GNSS User's Guide
or
How to query the CGNS, its API and WFS

History : A. Mainville, J-F. Saulnier, Sept. 2004.
A. Mainville, April 2005.
A. Mainville, J-F. Saulnier, Oct. 2005, version 1.0
A. Mainville, Nov. 2005, version 1.1 (changes in sections 2.0, 3.2 and 9.0)

Content :

1.0 - Introduction :

There are four ways (client interfaces) to query the Canadian Geographical Names Service (CGNS) through the Web: (1) The GNSS (Geographical Name Search Service), (2) its API, (3) its WFS (Web Feature Service), and (4) its Refer functionality. Or again:

1- http://cgns.nrcan.gc.ca or more directly http://gnss.nrcan.gc.ca/gnss-srt

2- http://gnss.nrcan.gc.ca/gnss-srt/api?geoname=Ottawa&output=html

3- http://cgns.nrcan.gc.ca/wfs/cubeserv.cgi?service=wfs&datastore=cgns

4- http://gnss.nrcan.gc.ca/gnss-srt/referExample.html

1 - The 1st way is to use the Web site http://gnss.nrcan.gc.ca/gnss-srt where you enter a name and get an answer from the geographical names database. There site offers many search options, output formats, windows as well as information describing the possibilities of the CGNS. For more details, see section 2.0.

2 - The 2nd way, a person (or software) may send a URI with parameters to the CGNS. For example, http://gnss.nrcan.gc.ca/gnss-srt/api?geoname=Ottawa&output=html&language=en. The possible options (parameters) are found at http://gnss.nrcan.gc.ca/gnss-srt/help.jsp, under "Developer". For more

details, see section 3.0. To see a list of the parameters, see section 5.0. To use our development sites, in view of testing your own applications, see section 7.0.

3 - The 3rd way is to query directly the WFS Web Service that holds the CGNS data. One has to send a XML file to the following URL:

http://cgns.nrcan.gc.ca/wfs/cubeserv.cgi?service=wfs&datastore=cgns

A first way to send the XML code is to insert it into the URI as shown here. Try it:

http://cgns.nrcan.gc.ca/wfs/cubeserv.cgi?service=wfs&datastore=cgns&typename=GEONAMES&request=GETFEATURE&filter=%3CFilter%3E%3CPropertyIsEqualTo%3E%3CPropertyName%3ECGNDB_KEY%3C/PropertyName%3E%3CLiteral%3EGAOFC%3C/Literal%3E%3C/PropertyIsEqualTo%3E%3C/Filter%3E

The same result is achieved if you try this:

http://www.cubewerx.com/cwpost/cwpost.cgi?serverUrl=http://cgdi-dev.geoconnections.org:6571/dev/wfst/cubeserv.cgi?service=wfs%26datastore=cgns&postBody=Paste%20your%20transaction%20here.

When asked for the server URL, paste:

http://cgns.nrcan.gc.ca/wfs/cubeserv.cgi?service=wfs&datastore=cgns.

And when asked for 'Paste your transaction here', paste the following XML code:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--      This query uses the LIKE function... -->
<GetFeature srsName="EPSG:4269">
<Query typeName="GEONAMES">
  <PropertyName>GEONAME</PropertyName>
  <PropertyName>CGNDB_KEY</PropertyName>
  <PropertyName>CONCISE_CODE</PropertyName>
  <PropertyName>REGION_CODE</PropertyName>
  <PropertyName>NTS_MAP</PropertyName>
  <PropertyName>HYPER_LINK</PropertyName>
  <PropertyName>GEOMETRY</PropertyName>
  <PropertyName>FEATURE_ID</PropertyName>
  <PropertyName>GENERIC_CODE</PropertyName>
<Filter>
<PropertyIsLike>
  <PropertyName>GEONAMES.NAME_KEY</PropertyName>
  <Literal>%ALBERT</Literal>
</PropertyIsLike>
```

</Filter>
</Query>
</GetFeature>

Push the “Submit” button to obtain the result. For more details, see section 4.0. To get a list of the parameters, keywords and their definition see section 5.0. To see an overview of the available data (the database data structure) see section 6.0. To use our developmental sites, in view of testing your own applications, see section 7.0.

4 - The 4th way is to get the Advanced Search page of the GNSS Web site to be implemented into your own application. The method is explained at the following URI: http://gnss.nrcan.gc.ca/gnss-srt/referExample.html and also at http://cgns-dev.nrcan.gc.ca/gnss-srt/help_api.jsp#refer

2.0 - GNSS Web site :

The URI http://gnss.nrcan.gc.ca/gnss-srt leads to the “Geographical Names Search Service” and its default tab called “Name Search” where you can perform a search by geographical name. Geographical names (geonames) are stored in the database as they would be displayed on a geographical map, that is, as “Lake Abitibi” (not as “Abitibi, Lake” as in a gazetteer).

Wildcard characters:

The wildcard character “*” is very useful to query the database. One or multiple wildcards may be used; for example, *Abitibi or s*abitibi or *s*t*j. The web site always adds a wildcard at the end of the geoname, unless you click the box “Exact”.

The wildcard characters % (used in many software) and %25 (the ASCII form of % often seen in URI) are equivalent to “*”.

Aboriginal characters:

Aboriginal characters are stored in the database using a number between two curly brackets, i.e., {1}, {2}, {3},…{35}. The list of characters are seen under the link “Click here to insert an Aboriginal character”. Geonames containing Aboriginal characters may be queried by requesting as Geoname, say: {1} or try this: *{2}.

Output format:

Four output format options are available (XML, html table, delimited list, standard display).

Advanced Search:

The “Advanced Search” tab enables more searching options. The user can search by CGNDB key, feature ID, concise term, region, status, by NTS map or by coordinates and a radius.

3.0 - GNSS API

This technique returns always the same columns (fields). It differs from the other method called WFS (see section 4.0), where the numbers of columns returned may be modified.

In fact, the API uses the WFS system to query the database CGNS. The API is a front end to the WFS and was created to be simple to use, to validate the request, provide informative error messages, and return specific queries in a simple way.

Now on are examples of queries using the API:

3.1 - Output format :

Three output formats are available.

http://gnss.nrcan.gc.ca/gnss-srt/api?geoname=montreal&output=html
http://gnss.nrcan.gc.ca/gnss-srt/api?geoname=toronto&output=csv
http://gnss.nrcan.gc.ca/gnss-srt/api?geoname=vancouver&output=xml

output = html, csv or xml (by default).

3.2 - Searching feature name (geoname) :

http://gnss.nrcan.gc.ca/gnss-srt/api?geoname=toronto&output=html&match=exact

http://gnss.nrcan.gc.ca/gnss-srt/api?geoname=toron&output=html&match=leading
is the same as
http://gnss.nrcan.gc.ca/gnss-srt/api?geoname=toron&output=html

geoname = [minimum of 2 characters]
match = exact or leading (by default)

When using wildcards in the API search, you may add a parameter called "match" in the call with either a value of "exact" or "leading". Leading puts a wildcard at the end (not at the beginning) of the name key for the search. The wildcard is %25, for example:

http://gnss.nrcan.gc.ca/gnss-srt/api?geoname=%25white%25mount&output=html

To search Aborignal names containing special characters (see Section 2.0), try this:

http://gnss.nrcan.gc.ca/gnss-srt/api?geoname=%25{2}&output=html

3.3 - Searching by CGNDB key :

http://gnss.nrcan.gc.ca/gnss-srt/api?cgndbKey=BADHP&output=html
http://gnss.nrcan.gc.ca/gnss-srt/api?cgndbKey=BADHP&output=csv
http://gnss.nrcan.gc.ca/gnss-srt/api?cgndbKey=BADHP&output=xml

cgndbKey = [5 caractères]

The keyword “match” doesn’t work with the parameter cgndbKey (to use wildcard).

3.4 - Searching by featureId :

http://gnss.nrcan.gc.ca/gnss-srt/api?featureId=4d00fe4cba2011d892e2080020a0f4c9&output=html&language=en

featureId = [32 characters]

3.5 - Searching by dates :

http://gnss.nrcan.gc.ca/gnss-srt/api?conciseCode=CITY&regionCode=35&dateSince=1990-02-01&output=html

dateSince = YYYY-MM-DD

3.6 - Seaching by circle :

http://gnss.nrcan.gc.ca/gnss-srt/api?circle=-98,54,180&conciseCode=CITY,TOWN&output=html

circle = longitude of centre (in degrees with decimals, negative for longitudes in the north-americain system), latitude of centre, and radius of search area in kilometres.

3.7 - Searching by conciseCode :

http://gnss.nrcan.gc.ca/gnss-srt/api?regionCode=35&conciseCode=CITY,TOWN&output=html

conciseCode = [2-4 characters],[2-4 characters],etc.

Many concise codes may be entered. For example to get all city, town, municipalities, … set:

ConciseCode=CITY,TOWN,VILG,HAM,UNP,MUN1,MUN2

CITY City

TOWN	Town
VILG	Village
HAM	Hamlet
UNP	Unincorporated area
IR	Indian Reserve
GEOG	Geographical area
PARK	Conservation area
MIL	Military area
PROV	Province
TERR	Territory
MUN1	Other municipal/district area - major agglomeration
MUN2	Other municipal/district area - miscellaneous

Other concise codes :

SEAU	Undersea feature
RIV	River
LAKE	Lake
SPRG	Spring
CHAN	Channel
FALL	Falls
RAP	Rapids
RIVF	River feature
BAY	Bay
CAPE	Cape
BCH	Beach
SHL	Shoal
ISL	Island
MTN	Mountain
VALL	Valley
PLN	Plain
GLAC	Glacier
FOR	Forest
CAVE	Cave
CRAT	Crater
SEA	Sea
MISC	Miscellaneous
VEGL	Low vegetation
SEAF	Sea feature
CLF	Cliff

3. 8 - Searching by RegionCode :

http://gnss.nrcan.gc.ca/gnss-srt/api?geoname=Charlot&regionCode=11

regionCode = [2 digits],[2 digits],etc.

10 Newfoundland and Labrador
11 Prince Edward Island
12 Nova Scotia
13 New Brunswick
24 Quebec
35 Ontario
46 Manitoba
47 Saskatchewan
48 Alberta
59 British Columbia
60 Yukon
61 Northwest Territories
62 Nunavut

3. 8.1 – Downloading the whole database - Searching by RegionCode and BBox:

There are about 350,000 names in the database, and there is a limit to the number of records returned by one query, the limit is currently 10,000 records on one query. To download the whole database, one requires 41 queries that look like this:

http://gnss.nrcan.gc.ca/gnss-srt/api?bbox=-70.0,45.0:-60.0,47.0&regionCode=13

http://gnss.nrcan.gc.ca/gnss-srt/api?bbox=-70.0,47.0:-60.0,50.0&regionCode=13

That is, query by regions and Bounding Box. Some regions have more than 10,000 records so require more queries:

QC = regionCode = 24 requires 12 queries,
ON = regionCode = 35 requires 6 queries,
BC = regionCode = 59 requires 5 queries,
NL = regionCode = 10 requires 3 queries,
NS = regionCode = 12 requires 3 queries,
NB = regionCode = 13 requires 2 queries,
MB = regionCode = 46 requires 2 queries,
SK = regionCode = 47 requires 2 queries,

regionCode = 11, 48, 60, 61, 62 and 73 requires 1 query.

If you encounter a space-memory problem, then the same query may be obtained using the WFS service described in Section 4.3.

3.9 - Searching by status code:

http://gnss.nrcan.gc.ca/gnss-srt/api?statusCode=A1&regionCode=11

statusCode = [2 charcaters],[2 characters],etc.

Note : The API returns a field called status_term = Official, Rescinded or Withdrawn. To get the API to return only the Official names, set the statusCode to :

statusCode=A1,A2,A3,A4,A5,A6,A7,A8,A10,A11,A12,A13,M1,P1,Q1,S1,U1,U2,U3,W1

3.10 - Searching by NTS map number :

http://gnss.nrcan.gc.ca/gnss-srt/api?ntsMap=011L03
or
http://gnss.nrcan.gc.ca/gnss-srt/api?ntsMap=C.4405

ntsMap = [NTS no of 6 characters, 999A99] or [Can. hydrographic chart no, C.9999]

For example, the Geobase Portal (http://geobase.ca) uses the following command to provide names to their users, on a map basis :

http://gnss.nrcan.gc.ca/gnss-srt/api?statusCode=A1,A2,A3,A4,A5,A6,A7,A8,A10,A11,A12,A13,M1,P1,Q1,S1,U1,U2,U3,W1&ntsMap=011L03&language=en

3.11 - Restricting the results to output either the English version of the name, or the French version (but not both) when name exist in both official languages :

http://gnss.nrcan.gc.ca/gnss-srt/api?nameLanguage=fr&regionCode=11

nameLanguage = en or fr (not implemented yet, to come)

3.12 - Returning the results with headings and code definitions in English or in French :

http://gnss.nrcan.gc.ca/gnss-srt/api?language=fr&regionCode=11

language = en or fr

3.13 - Example of results in XML format

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <SearchResults xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <PlaceName>
      <geoname>Charlottetown</geoname>
      <status_term>Official</status_term>
      <latitude>46° 15' North</latitude>
      <longitude>63° 8' West</longitude>
      <latdec>46.2500000</latdec>
```

```
    <londec>-63.1332999</londec>
    <coord_acc_m>2000</coord_acc_m>
    <concise_term>Geographical area</concise_term>
    <generic_term>Royalty</generic_term>
    <region_name>Prince Edward Island</region_name>
    <location>Queens</location>
    <nts_map>011L03,011L06</nts_map>
    <datum>NAD27</datum>
    <cgndb_key>BAETC</cgndb_key>
  <feature_id>0c0b95ab849c20c3ad1819042edaf7ec</feature_id>
 </PlaceName>
```

3.14 - Here are other outputs that could potentially be made available :

```
<value>STATUS_CODE</value>
<value>GENERIC_CODE</value>
<value>REGION_CODE</value>
<value>ORIGINAL_APPROVAL_DATE</value>
<value>DECISION_DATE</value>
<value>EFFECTIVE_DATE</value>
<value>NAME_KEY</value>
<value>RS_VALUE</value>
<value>DATE_CREATED</value>
<value>DATE_MODIFIED</value>
<value>COORDINATES_ACCURACY</value>
```

4.0 - GNSS WFS

Using the WFS technique, the numbers of columns returned is controlled.

Copy and paste the XML code below into the “cwpost” CubeWerx utility screen (used to debug the XML code) located at :

http://www.cubewerx.com/cwpost/cwpost.cgi?serverUrl=http://cgdi-dev.geoconnections.org:6571/dev/wfst/cubeserv.cgi?service=wfs%26datastore=cgns&postBody=Paste%20your%20transaction%20here

When asked for the server URL, enter :

http://cgns.nrcan.gc.ca/wfs/cubeserv.cgi?service=wfs&datastore=cgns

4.1.1 - Searching using CGNDB key :

Here the simplest case:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<GetFeature srsName="EPSG:4269">
<Query typeName="GEONAMES">
<Filter>
<PropertyIsEqualTo>
  <PropertyName>CGNDB_KEY</PropertyName>
  <Literal>GAOFC</Literal>
</PropertyIsEqualTo>
</Filter>
</Query>
</GetFeature>
```

4.1.2 - Searching using geoname :

Here's an example of a query using the variable name_key (e.g., name_key like '%ALBERT%'.) Adding the percent after Albert is not necessary, it is added by the CGNS software application. A percent in front of ALBERT returns for example Prince Albert. Name_key requires ALBERT to be entered in upper case, without blank spaces and without special characters (except the % wildcard character):

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--      This query uses the LIKE function... -->
<GetFeature srsName="EPSG:4269">
<Query typeName="GEONAMES">
  <PropertyName>GEONAME</PropertyName>
  <PropertyName>CGNDB_KEY</PropertyName>
  <PropertyName>CONCISE_CODE</PropertyName>
  <PropertyName>REGION_CODE</PropertyName>
  <PropertyName>NTS_MAP</PropertyName>
  <PropertyName>HYPER_LINK</PropertyName>
  <PropertyName>GEOMETRY</PropertyName>
  <PropertyName>FEATURE_ID</PropertyName>
  <PropertyName>GENERIC_CODE</PropertyName>
<Filter>
<PropertyIsLike>
  <PropertyName>NAME_KEY</PropertyName>
  <Literal>%ALBERT</Literal>
</PropertyIsLike>
</Filter>
</Query>
</GetFeature>
```

4.2 - Searching using RS_value (relevance of name at map scale) :

```
<PropertyIsEqualTo>
  <PropertyName>RS_VALUE</PropertyName>
```

```
  <Literal>15000000</Literal>
</PropertyIsEqualTo>
```

or

http://cgns.nrcan.gc.ca/wfs/cubeserv.cgi?service=wfs&datastore=cgns&typename=GEONAMES&request=GETFEATURE&propertyname=geoname&filter=<Filter><PropertyIsEqualTo><PropertyName>RS_VALUE</PropertyName><Literal>15000000</Literal></PropertyIsEqualTo></Filter>

4.3 – Downloading the whole database - Searching by RegionCode and BBox:

There are about 350,000 names in the database, and there is a limit to the number of records returned by one query, the limit is currently 10,000 records on one query. To download the whole database, one requires 41 queries that looks like this:

http://cgns.nrcan.gc.ca/wfs/cubeserv.cgi?service=wfs&datastore=cgns&typename=GEONAMES&request=GETFEATURE&filter=<Filter><And><PropertyIsEqualTo><PropertyName>REGION_CODE</PropertyName><Literal>13</Literal></PropertyIsEqualTo><Not><Disjoint><PropertyName>GEOMETRY</PropertyName><Box srsName="EPSG:4269"><coordinates>-70.0,45.0 -60.0,47.0</coordinates></Box></Disjoint></Not></And></Filter>

That is, query by regions and Bounding Box. Some regions have more than 10,000 records so require more queries. See Section 3.8.1.

4.4 - Searching records created by Provinces/Territories :

```
<PropertyIsEqualTo>
  <PropertyName>source_created</PropertyName>
  <Literal>GNAPP</Literal>
</PropertyIsEqualTo>
```

4.5 - Here's an example of a query using coordinates within a box, a spatial area.

```
<Not>
<Disjoint>
<PropertyName>GEOMETRY</PropertyName>
  <gml:Box srsName="EPSG:4269" >
  <gml:coordinates cs="," decimal="." ts=":">-98.8,53.2:-97.2,54.8
  </gml:coordinates>
  </gml:Box>
</Disjoint>
</Not>

<Not>
```

```
<Disjoint>
<PropertyName>GEOMETRY</PropertyName>
  <Box srsName="EPSG:4269" >
  <coordinates>-98.8,53.2 -97.2,54.8
  </coordinates>
  </Box>
</Disjoint>
</Not>

<Bbox>
<PropertyName>GEOMETRY</PropertyName>
  <Box srsName="EPSG:4269" >
  <coordinates>-98.8,53.2 -97.2,54.8</coordinates>
  </Box>
</Bbox>
```

4.6 - Example of using a filter with “And” or “Or”:

```
<And>
<PropertyIsLike>
 <PropertyName>NAME_KEY</PropertyName>
 <Literal>%FORT</Literal>
</PropertyIsLike>
<PropertyIsEqualTo>
 <PropertyName>REGION_CODE</PropertyName>
 <Literal>61</Literal>
</PropertyIsEqualTo>
</And>

<Or>
<PropertyIsEqualTo>
 <PropertyName>PC_NAME_LANG</PropertyName>
 <Literal>E </Literal>
</PropertyIsEqualTo>
<PropertyIsEqualTo>
 <PropertyName> PC_NAME_LANG </PropertyName>
 <Literal>F</Literal>
</PropertyIsEqualTo>
</Or>
```

N.B.: The box must be the last constraint:

```
<And>
<PropertyIsEqualTo>
 <PropertyName>SOURCE_CREATED</PropertyName>
 <Literal>GNAPP</Literal>
```

```
</PropertyIsEqualTo>
<Not>
<Disjoint>
  <PropertyName>GEOMETRY</PropertyName>
  <Box srsName="EPSG:4269">
  <coordinates>-105,59 -100,60</coordinates>
  </Box>
</Disjoint>
</Not>
</And>
```

4.7 - Example of a filter using a mixture of “And” and “Or”:

(NAME = Winnipeg OR NAME = Ottawa) AND (key = GBEIN):

```
<And>
<Or>
<PropertyIsEqualTo>
  <PropertyName>FEATURE_NAME</PropertyName>
  <Literal>Winnipeg</Literal>
</PropertyIsEqualTo>
<PropertyIsEqualTo>
  <PropertyName>FEATURE_NAME</PropertyName>
  <Literal>Ottawa</Literal>
</PropertyIsEqualTo>
</Or>
<PropertyIsEqualTo>
  <PropertyName>cgndb_key</PropertyName>
  <Literal>GBEIN</Literal>
</PropertyIsEqualTo>
</And>
```

4.8 - Using isBetween:

```
<PropertyIsBetween>
    <PropertyName>generic_code</PropertyName>
    <LowerBoundary>1</LowerBoundary>
    <UpperBoundary>3</UpperBoundary>
</PropertyIsBetween>

<Between>
     <PropertyName>SAMPLE_DATE</PropertyName>
     <LowerBoundary>
     <Literal>2001-01-15T20:07:48.11</Literal>
     </LowerBoundary>
```

```
    <UpperBoundary>
    <Literal>2001-03-06T12:00:00.00</Literal>
    </UpperBoundary>
</Between>
```

4.9 -Using isLess or isGreather:

```
<PropertyIsLessThan>
  <PropertyName>generic_code</PropertyName>
  <Literal>2</Literal>
</PropertyIsLessThan>
```

4.10 - Using isLessThanOrEqualTo or is GreatherThanOrEqualTo:

```
<PropertyIsLessThanOrEqualTo>
  <PropertyName>generic_code</PropertyName>
  <Literal>2</Literal>
</PropertyIsLessThanOrEqualTo>
```

4.11 - Using isNull :

```
<PropertyIsNull>
  <PropertyName>feature_id</PropertyName>
</PropertyIsNull>
```

4.12 - Choosing output columns :

In the following example, the line <PropertyName>GEONAME</PropertyName> allows to request that the column Geoname be returned from the database. If no column is requested, all columns are returned. We may thus request here the columns desired. And we control thus the amount of information returned.

If at least one column is requested it will return that columns plus all the columns defined as being not null in the database. (This is so because the WFS specification states that it will return an XML document that validates against the database schema. So every database columns defined as “not null” will be returned.)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<GetFeature srsName="EPSG:4269">
<Query typeName="GEONAMES">
  <PropertyName>GEONAME</PropertyName>
<Filter>
<PropertyIsEqualTo>
  <PropertyName>REGION_CODE</PropertyName>
  <Literal>11</Literal>
```

```
</PropertyIsEqualTo>
</Filter>
</Query>
</GetFeature>
```

To get all columns returned from the CGNS database, for a specific region, it is not necessary to specify all the fields. Hence, case (a) works as well as case (b):

Case (a):

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<GetFeature srsName="EPSG:4269">
<Query typeName="GEONAMES">
<Filter>
<PropertyIsEqualTo>
  <PropertyName>REGION_CODE</PropertyName>
  <Literal>11</Literal>
</PropertyIsEqualTo>
</Filter>
</Query>
</GetFeature>
```

Case (b):

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<GetFeature srsName="EPSG:4269">
<Query typeName="GEONAMES">
  <PropertyName>GEONAME</PropertyName>
  <PropertyName>CGNDB_KEY</PropertyName>
  <PropertyName>REGION_CODE</PropertyName>
  <PropertyName>STATUS_CODE</PropertyName>
  <PropertyName>PC_NAME_LANG</PropertyName>
  <PropertyName>CONCISE_CODE</PropertyName>
  <PropertyName>GENERIC_CODE</PropertyName>
  <PropertyName>LOCATION</PropertyName>
  <PropertyName>NTS_MAP</PropertyName>
  <PropertyName>COORDINATES_ACCURACY</PropertyName>
  <PropertyName>GEOMETRY</PropertyName>
  <PropertyName>SOURCE_RECORD_ID</PropertyName>
  <PropertyName>SOURCE_FEATURE_ID</PropertyName>
  <PropertyName>FEATURE_ID</PropertyName>
  <PropertyName>NAME_KEY</PropertyName>
  <PropertyName>RS_VALUE</PropertyName>
  <PropertyName>HYPER_LINK</PropertyName>
  <PropertyName>ORIGINAL_APPROVAL_DATE</PropertyName>
  <PropertyName>DECISION_DATE</PropertyName>
```

```
  <PropertyName>EFFECTIVE_DATE</PropertyName>
  <PropertyName>DATE_CREATED</PropertyName>
  <PropertyName>SOURCE_CREATED</PropertyName>
  <PropertyName>DATE_MODIFIED</PropertyName>
  <PropertyName>SOURCE_MODIFIED</PropertyName>
  <PropertyName>CONCISE_GAZ_FLAG</PropertyName>
<Filter>
<PropertyIsEqualTo>
  <PropertyName>REGION_CODE</PropertyName>
  <Literal>11</Literal>
</PropertyIsEqualTo>
</Filter>
</Query>
</GetFeature>
```

4.13 - Possible output formats (see the parameter "getCapability", section 6.0):

```
<GetFeature srsName="EPSG:4269" outputFormat="GML.1">
<GetFeature srsName="EPSG:4269" outputFormat="GML2">
<GetFeature srsName="EPSG:4269" outputFormat="SHAPE">
<GetFeature srsName="EPSG:4269" outputFormat="MIF">
```

4.14 - WFS Reference Manuals

More examples (and the WFS reference manuals) are found in Annex A of http://www.opengeospatial.org/docs/02-059.pdf which is extracted from http://www.opengeospatial.org under Documents/OpenGIS Specifications/Filter Encoding.

4.15 - WFS applications

The CubeWerx utility screen is used to debug the XML code. When the XML is debuged, then it may be inserted into an URI or a HTML, Java, PL/sql, "C" or Visual Basic file.

Here two examples of a URI.

http://cgns.nrcan.gc.ca/wfs/cubeserv.cgi?service=wfs&datastore=cgns&typename=GEONAMES&request=GETFEATURE&filter=<Filter><PropertyIsEqualTo><PropertyName>CGNDB_KEY</PropertyName><Literal>GAOFC</Literal></PropertyIsEqualTo></Filter>

or

http://cgns.nrcan.gc.ca/wfs/cubeserv.cgi?service=wfs&datastore=cgns&typename=GEONAMES&request=GETFEATURE&filter=%3CFilter%3E%3CPropertyIsEqualTo%3E%

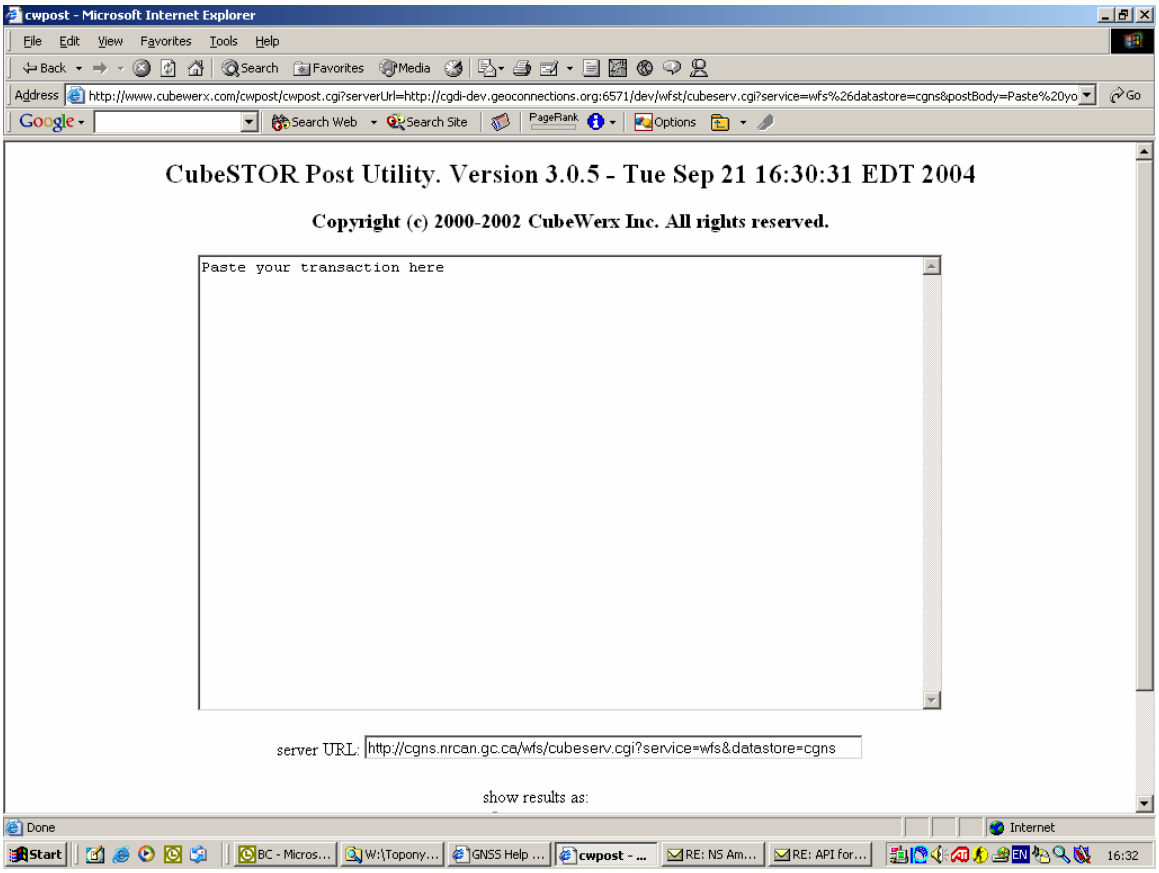
3CPropertyName%3ECGNDB_KEY%3C/PropertyName%3E%3CLiteral%3EGAOFC%3C/Literal%3E%3C/PropertyIsEqualTo%3E%3C/Filter%3E

where the following hexadecimal characters are used.

4.16 - Hexadecimal characters

00 NUL	01 SOH	02 STX	03 ETX	04 EOT	05 ENQ	06 ACK	07 BEL
08 BS	09 HT	0A NL	0B VT	0C NP	0D CR	0E SO	0F SI
10 DLE	11 DC1	12 DC2	13 DC3	14 DC4	15 NAK	16 SYN	17 ETB
18 CAN	19 EM	1A SUB	1B ESC	1C FS	1D GS	1E RS	1F US
20 SP	21 !	**22 "**	23 #	24 $	25 %	26 &	27 '
28 (	29)	2A *	2B +	2C ,	2D -	2E .	**2F /**
30 0	31 1	32 2	33 3	34 4	35 5	36 6	37 7
38 8	39 9	**3A :**	3B ;	**3C <**	**3D =**	**3E >**	3F ?
40 @	41 A	42 B	43 C	44 D	45 E	46 F	47 G
48 H	49 I	4A J	4B K	4C L	4D M	4E N	4F O
50 P	51 Q	52 R	53 S	54 T	55 U	56 V	57 W
58 X	59 Y	5A Z	5B [	5C \	5D]	5E ^	5F _
60 `	61 a	62 b	63 c	64 d	65 e	66 f	67 g
68 h	69 i	6A j	6B k	6C l	6D m	6E n	6F o
70 p	71 q	72 r	73 s	74 t	75 u	76 v	77 w
78 x	79 y	7A z	7B {	7C |	7D }	7E ~	7F DEL

Figure 1 - Cubewerx utility screen « cwpost ».



4.17 - More information

To find more information on WMS and WFS please go to
http://atlas.gc.ca/site/english/dataservices/web_map_service.html

5.0 - CGNS Services

The CGNS Services provide the list of parameters, codes, abbreviations and definitions used by the CGNS services. The CGNS Services also allows your application to validate the parameters and codes before sending the request to the CGNS.

1
http://cgns.nrcan.gc.ca/cgnsServices/services?request=getCapabilities
2
http://cgns.nrcan.gc.ca/cgnsServices/services?request=validateConciseCode&code=RIV, RIVF, LAKE, SEA, etc.
3
http://cgns.nrcan.gc.ca/cgnsServices/services?request=validateGenericCode&code=111

4
http://cgns.nrcan.gc.ca/cgnsServices/services?request=validateStatusCode&code=A1
5
http://cgns.nrcan.gc.ca/cgnsServices/services?request=validateMap&mapNumber=011P16
6
http://cgns.nrcan.gc.ca/cgnsServices/services?request=getGenericCodeDescription&code=111&language=en
7
http://cgns.nrcan.gc.ca/cgnsServices/services?request=getStatusCodeDescription&code=A1&language=en

Note: The following two options (8 and 9) complement the CGNS Services. They don't use the cgnsServices. They use the OGC WFS (Web Feature Service) and its Oracle table cgns_codes.

8
http://cgns.nrcan.gc.ca/wfs/cubeserv.cgi?service=wfs&datastore=cgns&version=0.0.13&typename=cgns_codes&request=describeFeatureType

9
http://cgns.nrcan.gc.ca/wfs/cubeserv.cgi?service=wfs&datastore=cgns&typename=cgns_codes&request=getFeature

Summary of options:

Type:

http://cgns.nrcan.gc.ca/cgnsServices/services?request=

followed by one of these:

getCapabilities
validateConciseCode&code=RIV
validateGenericCode&code=111
validateStatusCode&code=A1
validateMap&mapNumber=011P16
getGenericCodeDescription&code=111&language=en
getStatusCodeDescription&code=A1&language=en

Responses obtained:

1- displays:

Table 1 - CGNS Services – Capabilities

Request	Parameters

getGenericCodeDescription	code
	language
validateGenericCode	code
validateStatusCode	code
getStatusCodeDescription	code
	language
validateMap	mapNumber
validateConciseCode	code

2-, 3-, 4- and 5- display:
response=true or response=false

6- displays a definition like:
response=A named railway siding, junction, or flag stop with or without an agent.

7- displays a definition like:
response=Name officially approved by the CPCGN member.

8- displays OGC XML announcing the availability of Oracle table **CGNS_CODES** having the following columns:

```
Name           Null?     Type
-------------  --------  ---------------
COLUMN_NAME    NOT NULL  VARCHAR2(30)
VALUE          NOT NULL  VARCHAR2(700)
MEANING_ENG    NOT NULL  VARCHAR2(240)
MEANING_FR     NOT NULL  VARCHAR2(80)
ABSTRACT_ENG             VARCHAR2(240)
ABSTRACT_FR              VARCHAR2(2000)
URL                      VARCHAR2(2000)
VIEWNAME                 VARCHAR2(30)
```

9- displays a list of the generic codes (some 1000) in the XML format together with its English and French definition.

6.0 - WFS Capabilities

6.1 - getCapabilities

To find the data model of the database at a WFS site, and the capabilities of a WFS site , first run a URI containing the getCapabilities, for example:

http://cgns.nrcan.gc.ca/wfs/cubeserv.cgi?service=wfs&datastore=cgns&request=getCapabilities&version=1.0.0

or

http://cgns-dev.nrcan.gc.ca/cgi-bin/cubeserv.cgi?service=wfs&datastore=cgns&request=getCapabilities

where we see, say, <Name>**cw:GEONAMES**</Name>. Here GEONAMES is a table of data. To find the fields of the table GEONAMES, run a URI containing request=describefeaturetype&typename=GEONAMES.

6.2 - describefeaturetype

A URI containing the parameters describefeaturetype&typename=GEONAMES (where geonames is the name of the table) returns the columns of the table :

6.3 - GEONAMES tables

http://cgns.nrcan.gc.ca/wfs/cubeserv.cgi?SERVICE=wfs&DATASTORE=cgns&request=describefeaturetype&typename=GEONAMES
or simply
http://cgns-dev.nrcan.gc.ca/cgi-bin/cubeserv.cgi?service=wfs&datastore=cgns&request=describeFeatureType

6.4 - CGNS_CODES tables

http://cgns.nrcan.gc.ca/wfs/cubeserv.cgi?SERVICE=wfs&DATASTORE=cgns&request=describefeaturetype&typename=CGNS_CODES

7.0 – Development sites

Our web sites distributing official data to the public are called production sites. To test your new applications, we have developmental sites available.

For **api**, change gnss to cgns-dev, i.e., change
http://ggns.nrcan.gc.ca/gnss-srt/api?geoname=toronto&output=html&match=exact
to
http://cgns-dev.nrcan.gc.ca/gnss-srt/api?geoname=toronto&output=html&match=exact

For **WFS**, change cgns to cgns-dev and wfs to cgi-bin, i.e., change
http://cgns.nrcan.gc.ca/wfs/cubeserv.cgi?service=wfs&datastore=cgns
to
http://cgns-dev.nrcan.gc.ca/cgi-bin/cubeserv.cgi?service=wfs&datastore=cgns

8.0 – Other sites

GeoBase

A user may want to retrieve “geographical names” in relation to other layers (administrative boundaries, canadian digital elevation data (CDED), canadian geodetic network, Landsat 7 orthoimages, aerial photo control points, national road network) for particular geographical areas in order to incorporate those into a GIS.

This is possible through GeoBase, a federal, provincial and territorial government initiative that is overseen by the Canadian Council on Geomatics (CCOG). It is undertaken to ensure the provision of, and access to, a common, up-to-date and maintained base of quality geospatial data for all of Canada. The GeoBase URI is http://www.geobase.ca and it provides geographical names pulled out the CGNS using CGNS API.

9.0 – Special characters

As output, a WFS site returns the special character & as & There may be other characters that are returned in its html form.

Wildcard characters:

See Section 2.0.

Aboriginal characters:

See Section 2.0.

If you have comments or questions, do not hesitate to contact us,

André Mainville
Geographical Names Section / Section des noms géographiques
Natural Resources Canada / Ressources naturelles Canada
615 Booth St., Rm. 634 / 615 rue Booth, pièce 634
Ottawa ON K1A 0E9
Government of Canada / Gouvernement du Canada
Phone/Téléphone : (613) 995-4504
Fax/Téléc. : (613) 943-8282
e-mail/courriel : andre.mainville@nrcan.gc.ca
****