more keyword entries:

```
<filterinstance type="keywords" comparison="exists"/>
```

Composite string filters

A composite filter is a string filter that contains two or more comparison strings. Each string is wrapped in a value element and the value elements are logically ANDed together. Use them when you want to filter on records that must meet multiple criteria. For example, this filter finds visits where the entry page contains "/clearance/" and ends with ".asp":

```
<filterinstance type="entrypage"> <values> <value
comparison="contains"/>/clearance/</value> <value
comparison="ends_with"/>.asp</value> </values>
</filterinstance>
```

### Working with numeric filters

Basic numeric filters contain a single numeric value.

```
<filterinstance type="isimpression">1</filterinstance>
```

## Numeric filter variations

Variations on numeric filters include:

### List filters

List filters are numeric filters that contain two or more values. As with composite string filters, each value in the list is stored as a separate sub-element within an **values** element. List filters are typically used with standard pick lists, such as days of the week or months of the year but are ORed rather than ANDed.

```
<filterinstance type="dow"> <values> <value>1</value>
<value>2</value> <value>5</value> </values>
</filterinstance>
```

### Range filters

Range filters define a numeric range. They consist of a **start** and **end** element defined within a **range** element. For example, this range filter finds visits in which the visitor viewed between one and four pages:

```
<filterinstance type="pagesviewed"> <range> <start>1</start>
<end>4</end> </range> </filterinstance>
```

### hour filter

The Time Of Day (**hour**) filter is a list filter based on a 24 hour clock. The hour from 12 a.m. to 1 a.m. carries a value of zero; 1 a.m. to 2 a.m. equals 1; and so on up to 23. For example, this filter returns records from 12 a.m. to 2 a.m.:

```
<filterinstance type="hour"> <values> <value>0</value>
<value>1</value> </values> </filterinstance>
```

### The **status** filters

The **status** filter type is a special numeric filter type used to filter errors based on their web server error code. For example, this filter returns 404 (Page Not Found) errors:

```
<filterinstance type="status">404</filterinstance>
```

## Working with date filters

Date filters can be specified in several different ways. All require a comparison attribute whose value must be either "equals" or "between." If the comparison is specified as "equals," the value of the filter can be entered as a date (YYYY-MM-DD) or by the keywords "yesterday" or "today." For example:

```
<filterinstance type="date" comparison="equals">2010-04-01</filterinstance>
<filterinstance type="date" comparison="equals">yesterday</filterinstance>
<filterinstance type="date" comparison="equals">today</filterinstance>
```

Date filters that use the "between" comparison can be defined as either a **range** or a **duration**. Date ranges include **start** and **end** sub-elements. Their values must be specified in YYYY-MM-DD format. For example:

```
<filterinstance type="date" comparison="between">
  <range>
    <start>2010-01-01</start>
    <end>2010-02-02</end>
  </range>
</filterinstance>
```


Date durations consist of a **duration** sub-element contained within a **values** element. A duration has two attributes--**type** and **direction**--both of which are required.

- The **type** attribute determines a duration's unit of time. Valid values are "days," "weeks," "months," "quarters," and "years."
- The **direction** attribute determines whether or not the filter includes the current time unit specified in **type**. Valid values are "current" (the current time unit *is* included) and "previous" (the current unit *is not* included). NetInsight uses the calendar definition of time units. For example, "previous week" is defined as the previous Sunday through Saturday, not the previous seven days. "Current year" is defined as the days since January 1, not the last 365 days.

Finally, a duration's value is a number that defines how many of the chosen time units make up the duration. For example, the first duration below specifies the five days previous to (but not including) today. The second duration specifies the previous and current month.

```
<filterinstance type="date" comparison="between">
  <values>
    duration type="days" direction="previous">5</duration>
  </values>
</filterinstance>

<filterinstance type="date" comparison="between">
  <values>
    <duration type="months" direction="current">2</duration>
  </values>
</filterinstance>
```

 If used in conjunction with the TimePeriod parameter, dimension date filters supercede TimePeriod.

## Working with path filters

Path filters limit data based on the path visitors followed through your web site. They are list-type filters, meaning you can specify multiple paths in a single filter.

### Attributes for path filters

Path filters require a **comparison** attribute. Optionally, you can also include the **maxpathlen** parameter to specify the maximum path length to return. Zero is equal to no limit (and is the default if the attribute is unspecified) .

The operators for the **comparison** attribute path filters differ from those used for strings. Valid comparison operators for paths are:

- **match\_exact**: Finds visits whose navigation sequence matches the path exactly.

- `match_within`: Finds visits that contain the path in their navigation sequence.
- `match_start`: Finds visits whose starting page matches the first page of the path (and whose subsequent pages match the rest of the given path).
- `match_end`: Finds visits whose ending page matches the last page of the path (and whose previous pages match the rest of the given path)

```
<filterinstance type="path" comparison="match_start" maxpathlen="4">
  <values>
    <value>/support/NTR/faq.html</value>
    <value>/products/NTR/</value>
  </values>
</filterinstance>
```

### Working with filter groups

A filter group is a filter comprised of two or more individual filters grouped together and saved under a single name in NetInsight profile administration. For example, the sample filter group "\_Converted" that ships with NetInsight consists of two page filters that work together to determine if a sale transaction has been completed. The individual filters that comprise that group can be of different types. Functionally they are logically ANDed together. You reference a filter group by specifying its type as **group** and its value as the group's underlying NetInsight name.

```
<filterinstance type="group">_Converted</filterinstance>
```

### Using the negate attribute in a filter

Optionally, if you want to filter on records that *do not* match a specified value, you can add a **negate** attribute to a filter and set its value to true. For example, this filter finds visits where the entry page does not equal "index.php":

```
<filterinstance type="entrypage" comparison="equals"
negate="true">index.php</filterinstance>
```

The **negate** attribute can be used with string, numeric, and path filters. It cannot be used with dates. It can be used with individual items within a list. For example:

```
<filterinstance type="pagewithbr">
  <values>
    <value comparison="equals" negate="true">index.cfm</value>
    <value comparison="equals">br_page</value>
  </values>
</filterinstance>
```



## Filter reference

The following table lists the dimension filters that ship with NetInsight and gives information on their use. If you are an Enterprise user with access to files on the NetInsight server, you can experiment with filters and test syntax by applying and saving filters to a custom report and then examining that report's XML file (for example, *NetInsight\_root/data/profile\_name/report001.xml*).

| Filter Name in UI     | Dimension Name | Filter Type       | Example   | List Values  |
|-----------------------|----------------|-------------------|---|--|
| 3-digit Zip Code      | zip3           | stringexists      | <pre>&lt;filterinstance type="zip3" comparison="equals"&gt;019&lt;/filterinstance&gt;</pre>   |  |
| 5-digit Zip Code      | zip5           | stringexists      | <pre>&lt;filterinstance type="zip5" comparison="starts_with"&gt;02152&lt;/filterinstance&gt;</pre>  |  |
| Area Code             | aconly         | stringexists      | <pre>&lt;filterinstance type="aconly" comparison="equals"&gt;781&lt;/filterinstance&gt;</pre>   |  |
| Area Code with County | area-code      | composite         | <pre>&lt;filterinstance type="areacode"&gt; &lt;values&gt; &lt;value comparison="equals"&gt;781&lt;/value&gt; &lt;value comparison="equals"&gt;us&lt;/value&gt; &lt;/values&gt; &lt;/filterinstance&gt;</pre> | <p>First value = area code, second value = country</p> <p>The value for country is matched against valid country codes in the Geo data conduit database.</p> |
| Banner Ad             | bannerad       | bannerad (string) | <pre>&lt;filterinstance type="bannerad" comparison="contains"&gt;/ads/images/ADS/22/49/&lt;/filterinstance&gt;</pre>  |  |
| Browser               | browser        | stringexists      | <pre>&lt;filterinstance type="browser" comparison="equals"&gt;mozilla&lt;/filterinstance&gt;</pre>  |  |
| Browser Break-down    | browserbr      | string            | <pre>&lt;filterinstance type="browserbr" comparison="ends_with"&gt;Firefox/2.0.0.7&lt;/filterinstance&gt;</pre>   |  |

| Filter Name in UI           | Dimension Name        | Filter Type         | Example   | List Values   |
|-----------------------------|-----------------------|---------------------|---|---|
| Campaign                    | campaign              | stringexists        | <pre>&lt;filterinstance type="campaign" comparison="equals"&gt;I100227A&lt;/filterinstance&gt;</pre>  |   |
| Campaign Channel            | campaign-channel      | stringexists        | <pre>&lt;filterinstance type="campaignchannel" comparison="equals"&gt;nwslttr&lt;/filterinstance&gt;</pre>  |   |
| Campaign Channel Type       | campaign-channel-type | stringexists        | <pre>&lt;filterinstance type="campaignchanneltype" comparison="equals"&gt;drmail&lt;/filterinstance&gt;</pre>   |   |
| Campaign Segment            | campaign-segment      | stringexists        | <pre>&lt;filterinstance type="campaignsegment" comparison="equals"&gt;FF1&lt;/filterinstance&gt;</pre>  |   |
| City                        | cityonly              | city (stringexists) | <pre>&lt;filterinstance type="cityonly" equals=" Tokyo"/&gt;</pre>  | The value for city must be a valid code from the Geo data conduit database.   |
| City with State and Country | city                  | composite           | <pre>&lt;filterinstance type="city"&gt; &lt;values&gt; &lt;value comparison="equals"&gt;Springfield&lt;/value&gt; &lt;value comparison="equals"&gt;MA&lt;/value&gt; &lt;value comparison="equals"&gt;us&lt;/value&gt; &lt;/values&gt; &lt;/filterinstance&gt;</pre> | <p>First value = city, second value = state, third value = country</p> <p>The value for city, state, and country are matched against valid city, state, and country codes in the Geo data conduit database.</p> |

| Filter Name in UI | Dimension Name | Filter Type              | Example   | List Values  |
|-------------------|----------------|--------------------------|---|--|
| Connect Type      | conn-type      | conn-type (stringexists) | <pre>&lt;filterinstance type="conntype" comparison="equals"&gt;SATELLITE&lt;/filterinstance&gt;</pre>       | The value for conntype is matched against valid country codes in the Geo data conduit database.    |
| Content Group     | content-group  | stringexists             | <pre>&lt;filterinstance type="contentgroup" comparison="exists"/&gt;</pre>                                  |  |
| Continent         | continent      | continent (stringexists) | <pre>&lt;filterinstance type="continent" comparison="equals"&gt;NA&lt;/filterinstance&gt;</pre>             | The value for continent is matched against valid continent codes in the Geo data conduit database. |
| Cookie            | cookie         | string                   | <pre>&lt;filterinstance type="cookie" comparison="contains"&gt;coffee-time.com&lt;/filterinstance&gt;</pre> |  |
| Country           | country        | country (stringexists)   | <pre>&lt;filterinstance type="country" comparison="equals"&gt;ba&lt;/filterinstance&gt;</pre>               | The value for country is matched against valid country codes in the Geo data conduit database.     |

| Filter Name in UI | Dimension Name | Filter Type | Example   | List Values   |
|-------------------|----------------|-------------|---|---|
| Date              | date           | date        | <pre>&lt;filterinstance type="date" comparison="equals"&gt;yesterday&lt;/filterinstance&gt; &lt;filterinstance type="date" comparison="between"&gt; &lt;values&gt; &lt;duration direction="current" type="weeks"&gt;2&lt;/duration&gt; &lt;/values&gt; &lt;/filterinstance&gt; &lt;filterinstance type="date" comparison="between"&gt; &lt;values&gt; &lt;duration direction="previous" type="quarters"&gt;2&lt;/duration&gt; &lt;/values&gt; &lt;/filterinstance&gt; &lt;filterinstance type="date" comparison="between"&gt; &lt;values&gt; &lt;duration direction="current" type="years"&gt;1&lt;/duration&gt; &lt;/values&gt; &lt;/filterinstance&gt; &lt;filterinstance type="date" comparison="between"&gt; &lt;range&gt; &lt;start&gt;2010-01-01&lt;/start&gt; &lt;end&gt;2010-02-27&lt;/end&gt; &lt;/range&gt; &lt;/filterinstance&gt;</pre> | <p>For <b>comparison</b>, valid values are "equals" and "between."</p> <p>For <b>comparison="equals"</b>:<br/>Valid values are "today," "yesterday," and YYYY-MM-DD.</p> <p>For <b>comparison="between"</b>:</p> <ol style="list-style-type: none"> <li>1. For <b>direction</b>, valid values are "previous" and "current."</li> <li>2. For <b>type</b>, valid values are "days," "weeks," "months," "quarters," and "years."</li> <li>3. Start and end date values must be specified as YYYY-MM-DD.</li> </ol> |
| Day of the Week   | dow            | list        | <pre>&lt;filterinstance type="dow"&gt; &lt;values&gt; &lt;value&gt;1&lt;/value&gt; &lt;value&gt;7&lt;/value&gt; &lt;/values&gt; &lt;/filterinstance&gt;</pre>   | Days of the week defined as numbers 1 through 7 where Sunday = 1, Monday = 2, and so on.  |
| Department        | department     | string      | <pre>&lt;filterinstance type="department" comparison="equals"&gt;eu_sales&lt;/filterinstance&gt;</pre>  |   |
| Directory         | directory      | string      | <pre>&lt;filterinstance type="directory" comparison="equals"&gt;/support/kb/&lt;/filterinstance&gt;</pre>   |   |

| Filter Name in UI          | Dimension Name | Filter Type     | Example   | List Values                                  |
|----------------------------|----------------|-----------------|---|--|
| Domain                     | domain         | domain (string) | <pre>&lt;filterinstance type="domain" comparison="contains"&gt;amazon.co.uk&lt;/filterinstance&gt;</pre>  |  |
| Entry Page                 | entrypage      | string          | <pre>&lt;filterinstance type="entrypage" comparison="equals"&gt;index.htm&lt;/filterinstance&gt;</pre>  |  |
| Entry Page Break-down      | entrypagebr    | string          | <pre>&lt;filterinstance type="entrypagebr" comparison="starts_with"&gt;/cgi-bin/demo&lt;/filterinstance&gt;</pre>   |  |
| Entry Page with Break-down | entrypagewbr   | composite       | <pre>&lt;filterinstance type="entrypagewbr"&gt;&lt;values&gt;&lt;value comparison="equals"&gt;/&lt;/value&gt;&lt;value comparison="contains"&gt;?type=new_registration&lt;/value&gt;&lt;/values&gt;&lt;/filterinstance&gt;</pre>      | First value = page, second value = breakdown |
| Error Message              | status         | status          | <pre>&lt;filterinstance type="status"&gt;404&lt;/filterinstance&gt;</pre>   |  |
| Exit Page                  | exitpage       |                 | <pre>&lt;filterinstance type="exitpage" comparison="contains"&gt;robots.txt&lt;/filterinstance&gt;</pre>  |  |
| Exit page Break-down       | exitpagebr     | string          | <pre>&lt;filterinstance type="exitpagebr" comparison="contains"&gt;wp-cron.php?check=&lt;/filterinstance&gt;</pre>  |  |
| Exit Page with Break-down  | exitpagewbr    | composite       | <pre>&lt;filterinstance type="exitpagewbr"&gt;&lt;values&gt;&lt;value comparison="equals"&gt;/&lt;/value&gt;&lt;value comparison="equals"&gt;symptomsearch?addterm=headache&lt;/value&gt;&lt;/values&gt;&lt;/filterinstance&gt;</pre> | First value = page, second value = breakdown |

| Filter Name in UI                  | Dimension Name          | Filter Type   | Example  | List Values |
|------------------------------------|-------------------------|---------------|--|-------------|
| Filter Group                       | group                   | group         | <pre>&lt;filterinstance type="group"&gt;_Converted&lt;/filterinstance&gt;</pre>  |             |
| Host                               | host                    | string        | <pre>&lt;filterinstance type="host" comparison="equals"&gt;167.211.140.243&lt;/filterinstance&gt;</pre>  |             |
| Initial Referrer                   | initialreferrer         | string        | <pre>&lt;filterinstance type="initialreferrer" comparison="equals"&gt;Google&lt;/filterinstance&gt;</pre>  |             |
| Initial Referrer Break-down        | initialreferrerbr       | string        | <pre>&lt;filterinstance type="initialreferrerbr" comparison="equals"&gt;http://www.scriptsearch.com/pages/14c3c28.shtml&lt;/filterinstance&gt;</pre> |             |
| Initial Spider Referrer            | initialspiderreferrer   | string        | <pre>&lt;filterinstance type="initialspiderreferrer" comparison="contains"&gt;ShopWiki&lt;/filterinstance&gt;</pre>                                  |             |
| Initial Spider Referrer Break-down | initialspiderreferrerbr | string        | <pre>&lt;filterinstance type="initialspiderreferrerbr" comparison="contains"&gt;SapphireWebCrawler&lt;/filterinstance&gt;</pre>                      |             |
| Key-words                          | key-words               | stringex-ists | <pre>&lt;filterinstance type="keywords" comparison="equals"&gt;buy gourmet coffee beans&lt;/filterinstance&gt;</pre>                                 |             |
| Link                               | linkpage                | string        | <pre>&lt;filterinstance type="linkpage" comparison="contains"&gt;/partner-sites/&lt;/filterinstance&gt;</pre>  |             |
| Local Key-words                    | localkey-words          | stringex-ists | <pre>&lt;filterinstance type="localkeywords" comparison="equals"&gt;arabica&lt;/filterinstance&gt;</pre>   |             |

| Filter Name in UI      | Dimension Name | Filter Type   | Example   | List Values  |
|------------------------|----------------|---------------|---|--|
| Number of Pages Viewed | pages-viewed   | range         | <pre>&lt;filterinstance type="pagesviewed"&gt; &lt;range&gt; &lt;start&gt;1&lt;/start&gt; &lt;end&gt;5&lt;/end&gt; &lt;/range&gt; &lt;/filterinstance&gt;</pre> | Set <end> to 0 to indicate an unlimited upper bound.   |
| Number of Visits       | retentioncode  | list          | <pre>&lt;filterinstance type="retentioncode"&gt; &lt;values&gt; &lt;value&gt;7&lt;/value&gt; &lt;/values&gt; &lt;/filterinstance&gt;</pre>                      | Possible values:<br>1 = 1<br>2 = 2<br>3 = 3<br>4 = 4<br>5-6 = 5<br>7-9 = 6<br>10-14 = 7<br>15-24 = 8<br>25-49 = 9<br>50-99 = 10<br>100+ = 11 |
| Order Number           | order-number   | stringex-ists | <pre>&lt;filterinstance type="ordernumber" comparison="starts_with"&gt;321&lt;/filterinstance&gt;</pre>   |  |
| Organization           | organization   | stringex-ists | <pre>&lt;filterinstance type="organization" comparison="equals"&gt;America Online&lt;/filterinstance&gt;</pre>  |  |
| Page                   | page           | string        | <pre>&lt;filterinstance type="page" comparison="equals"&gt;/links.htm&lt;/filterinstance&gt;</pre>  |  |
| Page Break-down        | pagebr         | string        | <pre>&lt;filterinstance type="pagebr" comparison="contains"&gt;.pdf&lt;/filterinstance&gt;</pre>  |  |

| Filter Name in UI                                    | Dimension Name | Filter Type   | Example   | List Values                                       |
|--|----------------|---------------|---|---|
| Page with Break-down                                 | pagewbr        | composite     | <pre>&lt;filterinstance type="pagewbr"&gt; &lt;values&gt; &lt;value comparison="equals"&gt;home.jsp&lt;/value &gt; &lt;value comparison="equals"&gt;redirect=http:// thebigtuna.net&lt;/value&gt; &lt;/values&gt; &lt;/filterinstance&gt;</pre>                                 | First value = page, second value = breakdown      |
| Paid Key-words                                       | paidkey-words  | stringex-ists | <pre>&lt;filterinstance type="paidkeywords" comparison="starts_with"&gt;sumatra roast&lt;/filterinstance&gt;</pre>  |   |
| Path   | path           | path          | <pre>&lt;filterinstance type="path" comparison="match_within" maxpathlen="0"&gt; &lt;values&gt; &lt;value&gt;/index.php&lt;/value&gt; &lt;value&gt;/blends.php&lt;/value&gt; &lt;value&gt;/products/estate- coffee/&lt;/value&gt; &lt;/values&gt; &lt;/filterinstance&gt;</pre> |   |
| Platform   | platform       | string        | <pre>&lt;filterinstance type="platform" comparison="contains"&gt;Macintosh&lt;/fi lterinstance&gt;</pre>  |   |
| Referrer   | referrer       | string        | <pre>&lt;filterinstance type="referrer" comparison="equals"&gt;dogpile.com&lt;/fi lterinstance&gt;</pre>  |   |
| Referrer Break-down                                  | refer-erbr     | string        | <pre>&lt;filterinstance type="referrerbr" comparison="contains"&gt;?authtype=1&amp;r edirect=%2Fmanageagents%2Easp&lt;/filt erinstance&gt;</pre>  |   |
| Request Completion: (Successful or Aborted) Requests | aborted        | list          | <pre>&lt;filterinstance type="aborted"&gt; &lt;values&gt; &lt;value&gt;0&lt;/value&gt; &lt;/values&gt; &lt;/filterinstance&gt;</pre>  | Possible values:<br>Successful = 0<br>Aborted = 1 |



| Filter Name in UI        | Dimension Name    | Filter Type   | Example   | List Values   |
|--------------------------|-------------------|---------------|---|---|
| Re-requested Page        | errorpage         | string        | <pre>&lt;filterinstance type="errorpage" comparison="equals"&gt;/booth.php&lt;/filterinstance&gt;</pre>                                 |   |
| Retail Action            | action-type       | list          | <pre>&lt;filterinstance type="actiontype"&gt; &lt;values&gt; &lt;value&gt;1&lt;/value&gt; &lt;/values&gt; &lt;/filterinstance&gt;</pre> | Possible values:<br>abandoned products = 1<br>added products = 2<br>purchased products = 3<br>removed products = 4<br>viewed products = 5 |
| Robot-/Spider            | spider            | string        | <pre>&lt;filterinstance type="spider" comparison="equals"&gt;Grub&lt;/filterinstance&gt;</pre>  |   |
| Robot-/Spider Break-down | spiderbr          | string        | <pre>&lt;filterinstance type="spiderbr" comparison="contains"&gt;compatible; Charlotte/1.1&lt;/filterinstance&gt;</pre>                 |   |
| Screen Resolution        | screen-resolution | stringex-ists | <pre>&lt;filterinstance type="screenresolution" comparison="starts_with"&gt;1024&lt;/filterinstance&gt;</pre>                           |   |
| Server                   | server            | stringex-ists | <pre>&lt;filterinstance type="server" comparison="contains"&gt;/emea&lt;/filterinstance&gt;</pre>                                       |   |
| State/Province           | stateonly         | stringex-ists | <pre>&lt;filterinstance type="stateonly" comparison="equals"&gt;Texas&lt;/filterinstance&gt;</pre>                                      | The value for state is matched against valid state codes in the Geo data conduit database.  |

| Filter Name in UI           | Dimension Name | Filter Type  | Example  | List Values  |
|-----------------------------|----------------|--------------|--|--|
| State/Province with Country | state          | composite    | <pre>&lt;filterinstance type="state"&gt; &lt;values&gt; &lt;value comparison="equals"&gt;Texas&lt;/value&gt; &lt;value comparison="equals"&gt;us&lt;/value&gt; &lt;/values&gt; &lt;/filterinstance&gt;</pre> | <p>First value = state, second value = country</p> <p>The values for state and country are matched against valid state and country codes in the Geo data conduit database.</p> |
| Time of Day                 | hour           | list         | <pre>&lt;filterinstance type="hour"&gt; &lt;values&gt; &lt;value&gt;0&lt;/value&gt; &lt;/values&gt; &lt;/filterinstance&gt;</pre>  | <p>Possible values:</p> <p>12 a.m. – 1 a.m. = 0</p> <p>1 a.m. – 2 a.m. = 1</p> <p>2 a.m. – 3 a.m. = 2</p> <p>through...</p> <p>11 p.m. – 12 a.m. = 23</p>                      |
| Time Zone                   | time-zone      | list         | <pre>&lt;filterinstance type="timezone"&gt; &lt;values&gt; &lt;value&gt;-14&lt;/value&gt; &lt;value&gt;0&lt;/value&gt; &lt;value&gt;13&lt;/value&gt; &lt;/values&gt; &lt;/filterinstance&gt;</pre>           | <p>Values are in relation to Greenwich Mean Time. Examples:</p> <p>GMT = 0</p> <p>GMT -5 = -5</p> <p>GMT + 5 = 5</p>   |
| User                        | user           | stringexists | <pre>&lt;filterinstance type="user" comparison="exists"/&gt;&lt;/filterinstance&gt;</pre>  |  |

| Filter Name in UI | Dimension Name | Filter Type | Example  | List Values   |
|-------------------|----------------|-------------|--|---|
| Visit Duration    | visitduration  | list        | <pre>&lt;filterinstance type="visitduration"&gt; &lt;values&gt; &lt;value&gt;1&lt;/value&gt; &lt;value&gt;15&lt;/value&gt; &lt;/values&gt; &lt;/filterinstance&gt;</pre>   | <p>Possible values:</p> <p>Didn't stay = 1</p> <p>Less than 1 minute = 2</p> <p>1-2 minutes = 3</p> <p>2-3 minutes = 4</p> <p>3-4 minutes = 5</p> <p>4-5 minutes = 6</p> <p>5-7 minutes = 7</p> <p>7-10 minutes = 8</p> <p>10-15 minutes = 9</p> <p>15-20 minutes = 10</p> <p>20-30 minutes = 11</p> <p>30-40 minutes = 3</p> <p>40-50 minutes = 3</p> <p>50-60 minutes = 3</p> <p>1-2 hours = 14</p> <p>More than 2 hours = 15</p> |
| Visit Type        | visittype      | list        | <pre>&lt;filterinstance type="visittype"&gt; &lt;values&gt; &lt;value&gt;1&lt;/value&gt; &lt;/values&gt; &lt;/filterinstance&gt;</pre>   | <p>Possible values:</p> <p>New Visits = 1</p> <p>Repeat Visits = 2</p>  |
| Visitor           | visitor        | string      | <pre>&lt;filterinstance type="visitor" comparison="equals"&gt;69.48.38.239:Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.0.04506; MSOffice 12)&lt;/filterinstance&gt;</pre> |   |

| Filter Name in UI | Dimension Name  | Filter Type   | Example   | List Values |
|-------------------|-----------------|---------------|---|-------------|
| Visitor Profile   | visitor-profile | stringex-ists | <pre>&lt;filterinstance type="visitorprofile" comparison="equals"&gt;big_spenders&lt;/f ilterinstance&gt;</pre> |             |

### Pulling it all together...

Following is an example of a complete GetReportData call. It returns the ten most viewed pages by Mac users who visited the site yesterday between noon and 3 p.m. are were referred by Google.

```
<?xml version="1.0" encoding="UTF-8"?>
<methodCall>
  <methodName>GetReportData</methodName>
  <params>
    <param>
      <value>page</value>
    </param>
    <param>
      <value>
        <struct>
          <member>
            <name>show</name>
            <value>10</value>
          </member>
          <member>
            <name>filters</name>
            <value><![CDATA[
              <filterinstances>
                <filterinstance type="date"
comparison="equals">yesterday</filterinstance>
                <filterinstance type="hour">
                  <values>
                    <value>12</value>
                    <value>13</value>
                    <value>14</value>
                  </values>
                </filterinstance>
                <filterinstance type="referrer"
comparison="equals">Google</filterinstance>
                <filterinstance type="platform"
comparison="equals">Macintosh</filterinstance>
              </filterinstances>
            ]]></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

## Other methods

Aside from GetReportData, there are several other methods available that may be of use.

### GetReportList

The GetReportList method returns the list of available reports for the default view of the account used in the API connection. One example for use is if you create an application to monitor your Unica NetInsight installation. The only parameter it requires or accepts is **methodName**, the value for which must be **GetReportList**.

*Example: GetReportList request*

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>GetReportList</methodName>
</methodCall>
```

### GetProfiles

The GetProfiles method returns the list of profiles defined in a given Unica NetInsight installation. Like the GetReportList method, a possible use for this method is if you create an application to monitor your Unica NetInsight installation. The only parameter it requires or accepts is **methodName**, the value for which must be **GetProfiles**.

*Example: GetProfiles request*

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>GetProfiles</methodName>
</methodCall>
```

- ⚡ Unlike the GetReportData and GetReportList methods whose requests must be fired from within a profile directory (for example, <http://myserver.mydomain.com/scripts/netinsight/myprofile/ntcgi.exe>), GetProfile must be fired from a directory level above any particular profile (for example, <http://myserver.mydomain.com/scripts/netinsight/ntcgi.exe>).

## Making an API call to NetInsight

NetInsight's API supports a number of call interface languages including Perl, Java, Ruby, and .NET. For a range of examples see <http://www.xmlrpc.com/directory/1568/implementations>.


The requesting application calls the connection script file as:

```
./script_file netinsight_cgi_path/profile_name/ntcgi.cgi
xml_request_file
```

For example:

```
netinsightapi.pl
http://netinsight.mydomain.com/scripts/netinsight/myprofile/ntcgi.exe
getreportdata.xml
```

The sample connection script shown here is in Perl. In it, the connection to NetInsight is made using basic authentication. Then an XML request stored in an external file is invoked to retrieve a response from NetInsight.

 See the lab exercise for a step-by-step example of invoking a call.

```
#!/usr/bin/perl -w

use strict;
use LWP;
use HTTP::Request;

my $url = shift;
my $filename = shift;
open( F, "< $filename" ) or die "Error: Can't open $filename : $!\nUsage:
netinsightapi.pl targetURL xmlrequestfile\n";
my $content = join("",<F>);

print "Sent:\n$content\n";
print "Returned:\n";

my $conf_username = "username";
my $conf_password = "password";

my ( $response, $request, $header, $ua, $uname, $password );

$header = HTTP::Headers->new;
$header->authorization_basic( $conf_username, $conf_password );
$header->header('Content-Type' => 'text/xml');
$request = HTTP::Request->new( "POST", $url, $header, $content );
$ua = LWP::UserAgent->new;
$ua->agent(
"Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1;+.NET+CLR+1.1.4322)"
);

$response = $ua->request( $request );
if ( $response->is_success )
{
    print $response->content . "\n";
}
else
{
    print $response->status_line . "\n";
}
print "Done.";
```

## Authenticating the call

Remote requests must provide authentication as required by your NetInsight installation. Requests initially attempt basic authentication at the web server. If web server authentication is disabled, requests next attempt authentication against Unica NetInsight. NetInsight OnDemand installations use basic web server authentication. Those accounts are then synchronized to internal Unica NetInsight accounts.

## API call response examples

### Sample GetReportData response

Here's an example of the data returned in response to a GetReportData call. In the sample call that generated this response, the Page Summary report was requested. In the interest of space only three rows were requested. A date filter was also applied. Note that the actual report data is returned XML-encoded.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodResponse>
  <params>
    <param>
      <value>&lt;?xml version=&quot;1.0&quot; encoding=&quot;iso-8859-
1&quot;?&gt;
&lt;xmlreport title=&quot;Widget Store: Page Summary&quot;
dates=&quot;March 13, 2010&quot;&gt;
  &lt;columns&gt;
    &lt;column name=&quot;Page&quot;
type=&quot;dimension&quot;&gt;Page&lt;/column&gt;
    &lt;column name=&quot;Last Visit&quot; type=&quot;metric&quot;&gt;Last
Visit&lt;/column&gt;
    &lt;column name=&quot;Average Viewing Time&quot;
type=&quot;metric&quot;&gt;Average Viewing Time&lt;/column&gt;
    &lt;column name=&quot;Views&quot; type=&quot;metric&quot;
hasTotals=&quot;true&quot;&gt;Views&lt;/column&gt;
    &lt;column name=&quot;Visits&quot;
type=&quot;metric&quot;&gt;Visits&lt;/column&gt;
  &lt;/columns&gt;
  &lt;rows&gt;
    &lt;row rownum=&quot;1&quot;&gt;
      &lt;cell columnName=&quot;page&quot;
csv=&quot;&amp;quot;/product_detail.php&amp;quot;,&amp;quot;/product_detail.
php&amp;quot;&quot;&gt;/product_detail.php&lt;/cell&gt;
      &lt;cell columnName=&quot;lastvisitdatetime&quot; csv=&quot;3/13/2010
10:33&quot;&gt;March 13, 2010 at 10:33 a.m.&lt;/cell&gt;
      &lt;cell columnName=&quot;averageviewingtime&quot;
csv=&quot;75&quot;&gt;1 minute, 15 seconds&lt;/cell&gt;
      &lt;cell columnName=&quot;views&quot;
percentage=&quot;&amp;quot;0.0&amp;quot;&quot;&gt;34&lt;/cell&gt;
      &lt;cell columnName=&quot;visits&quot;
percentage=&quot;&amp;quot;0.0&amp;quot;&quot;&gt;19&lt;/cell&gt;
    &lt;/row&gt;
    &lt;row rownum=&quot;2&quot;&gt;
```

```

    <cell columnname="page">
    csv="&quot;/support_home.php&quot;,"&quot;/support_home.php&
    amp;quot;"/support_home.php"</cell>
    <cell columnname="lastvisitdatetime" csv="3/13/2010
    9:49">March 13, 2010 at 9:49 a.m.</cell>
    <cell columnname="averageviewingtime">
    csv="10">10 seconds</cell>
    <cell columnname="views">
    percentage="&quot;0.0%&quot;">16</cell>
    <cell columnname="visits">
    percentage="&quot;0.0%&quot;">13</cell>
    </row>
    <row rownum="3">
    <cell columnname="page">
    csv="&quot;/widget_accessories.php&quot;,"&quot;/widget_acce
    ssories.php&quot;"/widget_accessories.php"</cell>
    <cell columnname="lastvisitdatetime" csv="3/13/2010
    10:13">March 13, 2010 at 10:13 a.m.</cell>
    <cell columnname="averageviewingtime">
    csv="85">1 minute, 25 seconds</cell>
    <cell columnname="views">
    percentage="&quot;0.0%&quot;">11</cell>
    <cell columnname="visits">
    percentage="&quot;0.0%&quot;">8</cell>
    </row>
    </rows>
    <totals>
    <pagetotals>
    <total columnname="lastvisitdatetime">
    value="1.000000">-</total>
    <total columnname="averageviewingtime">
    value="0.000000">-</total>
    <total columnname="views">
    value="79.000000">79 (0.0%)</total>
    <total columnname="visits">
    value="0.000000">-</total>
    </pagetotals>
    <reporttotals>
    <total columnname="lastvisitdatetime">
    value="1.000000">-</total>
    <total columnname="averageviewingtime">
    value="0.000000">-</total>
    <total columnname="views">
    value="136.000000">136 (0.0%)</total>
    <total columnname="visits">
    value="0.000000">-</total>
    </reporttotals>
    <timeperiodtotals>
    <total columnname="lastvisitdatetime">
    value="0.000000">-</total>
    <total columnname="averageviewingtime">
    value="0.000000">-</total>
    <total columnname="views">
    value="839886.000000">839,886 (100.0%)</total>
    <total columnname="visits">
    value="258902.000000">-</total>
    </timeperiodtotals>
    </totals>

```



```
</xmlreport>
</value>
  </param>
</params>
</methodResponse>
```

Decoding the XML provides a clearer look at the data. The **value** element contains the exact XML that would be created by exporting a report in XML format from the Unica NetInsight interface:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodResponse>
  <params>
    <param>
      <value><?xml version="1.0" encoding="iso-8859-1"?>
<xmlreport title="The Widget Store: Page Summary" dates="March 13, 2010">
  <columns>
    <column name="Page" type="dimension">Page</column>
    <column name="Last Visit" type="metric">Last Visit</column>
    <column name="Average Viewing Time" type="metric">Average Viewing
Time</column>
    <column name="Views" type="metric" hastotals="true">Views</column>
    <column name="Visits" type="metric">Visits</column>
  </columns>
  <rows>
    <row rownum="1">
      <cell columnname="page"
csv=""/product_detail.php","/product_detail.php"/>/product_detail.php</cell>

      <cell columnname="lastvisitdatetime" csv="3/13/2010 10:33">March 13,
2010 at 10:33 a.m.</cell>
      <cell columnname="averageviewingtime" csv="75">1 minute, 15
seconds</cell>
      <cell columnname="views" percentage="0.0%">34</cell>
      <cell columnname="visits" percentage="0.0%">19</cell>
    </row>
    <row rownum="2">
      <cell columnname="page"
csv=""/support_home.php","/support_home.php"/>/support_home.php</cell>
      <cell columnname="lastvisitdatetime" csv="3/13/2010 9:49">March 13,
2010 at 9:49 a.m.</cell>
      <cell columnname="averageviewingtime" csv="10">10 seconds</cell>
      <cell columnname="views" percentage="0.0%">16</cell>
      <cell columnname="visits" percentage="0.0%">13</cell>
    </row>
    <row rownum="3">
      <cell columnname="page"
csv=""/widget_accessories.php.php","/widget_accessories.php.php"/>/widget_ac
cessories.php.php</cell>
      <cell columnname="lastvisitdatetime" csv="3/13/2010 10:13">March 13,
2010 at 10:13 a.m.</cell>
      <cell columnname="averageviewingtime" csv="85">1 minute, 25
seconds</cell>
      <cell columnname="views" percentage="0.0%">11</cell>
      <cell columnname="visits" percentage="0.0%">8</cell>
    </row>
  </rows>
```

```

<totals>
  <pagetotals>
    <total columnname="lastvisitdatetime" value="1.000000">-</total>
    <total columnname="averageviewingtime" value="0.000000">-</total>
    <total columnname="views" value="79.000000">79 (0.0%)</total>
    <total columnname="visits" value="0.000000">-</total>
  </pagetotals>
  <reporttotals>
    <total columnname="lastvisitdatetime" value="1.000000">-</total>
    <total columnname="averageviewingtime" value="0.000000">-</total>
    <total columnname="views" value="136.000000">136 (0.0%)</total>
    <total columnname="visits" value="0.000000">-</total>
  </reporttotals>
  <timeperiodtotals>
    <total columnname="lastvisitdatetime" value="0.000000">-</total>
    <total columnname="averageviewingtime" value="0.000000">-</total>
    <total columnname="views" value="839886.000000">839,886
(100.0%)</total>
    <total columnname="visits" value="258902.000000">-</total>
  </timeperiodtotals>
</totals>
</xmlreport>
</value>
</param>
</params>
</methodResponse>

```

## Sample GetProfiles response

Here's an example of the data returned in response to a GetProfiles call. In the response:

- name = the underlying NetInsight short profile name.
- title = the title of the profile as it appears in the NetInsight interface
- type = the profile type. This will always be "web."

```

<?xml version="1.0" encoding="iso-8859-1"?>
<methodResponse>
  <params>
    <param>
      <value>
        <array>
          <data>
            <value>
              <struct>
                <member>
                  <name>name</name>
                  <value>widgetstore</value>
                </member>
                <member>
                  <name>title</name>
                  <value>The Widget Store</value>
                </member>
                <member>
                  <name>type</name>
                  <value>web</value>
                </member>
              </struct>
            </value>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodResponse>

```

```

    </struct>
  </value>
  <value>
    <struct>
      <member>
        <name>name</name>
        <value>widgetsrus</value>
      </member>
      <member>
        <name>title</name>
        <value>Widgets R Us</value>
      </member>
      <member>
        <name>type</name>
        <value>web</value>
      </member>
    </struct>
  </value>
</data>
</array>
</value>
</param>
</params>
</methodResponse>

```

### Sample GetReportList response

Here's an example of the data returned in response to a GetReportList call. (For reasons of space, the example is truncated.) Report folders are listed in an array (only the Dashboard folder is shown here), and within each folder the individual reports are also listed in an array. The first **member** element for each folder contains its label--that is, the folder's name. The second **member** element, always named "reports," contains the report array.

For individual reports:

- label = the title of the report as it appears in the NetInsight interface
- ID = the report's NetInsight report key.

```

<?xml version="1.0" encoding="iso-8859-1"?>
<methodResponse>
  <params>
    <param>
      <value>
        <array>
          <data>
            <value>
              <struct>
                <member>
                  <name>label</name>
                  <value>Dashboards</value>
                </member>
                <member>
                  <name>reports</name>
                  <value>
                    <array>
                      <data>
                        <value>

```

```
<struct>
  <member>
    <name>label</name>
    <value>Executive</value>
  </member>
  <member>
    <name>ID</name>
    <value>execdash</value>
  </member>
</struct>
</value>
<value>
  <struct>
    <member>
      <name>label</name>
      <value>Marketing</value>
    </member>
    <member>
      <name>ID</name>
      <value>marketdash</value>
    </member>
  </struct>
</value>
<value>
  <struct>
    <member>
      <name>label</name>
      <value>Content</value>
    </member>
    <member>
      <name>ID</name>
      <value>contentdash</value>
    </member>
  </struct>
</value>
<value>
  <struct>
    <member>
      <name>label</name>
      <value>Visitor</value>
    </member>
    <member>
      <name>ID</name>
      <value>visdash</value>
    </member>
  </struct>
</value>
<value>
  <struct>
    <member>
      <name>label</name>
      <value>Traffic</value>
    </member>
    <member>
      <name>ID</name>
      <value>trafficdash</value>
    </member>
  </struct>
</value>
```

```
<value>
  <struct>
    <member>
      <name>label</name>
      <value>Technical</value>
    </member>
    <member>
      <name>ID</name>
      <value>techdash</value>
    </member>
  </struct>
</value>
</data>
</array>
</value>
</member>
</struct>
</value>
...intentionally truncated here...
</param>
</params>
</methodResponse>
```


## Lab exercise: Making a NetInsight API call

Here's an exercise that will get you started making API calls to NetInsight in just a few minutes. In it you will make GetProfiles and GetReportData method calls and write the responses to a file. You will also do some basic filtering. For this exercise we will use Perl to communicate with the NetInsight API.

For this exercise you will need:

1. An installation of NetInsight. From that instance you will need:
  - If it authenticates users, the username and password of an account with the privileges to run reports.
  - The HTTP path to NetInsight's CGI directory. For example, `http://netinsight.mycompany.com/cgi-bin/netinsight/` or `http://netinsight.mycompany.com/scripts/netinsight/`.
  - The short profile name of a profile you want to use.
  - Optionally, the report key of a custom report you want to use.
2. A computer with Active Perl installed. This can be the same computer on which NetInsight is installed. If it is not, it must be able to communicate with the NetInsight server via HTTP.

---

 Before starting the exercise, take a moment to ensure that your NetInsight instance is running and that you can connect to it normally via a browser.

---

- ! Throughout this tutorial, when copying and pasting code from this document, take heed of spaces and/or line breaks introduced by the copy process. In some cases these can break code or commands.

## Step 1: Create the script files

The first step is to create a Perl connection script and two XML files containing the method calls we want to make.

1. On the computer with Active Perl, create a directory named `niapiscripts`.
2. In a text editor, open a new file and copy and paste in the following code. You may need to make these edits:
  - If your NetInsight installation uses authentication, modify the values for `my $conf_username` and `my $conf_password` to an account with NetInsight access.
  - For most users the path to the Perl directory should be correct as is. But adjust it if necessary.

```
#!/usr/bin/perl -w
use strict;
use LWP;
use HTTP::Request;
my $url = shift;
my $filename = shift;
open( F, "< $filename" ) or die "Error: Can't open $filename : !\nUsage:
xmlrpc.pl targetURL xmlrequestfile\n";
my $content = join("",<F>);
print "Sent:\n$content\n";
print "Returned:\n";
my $conf_username = "username";
my $conf_password = "password";
my ( $response, $request, $header, $ua, $uname, $password );
$header = HTTP::Headers->new;
$header->authorization_basic( $conf_username, $conf_password );
$header->header('Content-Type' => 'text/xml');
$request = HTTP::Request->new( "POST", $url, $header, $content );
$ua = LWP::UserAgent->new;
$ua->agent(
"Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1;+.NET+CLR+1.1.4322)"
);
$response = $ua->request( $request );
if ( $response->is_success )
{
print $response->content . "\n";
}
else
{
print $response->status_line . "\n";
```

3. Save the file to the `niapiscripts` directory, naming it `niapi.pl`.

4. Open another new text file and paste the following code into it. Save the file to the `niapiscripts` directory, naming it `getprofiles.xml`.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
  <methodName>GetProfiles</methodName>
</methodCall>
```

5. Open a third new text file and paste the following code into it. You may need to make these edits:

- This call uses the report key for NetInsight's out-of-box Page Summary report (`<value>page</value>`). If your profile does not have the Page Summary report, or if you want to use a different report, enter a different report key.
- This call includes a filter that limits rows to the current calendar month. If your profile does not have data in that range, specify a different one or just remove the filter.

```
<?xml version="1.0" encoding="UTF-8"?>
<methodCall>
<methodName>GetReportData</methodName>
  <params>
    <param>
      <value>page</value>
    </param>
    <param>
      <value>
        <struct>
          <member>
            <name>show</name>
            <value>10</value>
          </member>
          <member>
            <name>filters</name>
            <value><![CDATA[
              <filterinstances>
                <filterinstance type="date" comparison="between">
                  <values>
                    <duration direction="current" type="months">1</duration>
                  </values>
                </filterinstance>
              </filterinstances>
            ]]></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

6. Save the file to the `niapiscripts` directory, naming it `getreportdata.xml`.

## Step 2: Make the GetProfiles method call

In this step we will get a list of the profiles running on the NetInsight instance.

1. On the computer with Active Perl, open a command prompt.
2. Change the current directory to the `niscscripts` directory.
3. At the command prompt, enter `niapi.pl`  
`http://ni_server_url/cgi_directory/ni_root_dir/ni_program_file`  
`getprofiles.xml`  
  
For example, if NetInsight is installed on Windows running IIS, you will enter something close to:  
`niapi.pl`  
`http://server.domain.com/scripts/netinsight/ntcgi.exe`  
`getprofiles.xml`  
  
If NetInsight is installed on Linux running Apache, you will enter something close to:  
`niapi.pl http://server.domain.com/cgi-gin/netinsight/ntcgi.cgi`  
`getprofiles.xml`
4. Press Enter. You should get back an XML-formatted list of the profiles in that NetInsight instance.

### Step 3: Make the GetReportData method call

In this step we will get data from an actual report and write it to a text file.

1. On the computer with Active Perl, open a command prompt.
2. Change the current directory to the `niscscripts` directory.
3. At the command prompt, enter `niapi.pl`  
`http://ni_server_url/cgi_directory/ni_root_dir/profile_name/ni_pr`  
`ogram_file getreportdata.xml > testdata.xml`  
  
The command we're entering this time differs in three ways from the one we made in the previous step: 1) The URL goes one directory level deeper, into a specific profile directory; 2) we're calling `getreportdata.xml`; the addition of "`> testdata.xml`" will direct the response output to a file named `testdata.xml` in the `niscscripts` directory.
4. Press Enter. After a moment examine the `niscscripts` directory. It should now contain a file named `testdata.xml`. And that file should contain XML-encoded NetInsight report data.

### Step 4: Experiment

Once you have gotten a successful response to to your initial GetReportData call, try modifying `getreportdata.xml` to retrieve data from other reports. Experiment with the different types of dimension filters described elsewhere in this document. For example, try creating calls that return:

- The most popular keyword/search engine combinations for your site
- Yesterday's ten most popular pages among Macintosh users
- The top weekday traffic hours for site visitors from the UK
- Today's most viewed products



## Troubleshooting API calls

This topic list some common error conditions and possible causes.

| Symptom  | Possible Cause   |
|--|--|
| The response returns: "Failed to deserialize drill file : Failed to deserialize reportdef <i>somereport</i> : ReportDefinition: :bDeserializeFromFile - failed to parse XML" | The report key you specified in the call is not valid.   |
| The response returns: "Error parsing XML-RPC Request"  | There is a syntax error in your request XML. Check it to make sure it is well-formed.  |
| The response returns: "Unknown method"   | The method name you specified is not valid. Verify that it is spelled correctly and that there are no spaces between it and its wrapper element.<br><br>...OR...<br><br>You attempted to make a GetReportData or Get-ReportList call from outside a profile directory. |
| The response returns: "GetProfiles: Function can only be called from the NetInsight homepage."   | You attempted to make a GetProfiles call from within a profile directory. GetProfiles calls must be made from one directory level higher.  |
| The response returns: "GetReportData: The report request limit has been exceeded."   | You have exceeded the monthly limit of report requests or retrieved report row allowed by your NetInsight license.   |
| The response returns: "404 Object Not Found"   | The path you specified in your request is invalid.   |

## Appendix: NetInsight Pre-Defined Report Keys

The table below list the report keys of the pre-defined reports that ship with Unica NetInsight. You can also lookup report keys from within the NetInsight UI: In the Custom Report Wizard on the Custom tab, view the list of available reports. Each report's key is shown in parentheses alongside its title.

| Report Title             | Report Key |
|--------------------------|------------|
| 3-digit Zip Code Summary | zip3       |
| 5-digit Zip Code Summary | zip5       |
| Area Code Summary        | areacode   |

| Report Title                  | Report Key       |
|-------------------------------|------------------|
| Banner Ad Summary             | ad               |
| Browser Breakdown             | browserbreakdown |
| Browser Summary               | browser          |
| Campaign Channel Summary      | channel          |
| Campaign Channel Type Summary | channeltype      |
| Campaign Segment Summary      | segment          |
| Campaign Summary              | campaign         |
| Cart Activity Trend           | carttrend        |
| City Summary                  | city             |
| Clickthrough Breakdown        | adclk            |
| Connection Type Summary       | conntype         |
| Content Dashboard             | contentdash      |
| Content Summary               | contentgroup     |
| Continent Summary             | continent        |
| Cookie Summary                | cookie           |
| Country Summary               | country          |
| Date Summary                  | date             |
| Day of the Week Summary       | dow              |
| Department Summary            | depart           |
| Directory Summary             | directory        |
| Domain Summary                | domain           |
| Entry Page Breakdown          | entrybreakdown   |
| Entry Page Summary            | entry            |

| <b>Report Title</b>        | <b>Report Key</b> |
|----------------------------|-------------------|
| Error Breakdown            | errorbreakdown    |
| Error Summary              | error             |
| Event Summary              | event             |
| Executive Dashboard        | execdash          |
| Exit Page Breakdown        | exitbreakdown     |
| Exit Page Summary          | exit              |
| Geographic Dashboard       | geodash           |
| Host Summary               | host              |
| Impression Breakdown       | adimp             |
| Initial Referrer Breakdown | initrefbreakdown  |
| Initial Referrer Summary   | initref           |
| Keyword Summary            | keyword           |
| Link Breakdown             | linkbr            |
| Link Summary               | link              |
| Local Keyword Summary      | lkey              |
| Marketing Dashboard        | marketdash        |
| Organization Summary       | organization      |
| Overlay Page Dashboard     | pageexec          |
| Overlay Site Dashboard     | siteexec          |
| Page Breakdown             | pagebreakdown     |
| Page Delivery Summary      | pagedeliv         |
| Page Summary               | page              |
| Page View Summary          | pageview          |

| Report Title                         | Report Key        |
|--------------------------------------|-------------------|
| Paid Keyword Summary                 | paidkeyword       |
| Paid Search Campaign Channel Summary | psichannel        |
| Paid Search Campaign Segment Summary | psisegment        |
| Paid Search Campaign Summary         | psicampaign       |
| Paid Search Paid Keywords Summary    | psipaidkeyword    |
| Path Summary                         | pathentry         |
| Platform Summary                     | platform          |
| Product Abandonment Summary          | prodabandon       |
| Product Action Summary               | prodaction        |
| Product Conversion Summary           | prodconversion    |
| Referrer Breakdown                   | referrerbreakdown |
| Referrer Summary                     | referrer          |
| Repeat Visitor Summary               | repeat            |
| Retail Dashboard                     | retaildash        |
| Revenue Trend                        | revenueread       |
| Robot/Spider Breakdown               | spiderbreakdown   |
| Robot/Spider Executive Dashboard     | spiderexecdash    |
| Robot/Spider Summary                 | spider            |
| Robot/Spider View Breakdown          | spiderview        |
| Robot/Spider Visit Breakdown         | spidervisit       |
| Robot/Spider Visitor Summary         | spidervisitor     |
| Screen Resolution Summary            | screenres         |
| Server Performance Summary           | serverperf        |

---

| <b>Report Title</b>       | <b>Report Key</b> |
|---------------------------|-------------------|
| Server Summary            | server            |
| State Summary             | state             |
| Technical Dashboard       | techdash          |
| Time Summary              | time              |
| Time Zone Summary         | timezone          |
| Traffic Dashboard         | traffidash        |
| Traffic Summary           | traffic           |
| User Summary              | user              |
| View Breakdown            | view              |
| Visit Breakdown           | visit             |
| Visit Duration Summary    | duration          |
| Visitor Dashboard         | visdash           |
| Visitor Profile Summary   | visprofile        |
| Visitor Retention Summary | retention         |
| Visitor Summary           | visitor           |

---