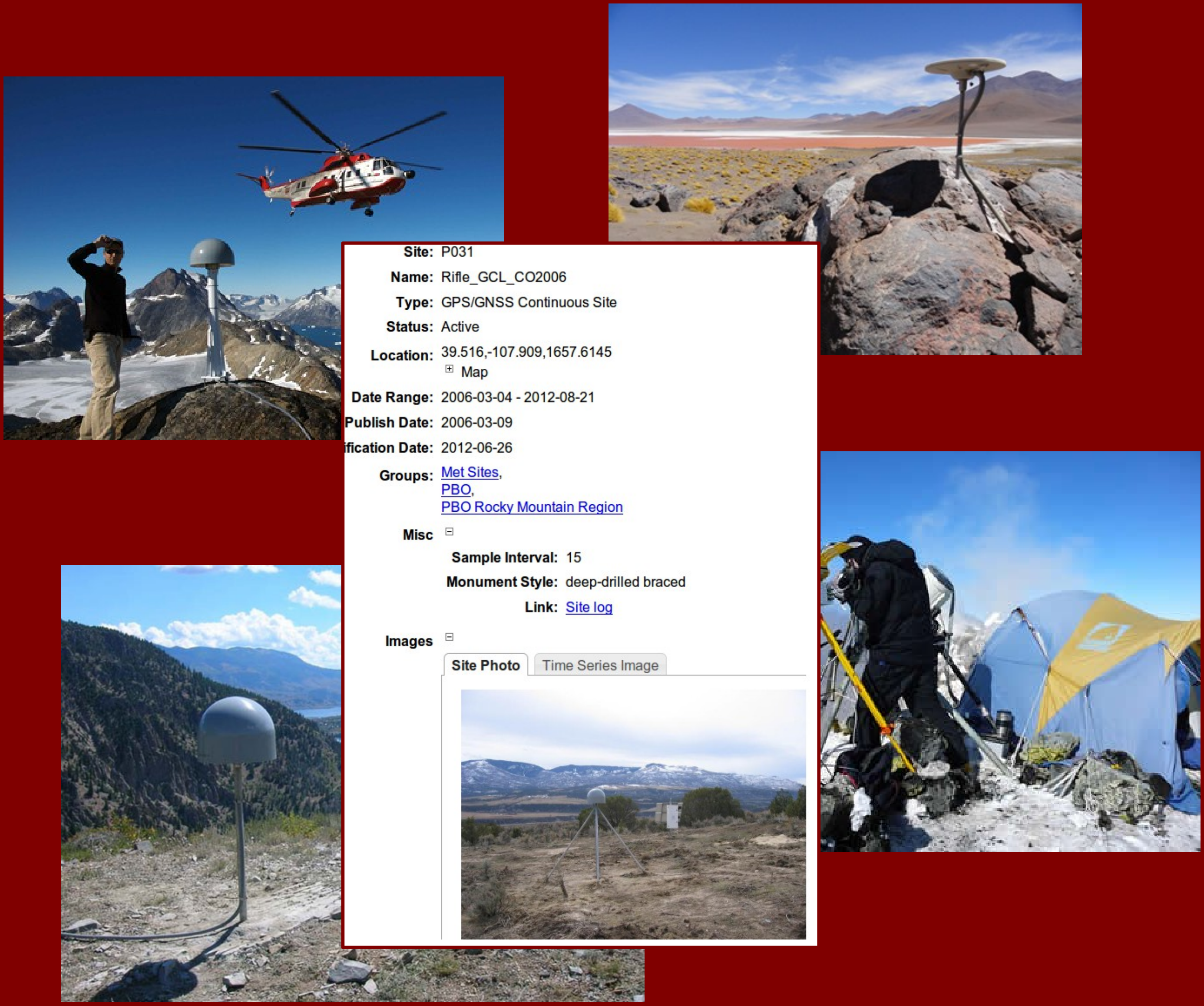
July 2015

UNAVCO GSAC:

User Guide for GSAC Data Repositories



Site: P031
Name: Rifle_GCL_CO2006
Type: GPS/GNSS Continuous Site
Status: Active
Location: 39.516,-107.909,1657.6145
⊞ Map
Date Range: 2006-03-04 - 2012-08-21
Publish Date: 2006-03-09
ification Date: 2012-06-26
Groups: Met Sites,
PBO,
PBO Rocky Mountain Region
Misc ⊟
Sample Interval: 15
Monument Style: deep-drilled braced
Link: Site log
Images ⊟

UNAVCO

User Guide for
GSAC Geoscience Data Repositories

July 2015

UNAVCO

Boulder Colorado

User Guide for GSAC Geoscience Data Repositories

Stuart Wier

UNAVCO
6350 Nautilus Drive
Boulder, Colorado U.S.A.
http://www.unavco.org

July 22, 2015

UNAVCO: http://www.unavco.org/

The GSAC web site is at www.unavco.org/software/data-management/gsac/gsac.html

Cover photographs from UNAVCO:
Hjornefjedet, Greenland. GNET 2007
Uturuncu volcano, Bolivia. PLUTONS Project Part II
Green Mountain Reservoir, Colorado. Monument RG16, Rio Grande Rift Network
Mt. Erebus, Antarctica. LIDaR station G-081, 2009

GSAC was a NASA ROSES ACCESS Program funded project (2010-2012), through the ROSES ACCESS program Cooperative Agreement NNX10AF07A, developed at UNAVCO, NASA CDDIS, and SOPAC UCSD. In 2012 and 2013 NSF funded GSAC development and support at UNAVCO in support of COOPEUS (2012-2015).

CDDIS: http://cddis.nasa.gov/

SOPAC: http://sopac.ucsd.edu/

Contents

1. GSAC Overview

GSAC is a software package of web services. GSAC provides search and download of data from Earth science data centers. Instrumented sites or stations make data files of measurements. GSAC provides searches to find daya you need, and ways to download the data. Searches use common criteria, including site location or geographic area, site names, date s, and instrument type. Data can be selected based on instrument type, data type, and data file format. Data files may be raw or reformatted observations, derived products, plot, and other related files.

GSAC provides a web browser user interface for exploring the contents of a data repository, an also an API for computerized reuquests and file downloads with your own command line requests to GSAC, or with your own scripts. There is a GSAC client program for programmatic command line interaction with GSAC server.

GSAC facilitates search, discovery, and retrieval of data from geoscience data centers. GSAC is based on the concept of scientific instruments instruments at Earth-located stations (sites, or "monuments") making data files of Earth science measurements. GSAC software supports single data repositories, and there is also a federated GSAC mode for joint search and download from several cooperating GSAC-enabled data centers.

To begin to learn GSAC, you may go directly to section 3, The Browser User Interface.

You can use GSAC to make a first exploration of a data center's holdings, or use it to download information about thousands of GNSS data files in one command line query. A GSAC implementation publishes on its web site its repository information in XML, a capabilities document about API interactions. This information specifies how to make API-based or RESTful queries to a GSAC repository using any coding system you like.

Like other web services, GSAC accepts incoming URL-based requests, handles requests, and returns results. The format of results from GSAC is selectable by the user, including HTML (web pages), CSV files, KMZ for Google Earth, and JSON. GSAC can create results in the geodesy formats SINEX, GAMIT station.info, "SOPAC XML site logs," and more. For data file and product downloads, GSAC returns URLs where data files can be downloaded from FTP or HTTP file servers.

You can search and download information from a GSAC server, using Linux commands on a command line or with the GSAC command line client, a Java-based client for accessing a GSAC repository, another way to do programmatic searches of a GSAC repository and to download data files without use of web forms.

Searches and downloads of instrumental data from many geodesy and geoscience data centers use the same or similar functions, such as site-name based queries, geographic location queries, and time queries. The GSAC package is for these common needs, and also has an ability to tailor details specific to a particular repository. Because there are different details about data repositories, the searches for sites and files may differ a little between repositories. The standard GSAC implementations are intended to have consistent interactions.

For complete information about GSAC, you can always Google "UNAVCO GSAC" to find the UNAVCO GSAC web site at http://www.unavco.org/software/data-management/gsac/gsac.html.

To install GSAC for your data center, see the Installation page at the UNAVCO GSAC web site,
http://www.unavco.org/software/data-management/gsac/installation/installation.html.

2. GSAC Design Principles

GSAC provides web services for data search and access from a data repository. GSAC code is middleware between a repository's database and file system on one hand, and the public on the other, allowing remote users to query metadata about instrumentation and sites (stations), or to access instrument data files and products.

GSAC is based on the concept of scientific instruments instruments at stations (sites, or geodesy monuments) making data files of Earth science measurements. GSAC is intended to be generalized.

A station or site is characterized with metadata including a four-character ID code, station name, latitude, longitude, ellipsoidal height, station installation date, country, province or state, place or city, and network name(s). GNSS stations can record the monument description, geodetic position (x, y, z), and the IERS DOMES ID value. You may include a URL to a station photograph, and an image of a time series data plot, such as the position time series for a GNSS station. Stations are usually identified with their four-character ID code, but due to duplicate codes in separate networks, and because some stations record multiple signal types, only the combination of four character ID code with station name may be unique.

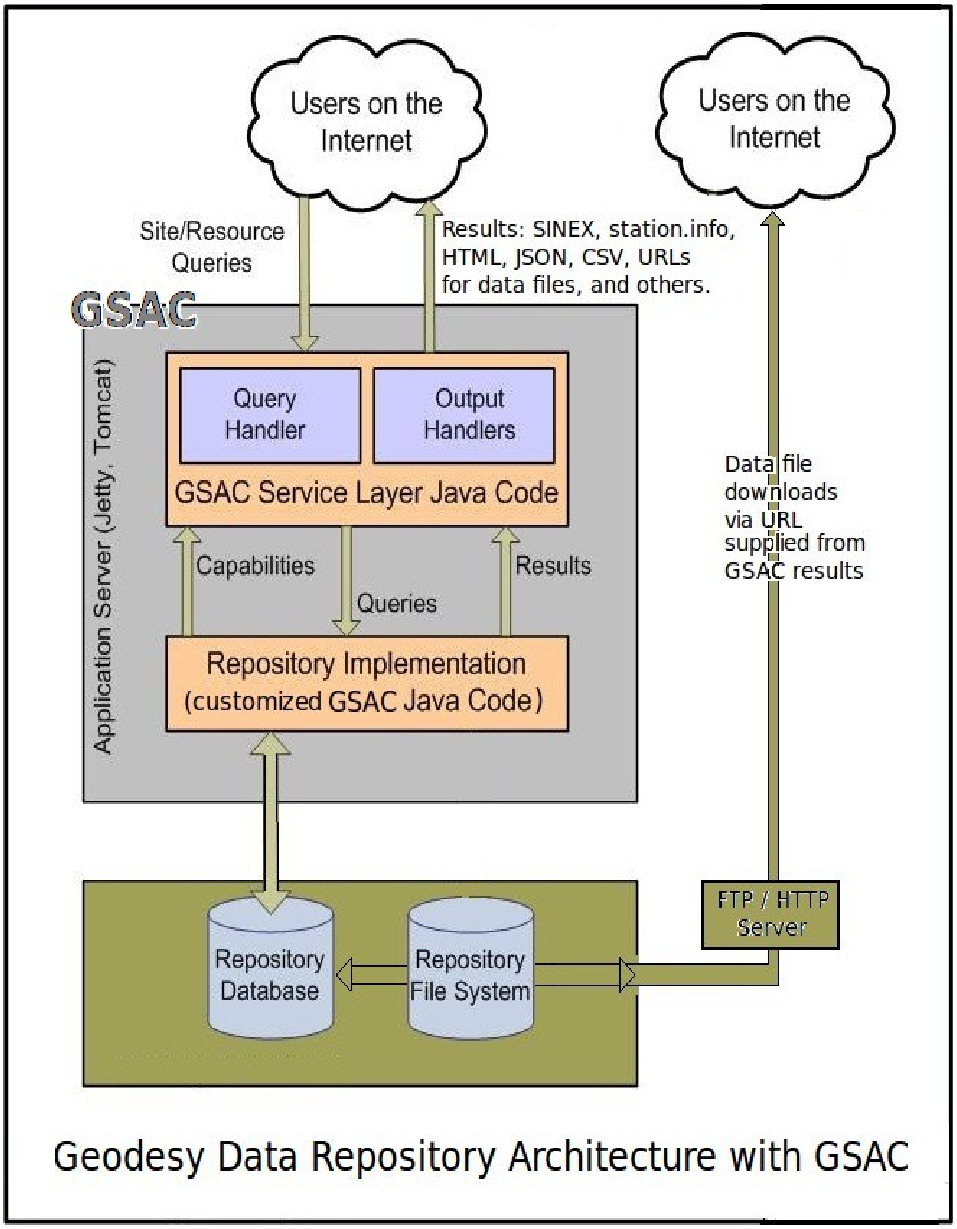
Instruments are described with GPS terminology, including antennas and receivers. In some other cases these parameters could be used for other data types, for example the parameters "antenna name" and "antenna serial number" can be used for other types of sensors, and "receiver" parameters can be used for other types of data loggers. Installed and removed dates have the same meaning in all situations. The receiver may have an associated firmware version. GPS stations may record the antenna radome type and its serial number.

GSAC can handle most any type of instrumental data file or product file. The standard GSAC includes more than 25 file types, such as 'instrument raw binary file,' 'RINEX GPS navigation file,' and 'Tiltmeter Plot.' It is very easy to add new file types to GSAC. An operator of a GSAC simply adds a new file type to the database (one new row), and labels each file entry in the database with the new type.

A useful concept is the *equipment session*, a time interval at one station when the instrumentation does not change so that the instrumental data files during that time interval from that station are consistent. GSAC provides complete equipment session information for each station (if the data is available to GSAC).

Information about instrumental data files or product files indicates the file type (via a database key or id number to the file type table), station (via a database key or id number to the station table), data start time, data end time, and the URL to a FTP or HTTP file server where the file can be downloaded. Given the data times, the corresponding equipment session information for that time at that station completely describes how the data was collected. Clearly GSAC can handle almost any kind of file from a station, or related to data from a station such as derived products.

A user may query a GSAC installation using forms on its web pages, or programmatically with HTTP (RESTful) formatted queries, or with direct command line queries on Linux or Mac OS, or by use of the GSAC client program. Results returned may may be in any of several formats specified in the query. Data file downloads are made from GSAC repositories which support FTP or HTTP file access.


Users on the Internet
Users on the Internet
Site/Resource Queries
Results: SINEX, station.info, HTML, JSON, CSV, URLs for data files, and others.
GSAC
Application Server (Jetty, Tomcat)
Query Handler
Output Handlers
GSAC Service Layer Java Code
Capabilities
Queries
Results
Repository Implementation (customized GSAC Java Code)
Data file downloads via URL supplied from GSAC results
FTP / HTTP Server
Repository Database
Repository File System
Geodesy Data Repository Architecture with GSAC

The GSAC database uses the best data types or formats, to help avoid errors in archives and processing. Numerical values are handled as decimal values; for example latitudes are only in decimal degrees. Characters are never used to represent numerical data. Dates and times are in the ISO 8601 format. Note that some types of so-called "numbers," such as "Radome serial number" often have characters in them, and are not numerical values, so they must be represented as characters.

The core GSAC code (sometimes called "GSAC Service Layer" or GSL) handles incoming requests, and interprets the queries; it passes the results of the queries to the output handlers (formatters). Queries to GSAC specify both search details, and the format desired for the returned data. Query options are available through web page forms in the browser, and the same options are also available through the GSAC API. GSAC has an XML document of query capabilities to applications calling GSAC, which you can also see on the repository web site (this is how a federated GSAC knows how to call other GSACs). All GSAC installations use the same core GSAC code.

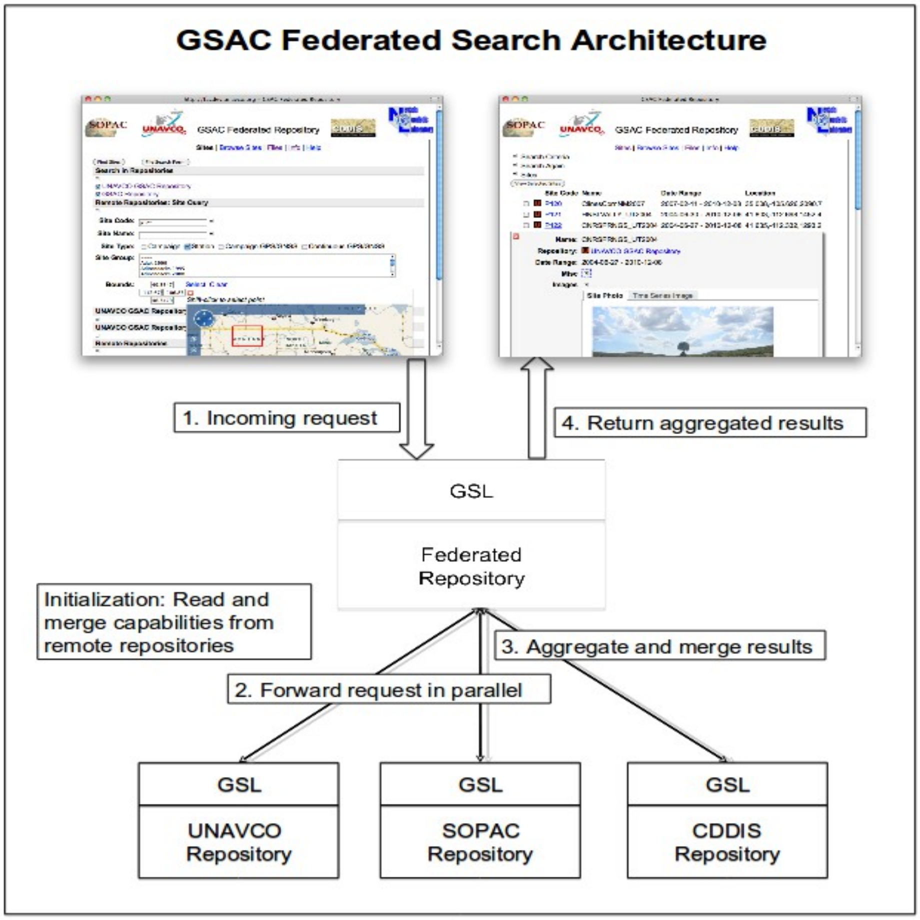
On its web site, each GSAC repository has detailed search criteria to identify instrumented stations (sites, monuments), including names, where instruments are located, and instrument type, and to identify data files produced by the instruments by data type and data file format. GSAC provides geographic and temporal constraints in searches. For station information, or for files that are specific to a site or group of sites, you can specify additional site criteria, such as station network name or geographic location or region. The GSAC browser user interface places all the search options on one page.

Every GSAC repository has an API for computerized reuquests and file downloads with your own command line requests or with your own scripts. The API is used for automated retrieval of stations' information or of many data files. Most "power" GSAC users use the API for routine or large downloads. (GSAC also provides a separate client program to programmatically query any GSAC repository and get results directly to your terminal to to files. The client program runs on any system with recent Java support; see section 5).

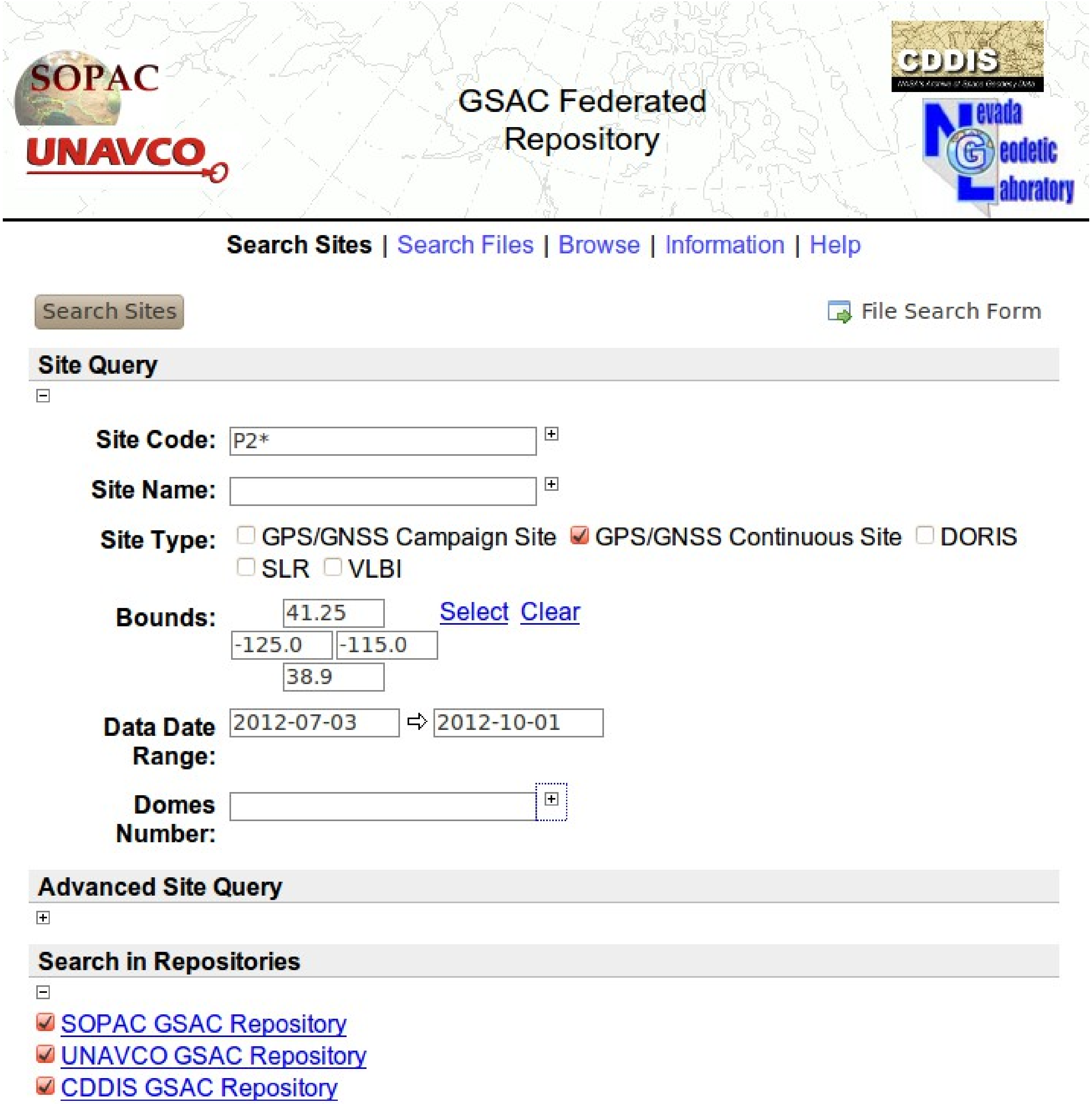
Federated GSAC

GSAC itself is not a file download server: GSAC provides URLs where you can download data files. One GSAC can serve mutilple download servers, which may or may not be located in the same hardware, or at the same agency, as GSAC itself. This allows "federated" GSACs, which unified provide data discovery and download URLs from two or more separate GSACs in several locations.

A single GSAC server can run as a federated search and file retrieval server, acting as a single web service for joint searches across several other GSAC-enabled repositories, offering the power of merging many data archives into one search and retrieval tool. A federated GSAC web service does not copy station metadata, or copy data files from remote servers, but rather knows how to query remote GSACs for information they hold. Federated searches are made possible by a GSAC web service being able to jointly access and use two or more other GSAC repositories, by means of the standard GSAC capabilities at those servers (figure by Jeff McWhirter).


GSAC Federated Search Architecture
1. Incoming request
4. Return aggregated results
GSL
Federated Repository
Initialization: Read and merge capabilities from remote repositories
3. Aggregate and merge results
2. Forward request in parallel
GSL
UNAVCO Repository
GSL
SOPAC Repository
GSL
CDDIS Repository

The picture below is from UNAVCO's Federated GSAC Repository, the site search page, showing the ability to simultaneously search for sites on one to three GSAC repositories (CDDIS, SOPAC, and UNAVCO), and the ability to search on any or all of the data types (GPS, DORIS, SLR, and VLBI) offered by those repositories. Note that not all of the repositories all the data types; a federated GSAC can join GSACs which have some differences.


SOPAC
UNAVCO
GSAC Federated Repository
CDDIS
Nevada Geodetic Laboratory
Search Sites | Search Files | Browse | Information | Help
Search Sites
File Search Form

Site Query

Site Code: P2*

Site Name:

Site Type: □ GPS/GNSS Campaign Site ☑ GPS/GNSS Continuous Site □ DORIS □ SLR □ VLBI

Bounds: 41.25 / -125.0 / -115.0 / 38.9 Select Clear

Data Date Range: 2012-07-03 ⇨ 2012-10-01

Domes Number:

Advanced Site Query

Search in Repositories

☑ SOPAC GSAC Repository
☑ UNAVCO GSAC Repository
☑ CDDIS GSAC Repository

3. The Browser User Interface

GSAC provides forms on web pages to search and retrieve information and data files from a data repository. Use GSAC web page tools to start learning about GSAC, and for finding site information and data from a GSAC archive.

Most of the power of GSAC services can be achieved by the use of web page forms. GSAC requests can also be made with programmatic interactions, command line requests or computer program actions. Learning GSAC firstwith the web pages gives a good understanding of what a GSAC can do. You learn the concepts about what GSAC provides; later you can decide which way to call GSAC is best for your needs. Also the web page results show the GSAC API command options to make the same query via a computer request direct to that GSAC service, so you can move from clicking to specify and make one request to creating computerized requests which can be automated or repeated. For access to large numbers of results, such as information about hundreds of stations, or to download hundreds of files, use of GSAC by programmatic requests is appropriate (see section 4, the GSAC API).

This section shows examples from using web interactions with typical GSAC implementations for GNSS sites and data files. Use any GSAC online to learn about GSAC. See the UNAVCO GSAC web site (http://www.unavco.org/software/data-management/gsac/repositories/repositories.html) to find a list of active GSAC data repositories.

The top of each GSAC web page has this list of links to key sections of the GSAC web site:

Search Sites | Search Files | Browse | Information | Help

Click on one of these to

search for information about sites (stations or monuments) and about instruments at the sites

find and download instrument data files from the sites, such as RINEX files and product files

browse for information by a single item

see information about this particular GSAC server's capabilities (API)

see a synopsis of help using GSAC.

Search items in GSAC pages (and in the API) are choices which limit or restrict *Searching*. No choice means no restrictions; it does not mean "do nothing." If you click on Search Sites alone, in a **Search Sites** page, you will see all the sites in a GSAC data repository, which may be thousands in some cases. If you click on Search Files alone, in a **Search Files** page, that GSAC may attempt to find and return information for more than 400 000 files in some cases, and it probably will not return a result very quickly.

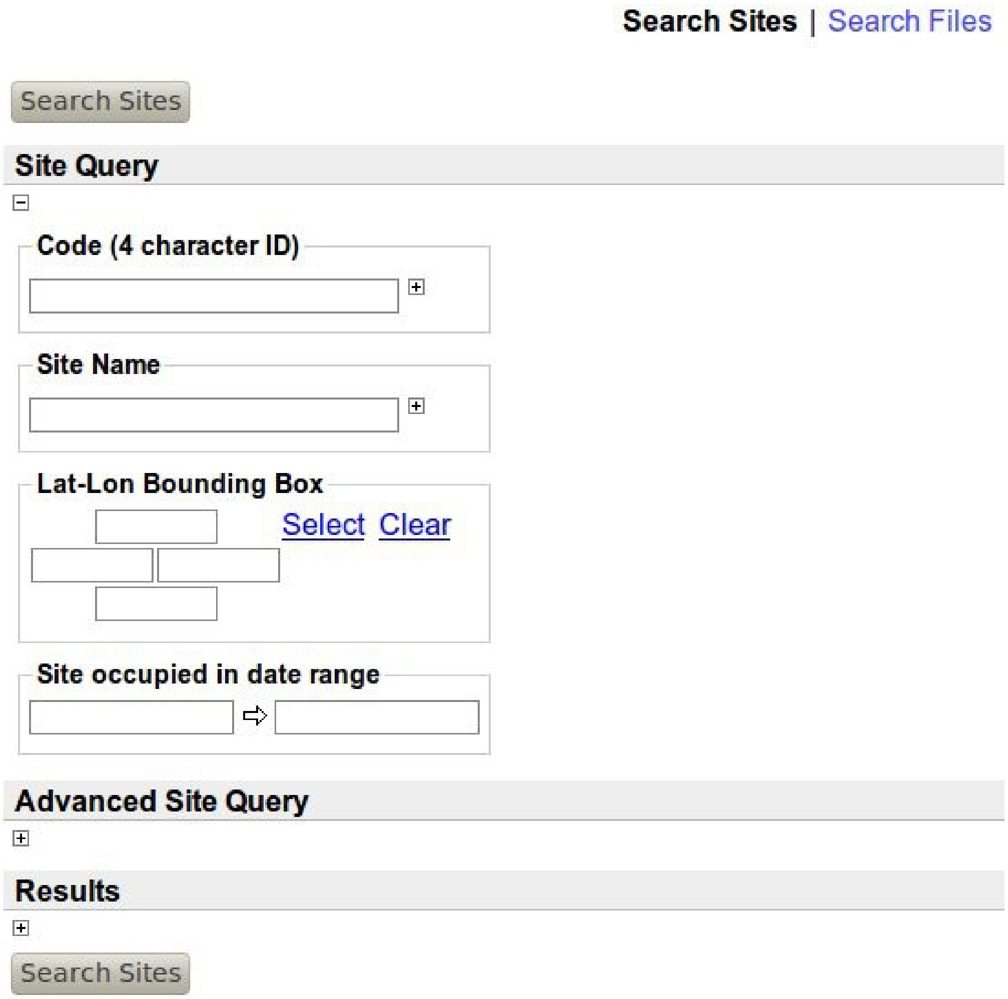
Search items do not modify the information in results, once the sites or files have been found by searches with site or file choices. If your search finds a site, the results (site information) is only determined by the output type of format. Search choices do not modify what is in the output, except for choosing the site for file to make output for.

3.1 Search Sites

GSAC is built on the concept of instrumented stations at known locations, with unique site IDs, 4 character labels, (or with unique combinations of site IDs and site names), with antennas or sensors, and with receivers or data loggers, making science data files. Times are assigned to stations' operations and to data collections.

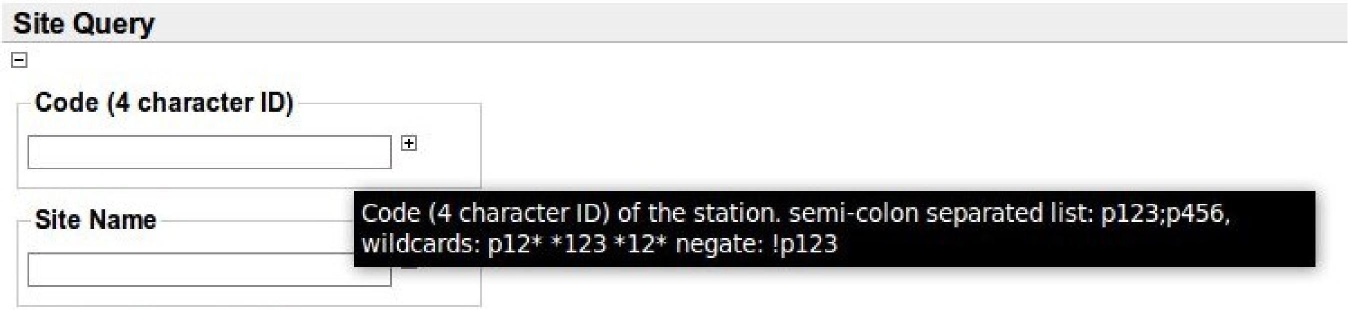
The GSAC **Search Sites** web page has entry forms to query site metadata to find sites that match. A "site" is a station or a geodesy "monument," with instruments at a fixed latitude-longitude-height location. You can search for sites by name, in a latitude-longitude bounding box, with dates when the sites operated, and with other choices, such as instrument types.

When you first click on the link Search Sites, you see a new web page with the site search choices:



If you click on the "Search Sites" button above **Site Query** (not the link with the same name at the top of the page) you will search for *all* sites in this repository (caution: the UNAVCO and SOPAC GSACs have thousands of sites).

Enter in **Code** (4 character ID) a site ID like KATC, or enter in **Site Name** a complete name like Katavia, you will limit the search for sites in the repository with those site values. The site name must exactly match the name in the database read by the GSAC in use.



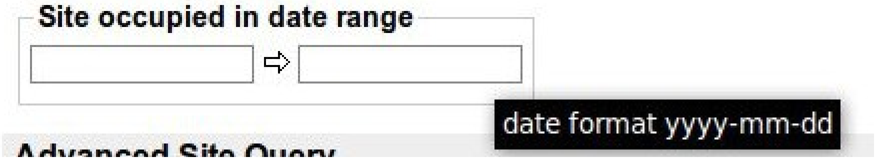
Code (4 character ID) of the station. semi-colon separated list: p123;p456, wildcards: p12* *123 *12* negate: !p123

You can see accepted input forms; for example for **Code**, hover the cursor over the Code entry box to see this help:

showing that you can search with a list of site Code-s separated with ";", or you can search for Codes with wildcards like P2*, or you can exclude a site Code from a search, as with !P123. Do not include quotes "" or ' ' in any search items. Commas between Code-s will result in no sites found; use ";".

To limit a search to sites in a geographic region use the Lat-Lon Bounding Box group of four value boxes. The top and bottom boxes are for the most northerly and most southerly latitudes. The left and right boxes are for the most westerly and most easterly longitude boxes. You can use a minus sign in the boxes. Do not enter characters in the boxes, such as north, S, or W. Theses boxes are for numerical entries only. GSAC does not use or recognize geographic positions in the troublesome and highly error-prone formats of degrees, minutes, and seconds. We admire the outstanding navigators of old, such as James Cook, William Clark, Louis-Antoine de Bougainville, and Alessandro Malaspina, but do not use their technologies.

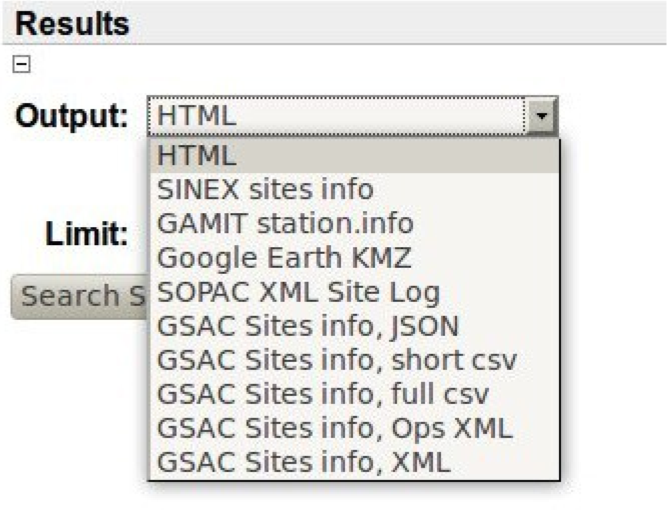
The station search by date range ("**Site occupied in date range**") is for *when instruments were present at the location*. This pair of boxes has this help when you do a cursor hover



date format yyyy-mm-dd

showing formats for time values. Alternately, clicking in any date entry box in GSAC shows a calendar you can use to choose date values.

The site occupied date range need not be entirely occupied by instruments; any time in the range will do. The station of course may have had instruments at other times as well, outside this date range. The instruments may not have been operational in this range. *This is not a search for data availability*; it is for site occupation. *There may be no data files from this site in this data range.* Also, the site search date range choice *does not control what may or may not be included in any of the several GSAC searvch results' output formats.*



Output: HTML

The **Results** area for Search Sites (click the [+] box under Results) shows an **Output:** option box with the choice of output formats for GSAC site query results. Some of these are standard formats; others are formats from GSAC, requested by GSAC users. Your site search results will be sent to the browser making the request, in the format chosen.

The site search output formats are:

HTML provides a GSAC web page of the results, which is useful for browsing a GSAC server. HTML is the default GSAC output format, used if you do not chose some other format.

The SINEX format for GPS sites is described in the IERS SINEX Format page,

http://www.iers.org/IERS/EN/Organization/AnalysisCoordinator/SinexFormat/sinex.html

The GAMIT station.info file format for GPS sites is described in the GAMIT documents page,

http://www-gpsg.mit.edu/~simon/gtgk/docs.htm

The Google Earth KMZ file shows station locations and related information in Google Earth.

The SOPAC XML Site Log format is described in

http://sopac.ucsd.edu/xmlGeodesy.shtml

GSAC Sites info, JSON files for sites show site name, location, and time of use. JSON is an IT format, intended to improve on XML. It is for reading by computer processing. GSAC Sites info, XML is the original (2011) XML site output from GSAC (deprecated in favor of GSAC Sites info, Ops XML, below).

Comma-separated-value files (CSV) are a simple format, using ASCII or UTF-8 characters. CSV files can be read by humans, and also can be easily interpreted by computer scripts. UNAVCO uses the "Standard CSV Format" for station data (see http://www.unavco.org/data/gsacws/docs/ UNAVCO_standard_CSV_format.html). This format has been perfected over several years for "point data", observations at fixed locations on or near the Earth. It is self-describing, with locations, names of data types, formats for each value, and units in some cases.

The GSAC Sites info, short csv file format has values only for a site's 4-character ID, its name, and its latitude, longitude and ellipsoid height.

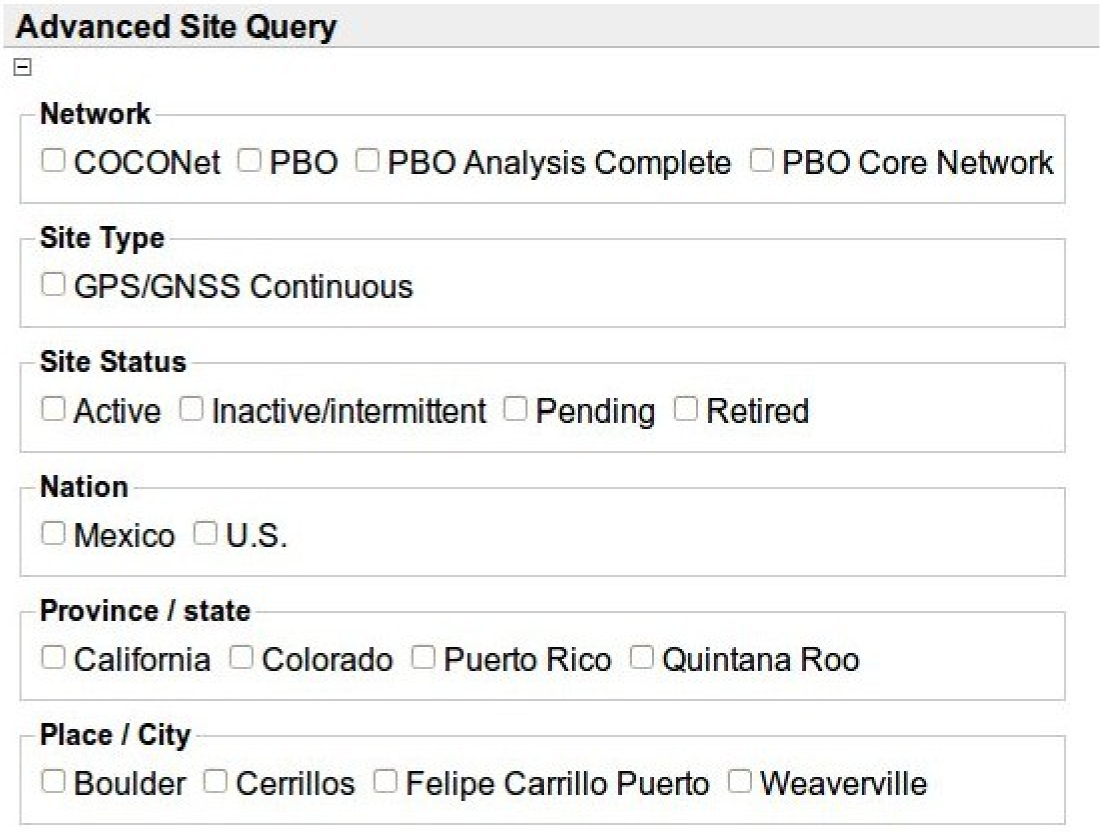
The GSAC Sites info, full csv file format has more information about stations and their instruments. Each line is for one "equipment session" at one station, a time interval when the instrumentation does not change so that the instrument data files during that interval from that station are consistent.

GSAC Sites info, Ops XML format is a UNAVCO GSAC format for site information in XML. It was designed by UNAVCO field engineers.

To see examples of GSAC output formats, use a GSAC server online to search sites.

The **Results** area also has a **Limit** box you can use to limit how many items or lines are returned from any search. The default value, shown in the box, is 500. If you expect to find more than 500 items for your query, entry a value larger than 500.

The **Advanced Site Query** area for Search Sites (click the **[+]** box to open it) has more choices to refine site searches:



Advanced Site Query

⊟

Network

□ COCONet □ PBO □ PBO Analysis Complete □ PBO Core Network

Site Type

□ GPS/GNSS Continuous

Site Status

□ Active □ Inactive/intermittent □ Pending □ Retired

Nation

□ Mexico □ U.S.

Province / state

□ California □ Colorado □ Puerto Rico □ Quintana Roo

Place / City

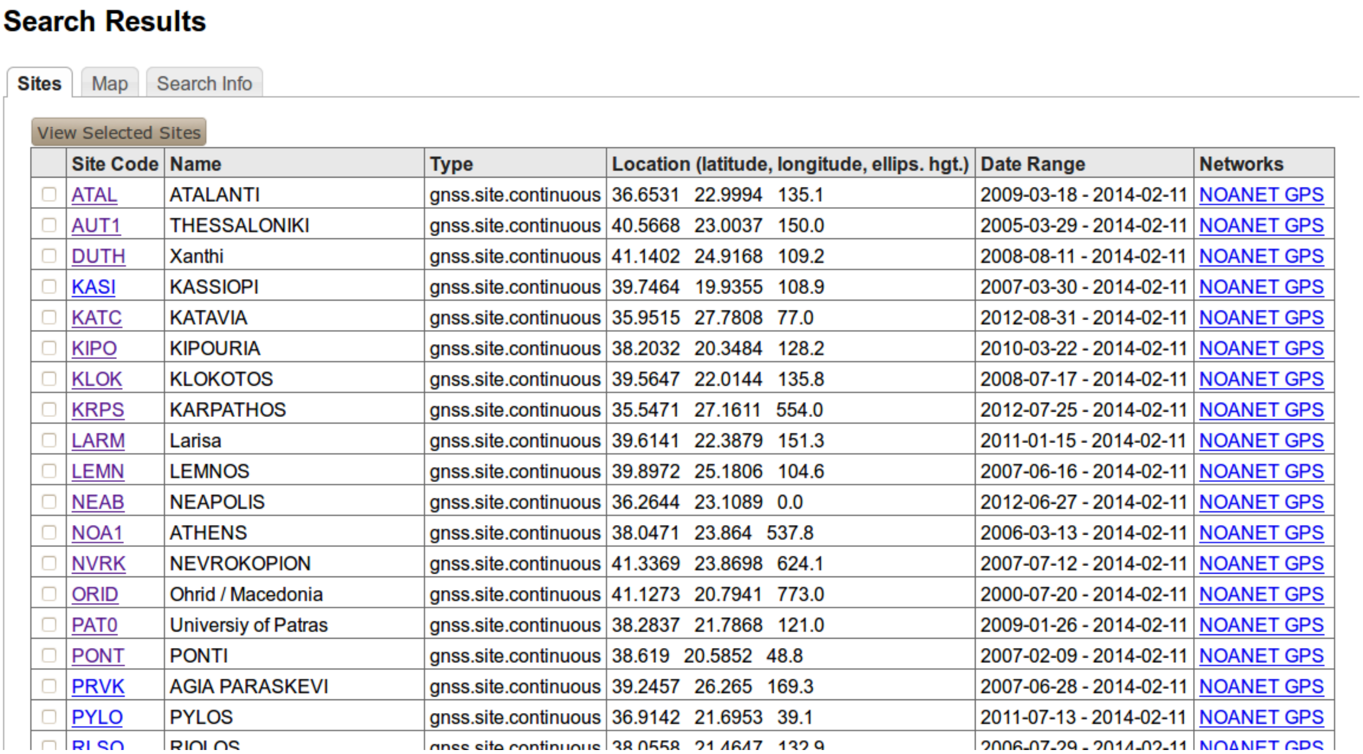
□ Boulder □ Cerrillos □ Felipe Carrillo Puerto □ Weaverville

Clicking on any of the Advanced Site Query items will add a new limitation to search for sites. You can for example in some GSACs search for receiver type, and add more selection items at the same time.

Next is shown an example of search results in HTML format, a table in the browser. A successful site search returns a table with one or more stations. The field or column Date Range (labeled Occupied Dates in some GSACs) is the full time period the site was occupied; when it had instruments installed; this time interval may have no data files at some times. These results are from the NOANET GSAC of the National Observatory of Athens.

In 2015 UNAVCO plans to add a new field, Latest Data Time, to the site search results table, to provide some indication when data files are available.

Many single GPS sites have data in several discontinuous time intervals. Use Search Files to best investigate data availability at a site.



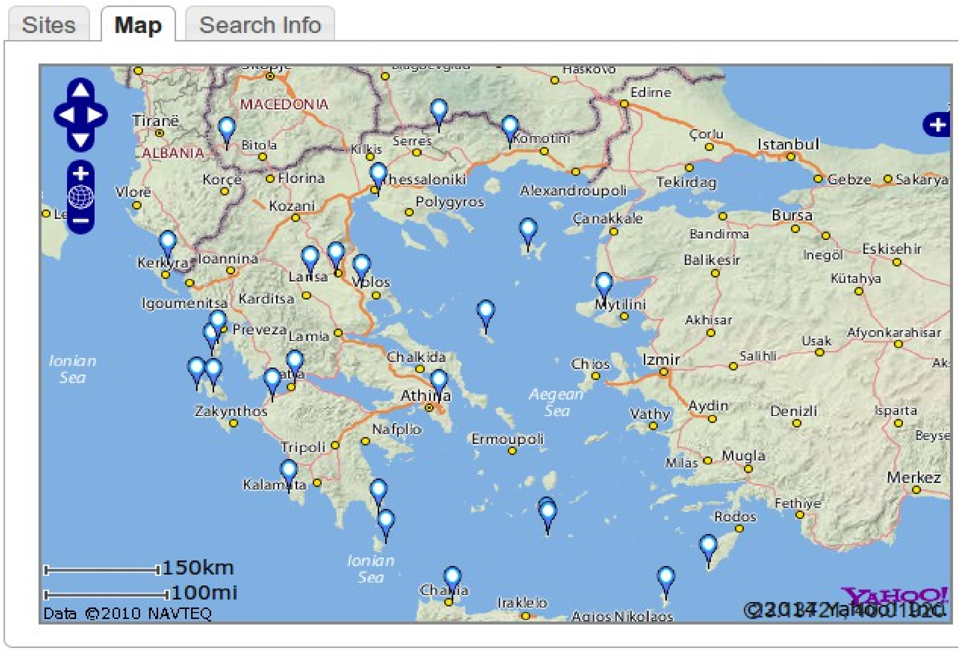
Search Results

	Site Code	Name	Type	Location (latitude, longitude, ellips. hgt.)	Date Range	Networks
□	ATAL	ATALANTI	gnss.site.continuous	36.6531 22.9994 135.1	2009-03-18 - 2014-02-11	NOANET GPS
□	AUT1	THESSALONIKI	gnss.site.continuous	40.5668 23.0037 150.0	2005-03-29 - 2014-02-11	NOANET GPS
□	DUTH	Xanthi	gnss.site.continuous	41.1402 24.9168 109.2	2008-08-11 - 2014-02-11	NOANET GPS
□	KASI	KASSIOPI	gnss.site.continuous	39.7464 19.9355 108.9	2007-03-30 - 2014-02-11	NOANET GPS
□	KATC	KATAVIA	gnss.site.continuous	35.9515 27.7808 77.0	2012-08-31 - 2014-02-11	NOANET GPS
□	KIPO	KIPOURIA	gnss.site.continuous	38.2032 20.3484 128.2	2010-03-22 - 2014-02-11	NOANET GPS
□	KLOK	KLOKOTOS	gnss.site.continuous	39.5647 22.0144 135.8	2008-07-17 - 2014-02-11	NOANET GPS
□	KRPS	KARPATHOS	gnss.site.continuous	35.5471 27.1611 554.0	2012-07-25 - 2014-02-11	NOANET GPS
□	LARM	Larisa	gnss.site.continuous	39.6141 22.3879 151.3	2011-01-15 - 2014-02-11	NOANET GPS
□	LEMN	LEMNOS	gnss.site.continuous	39.8972 25.1806 104.6	2007-06-16 - 2014-02-11	NOANET GPS
□	NEAB	NEAPOLIS	gnss.site.continuous	36.2644 23.1089 0.0	2012-06-27 - 2014-02-11	NOANET GPS
□	NOA1	ATHENS	gnss.site.continuous	38.0471 23.864 537.8	2006-03-13 - 2014-02-11	NOANET GPS
□	NVRK	NEVROKOPION	gnss.site.continuous	41.3369 23.8698 624.1	2007-07-12 - 2014-02-11	NOANET GPS
□	ORID	Ohrid / Macedonia	gnss.site.continuous	41.1273 20.7941 773.0	2000-07-20 - 2014-02-11	NOANET GPS
□	PAT0	Universiy of Patras	gnss.site.continuous	38.2837 21.7868 121.0	2009-01-26 - 2014-02-11	NOANET GPS
□	PONT	PONTI	gnss.site.continuous	38.619 20.5852 48.8	2007-02-09 - 2014-02-11	NOANET GPS
□	PRVK	AGIA PARASKEVI	gnss.site.continuous	39.2457 26.265 169.3	2007-06-28 - 2014-02-11	NOANET GPS
□	PYLO	PYLOS	gnss.site.continuous	36.9142 21.6953 39.1	2011-07-13 - 2014-02-11	NOANET GPS
□	RLSO	RIOLOS	gnss.site.continuous	38.0558 21.4647 132.9	2006-07-29 - 2014-02-11	NOANET GPS

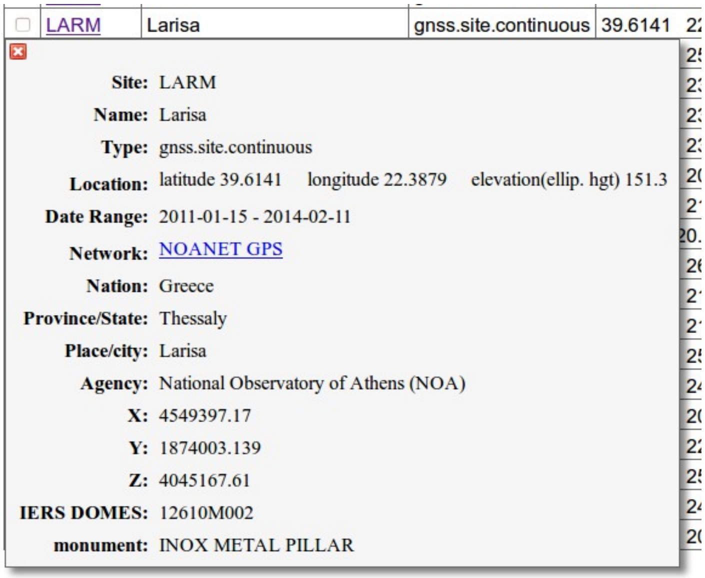
Clicking on a name in the Networks column in the table makes *a completely new site search* for *all the stations in that network*; *not* limited by your original search choices. Some networks have hundreds of sites.

Click on the **Map** tab to see a map with all the stations in the table:

If you click on a station location marker on the map, you will see a pop-up box with information about that station.



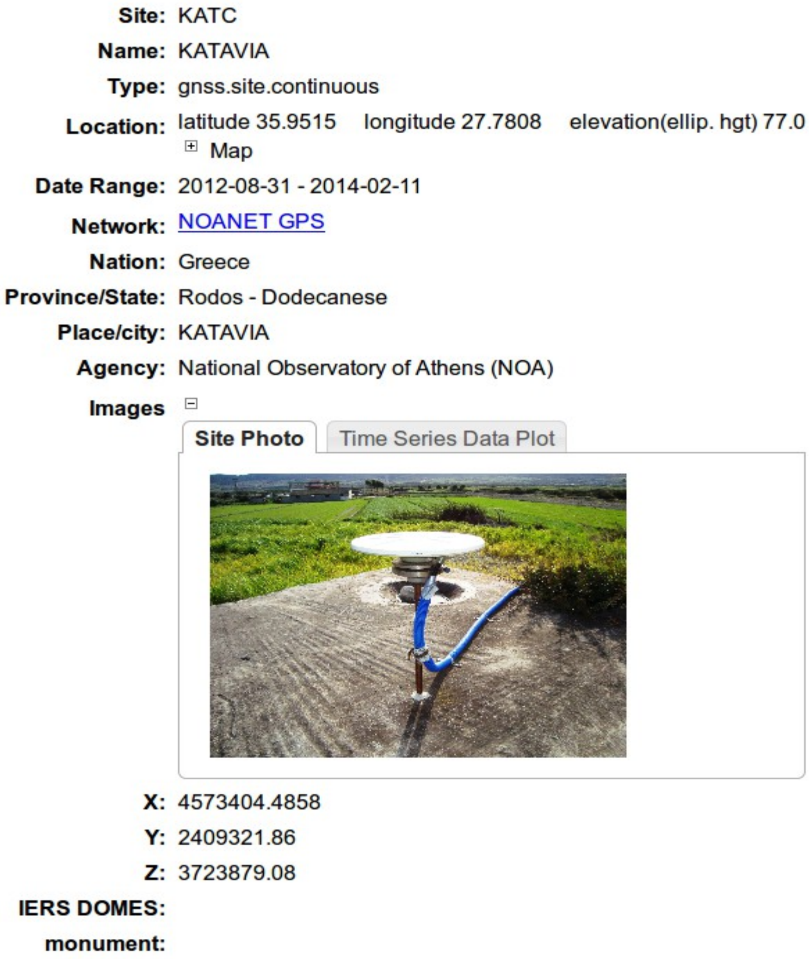
Click anywhere in a row in the table of search results (except a Site Code name) to see a pop-up box with information about the site, like this:



□	LARM	Larisa	gnss.site.continuous	39.6141

Site: LARM
Name: Larisa
Type: gnss.site.continuous
Location: latitude 39.6141 longitude 22.3879 elevation(ellip. hgt) 151.3
Date Range: 2011-01-15 - 2014-02-11
Network: NOANET GPS
Nation: Greece
Province/State: Thessaly
Place/city: Larisa
Agency: National Observatory of Athens (NOA)
X: 4549397.17
Y: 1874003.139
Z: 4045167.61
IERS DOMES: 12610M002
monument: INOX METAL PILLAR

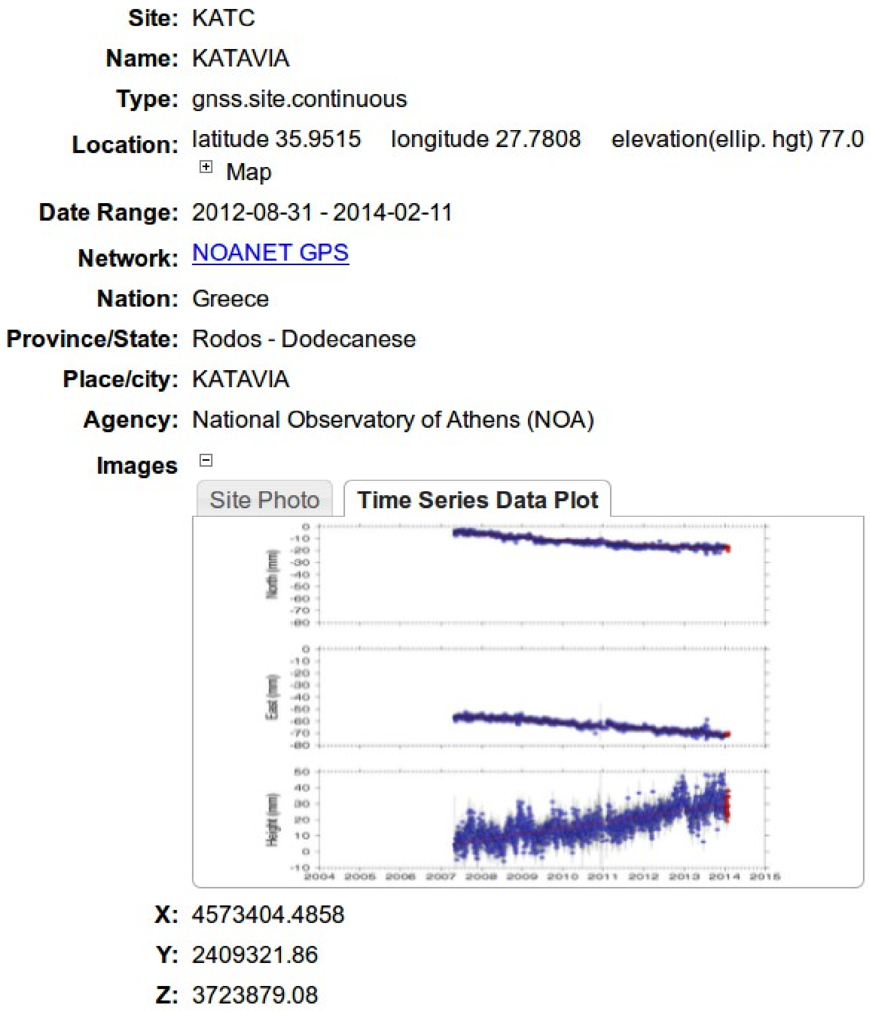
Click on a Site Code name in the table of search results to see a new web page about that one station, like this:



Site: KATC
Name: KATAVIA
Type: gnss.site.continuous
Location: latitude 35.9515 longitude 27.7808 elevation(ellip. hgt) 77.0
⊞ Map
Date Range: 2012-08-31 - 2014-02-11
Network: NOANET GPS
Nation: Greece
Province/State: Rodos - Dodecanese
Place/city: KATAVIA
Agency: National Observatory of Athens (NOA)
Images ⊟

X: 4573404.4858
Y: 2409321.86
Z: 3723879.08
IERS DOMES:
monument:

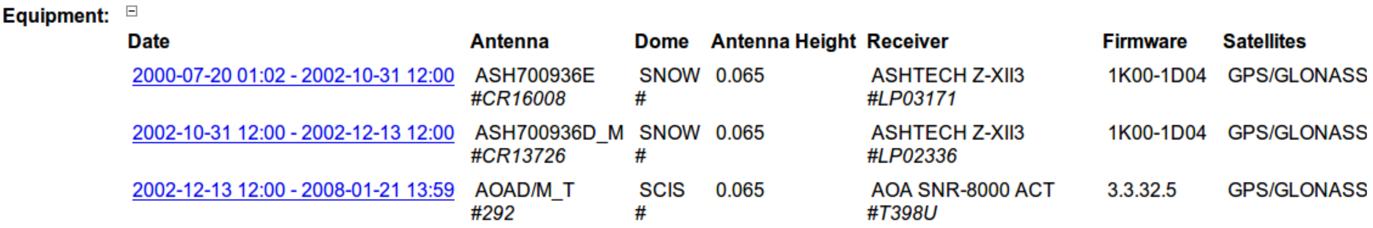
What is shown in these site information presentations depends in part on how the GSAC was implemented at each data repository. Not all GSACs show the same information about sites. The first view of this page may show a station location photograph, if one is available. There may be a tab Time Series Data Plot with an image of time series data, is there is one:



Site:	KATC
Name:	KATAVIA
Type:	gnss.site.continuous
Location:	latitude 35.9515 longitude 27.7808 elevation(ellip. hgt) 77.0 ⊞ Map
Date Range:	2012-08-31 - 2014-02-11
Network:	NOANET GPS
Nation:	Greece
Province/State:	Rodos - Dodecanese
Place/city:	KATAVIA
Agency:	National Observatory of Athens (NOA)
Images	⊟
X:	4573404.4858
Y:	2409321.86
Z:	3723879.08

Click on [+] Map to see a map with a marker at the station location. Click on the station location marker on the map to see a popbox box with information about that station. Click the station name in the box to see this full web page. Note *for this* GSAC the site information has the IERS DOMES value, and the geodetic position values (x,y,z), if available. Not all GSACs have this information. GSACs are made to serce their data holdings.

Click on Equipment: [+] to open a list of the site's equipment sessions, time intervals when no change was made to the instrumentation at that site:



Equipment: ⊟

Date	Antenna	Dome	Antenna Height	Receiver	Firmware	Satellites
2000-07-20 01:02 - 2002-10-31 12:00	ASH700936E *#CR16008*	SNOW #	0.065	ASHTECH Z-XII3 *#LP03171*	1K00-1D04	GPS/GLONASS
2002-10-31 12:00 - 2002-12-13 12:00	ASH700936D_M *#CR13726*	SNOW #	0.065	ASHTECH Z-XII3 *#LP02336*	1K00-1D04	GPS/GLONASS
2002-12-13 12:00 - 2008-01-21 13:59	AOAD/M_T *#292*	SCIS #	0.065	AOA SNR-8000 ACT *#T398U*	3.3.32.5	GPS/GLONASS

Go back to the table of site search results. If you click on the Search Info tab at the top, a list of the Results Output formats appears at the bottom of a normal site search form (and your original search selections are preserved). Click on any of the Search Links or Result Output Formats to redo the original search, or to make a new kind of search of choose a new output format.

To learn how to compose the GSAC API request for site searches, use the GSAC's Search Site page make a request like you need. Use the default HTML output format. When the search finds the sites you want, in the table of sites found, click on the **API Request** tag (in any GSAC version after December 2104). This shows the API equivalent to the request you made with the web page choices. Merge this with the GSACs base URL (see the URL in the browser address line) to make the complete API for a programmatic request.

For example the **API Request**

> Base URL + /prototypegsac/gsacapi/site/search?
> site.code.searchtype=exact&output=site.html&limit=500&site.name.searchtype=exact

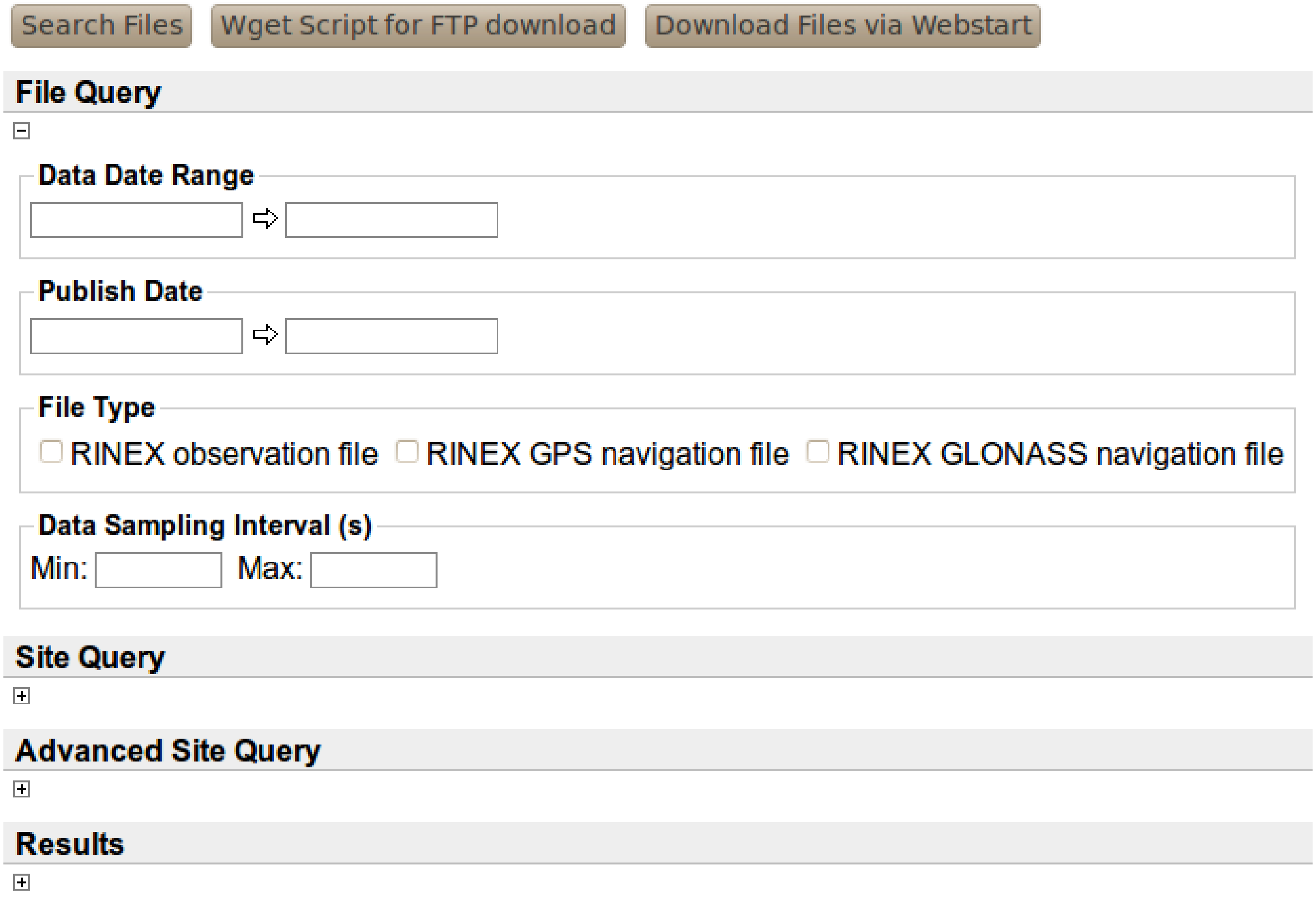
combined with the URL

> http://www.myrepository.org

can be used to compose this LINUX curl command to call the GSAC API (see section 4.1.1):

```
curl "http://www.myrepository.org/prototypegsac/gsacapi/site/search?
site.code.searchtype=exact&output=site.html&limit=500&site.name.searchtype=exact"
```

3.2 Search Files


Search Files
Wget Script for FTP download
Download Files via Webstart
File Query
⊟
Data Date Range
⇨
Publish Date
⇨
File Type
□ RINEX observation file
□ RINEX GPS navigation file
□ RINEX GLONASS navigation file
Data Sampling Interval (s)
Min:
Max:
Site Query
⊞
Advanced Site Query
⊞
Results
⊞

The Search Files web page has the form to query about instrumental data files and products from sites, available with this GSAC server. You can search for instrument data files and products by date range of data, by data type, from particular sites, and with other choices.

The Data Date Range entry boxes work the same way as in the site search page, to select for instrument data files and products by date range of data.

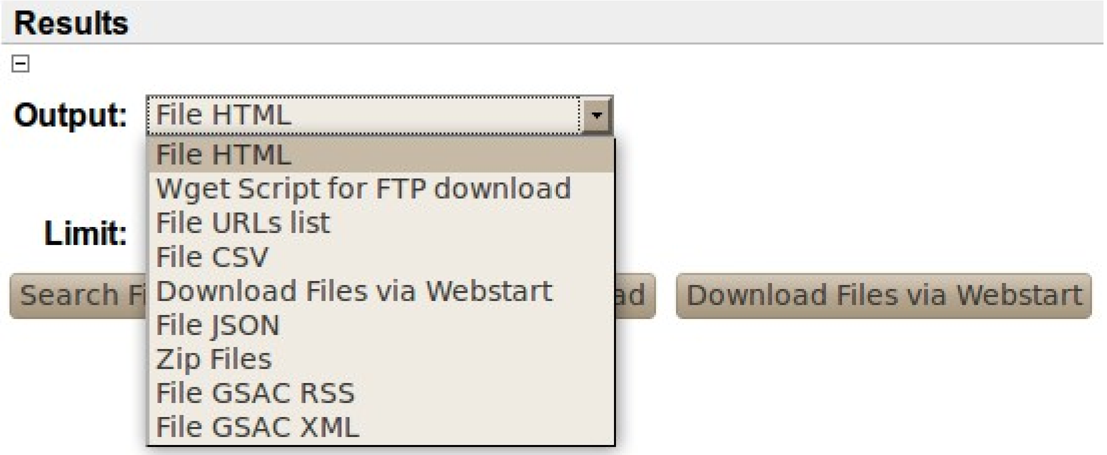
The Publish Date range can be used to find files modified or replaces since a previous download.

Check one or more boxes in the File Type area to limit files found to those types. GSAC can handle many types of data files besides the types shown in the particular example above. All file types available with GSAC at at data center are shown in this box.

The Data Sampling Interval range can be used to find data files with a given instrument sampling rate, in seconds. 50 Hz is 0.02 s.

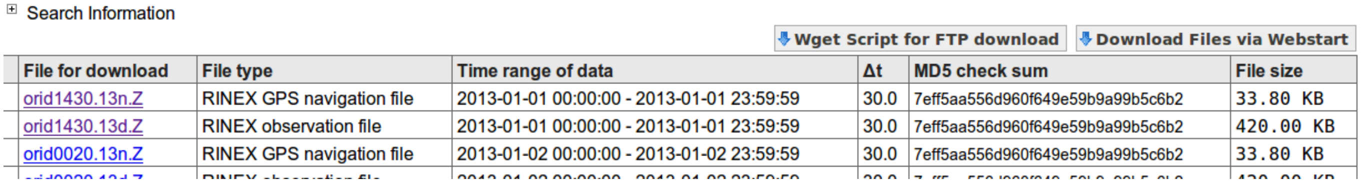
Click on the [+] under Site Query to see all the same controls by site you see in the site search page; likewise you can open the Advanced Site Query choices with its [+].

The Results [+] shows the Output (format) choices for file searches:



The Output box provides a choice of formats for the query results. Choices include, for example, HTML (for a table in web pages of results), a CSV list, of files, a Linux wget script for one or more data files, and simple list of data file URLs, and a Zip file containing one or more data files.

The HTML table of results from a file search looks like:



File for download	File type	Time range of data	Δt	MD5 check sum	File size
orid1430.13n.Z	RINEX GPS navigation file	2013-01-01 00:00:00 - 2013-01-01 23:59:59	30.0	7eff5aa556d960f649e59b9a99b5c6b2	33.80 KB
orid1430.13d.Z	RINEX observation file	2013-01-01 00:00:00 - 2013-01-01 23:59:59	30.0	7eff5aa556d960f649e59b9a99b5c6b2	420.00 KB
orid0020.13n.Z	RINEX GPS navigation file	2013-01-02 00:00:00 - 2013-01-02 23:59:59	30.0	7eff5aa556d960f649e59b9a99b5c6b2	33.80 KB

Note the time range of the data, the file size, and the file's MD5 check sum value may be shown.

You can click on a file name below File for Download to download that one file.

Click on the tab **Wget Script for FTP download** to download a file called gsacwget.sh, with lines like

```
wget http://egelados.gein.noa.gr/services/GPS/GPS_DATA/2013/001/orid1430.13n.Z
wget http://egelados.gein.noa.gr/services/GPS/GPS_DATA/2013/001/orid1430.13d.Z
wget http://egelados.gein.noa.gr/services/GPS/GPS_DATA/2013/002/orid0020.13n.Z
```

These lines can be used as a script in Linux to download all the data files and product files listed in the table of file search results.

Other choice of formats in the Search Files page include Files URLs list, a simple list of complete URLs for data files from the FTP or HTTP server associated with the GSAC (GSAC itself does not download files). The Wget Script for FTP download choice is the same as the tab in the file search results table.

The File CSV search result file more information about each file, in a csv formatted file, like

```
#fields=site_4char_ID[type='string'],Data_Type[type='string'],MD5[type='string'],FileSize,PublishDate[type='date'
format='yyyy-MM-dd HH:mm:ss'],URL[type='string'],dataStartTime[type='date' format='yyyy-MM-dd
HH:mm:ss'],dataStopTime[type='date' format='yyyy-MM-dd HH:mm:ss'],sampleInterval
#   Generated by UNAVCO GSAC Repository on 2014-02-12 19:52:27 +0000

MOPR,GNSS RINEX Observation QC Report,a257dd2929e8bb9d157be332612ddace,19918,2011-06-15
00:00:00,ftp://data-out.unavco.org/pub/rinex/qc/2009/197/mopr1970.09S,2009-07-16 00:00:00,2009-07-16
23:59:45,0.0,Mona Island Puerto Rico 1HZ

MOPR,GNSS RINEX Observation (Hatanaka  Unix
Compressed),87e462886411b2e9c9af26b4b70370ec,634297,2011-06-15 00:00:00,ftp://data-
out.unavco.org/pub/rinex/obs/2009/197/mopr1970.09d.Z,2009-07-16 00:00:00,2009-07-16 23:59:45,15.0,Mona
Island Puerto Rico 1HZ
```

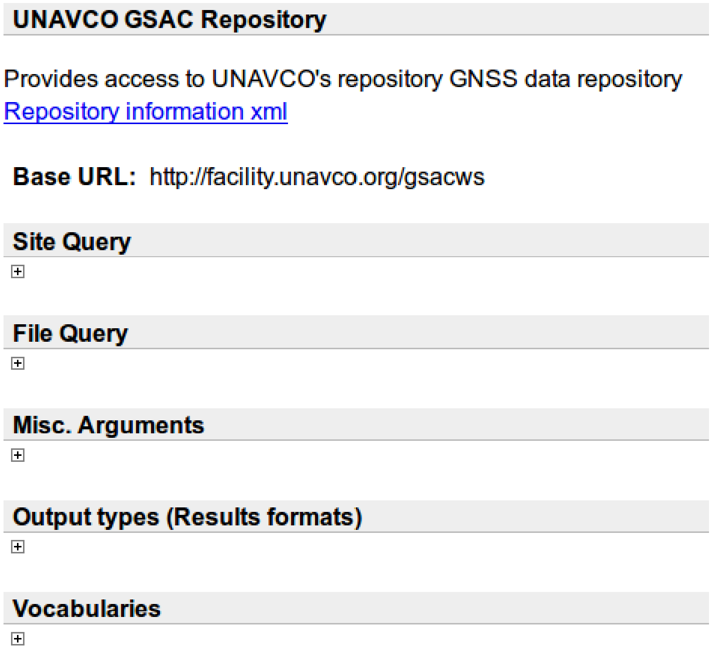
The choice Zip Files should make one new file called gsacresults.zip which contains all the data files listed in the table of file search results, zipped together, which you can download. If your list of files found is not too long, this is a fast way to get a collection of data files. Caution: before using this result, make your search with a web page table of results to see the total size of the files found, listed at the bottom of the table of results from a file search.

At the top of the table of results from a file search is a [+] Search Information. Click on it to see all the result output format choices. Clicking on one will repeat your most recent search, only now making a new output format.

To learn how to compose the GSAC API request for file searches, use the GSAC's Search Files page to make a request like you need. Use the default HTML output format. When the search finds the sites you want, in the table of files found, click on the **API Request** tag (in any GSAC version after December 2104). This shows the API equivalent to the request you made with the web page choices. Merge this with the GSACs base URL (see the URL in the browser address line) to make the complete API for a programmatic request.

3.3 API Information, Help, and Browse

The API Information page (from that link on top of every GSAC web page) has information needed to access a repository via API commans to a GSAC server. The page looks like this example from the UNAVCO GSAC:



UNAVCO GSAC Repository

Provides access to UNAVCO's repository GNSS data repository
Repository information xml

Base URL: http://facility.unavco.org/gsacws

Site Query
⊞

File Query
⊞

Misc. Arguments
⊞

Output types (Results formats)
⊞

Vocabularies
⊞

The Base URL is the first part of a URL for this particular GSAC-enabled repository. For example the UNAVCO GSAC repository has base URL http://www.unavco.org/gsacws. You use the Base URL in composing API queries as described in section 4. If they fail due to a bad connection the "Base URL" value shown on the Information web page may be wrong. Use instead the URL in the browser's address line for any GSAC page. if the GSAC operator has hidden the actual GSAC URL by a web site redirection, then you can't use API queries to their GSAC, a major problem outside of the control of GSAC.

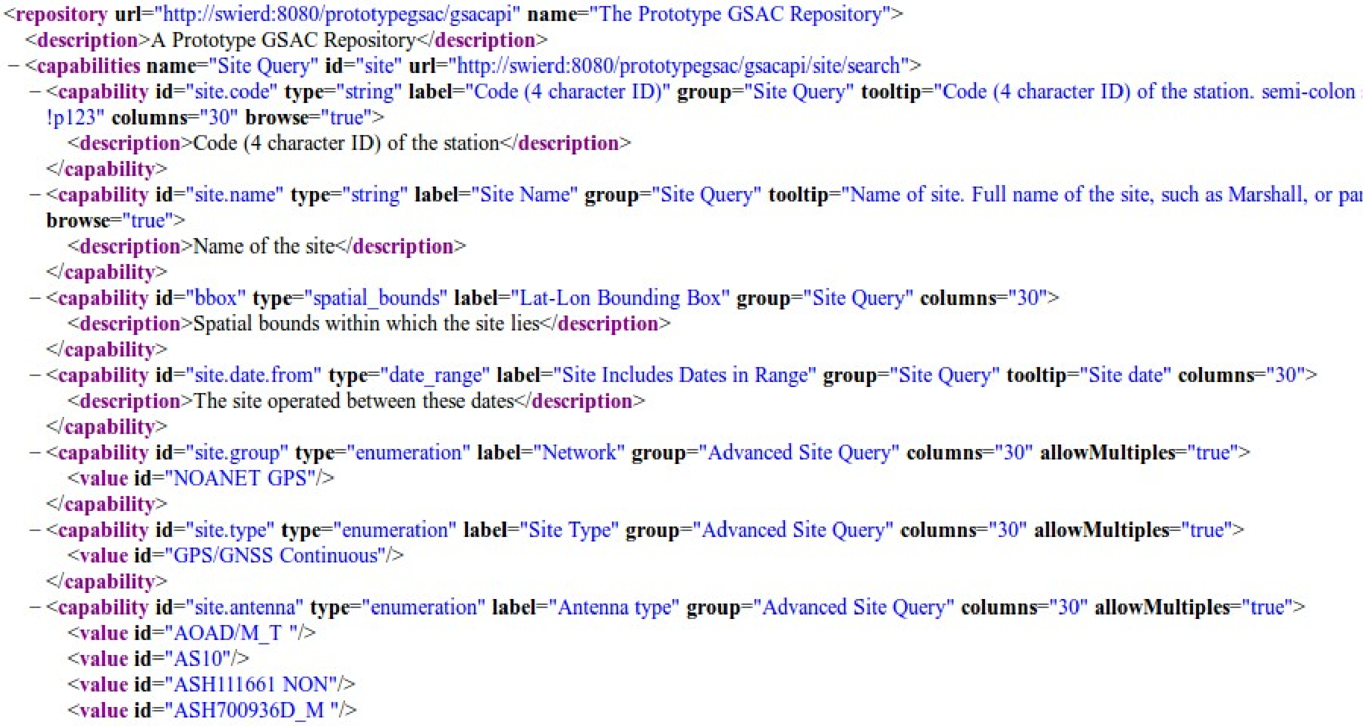
The link Repository information xml shows a collection of capabilities for this particular server. These are something like a WSDL (Web Services Description Language) document, an XML formatted file for describing network services, but not a true WSDL. See more details below.

This XML file lists "capabilites" of its GSAC server, which are named parameters that can be used to query GSAC. The "site." values after each "capability id=" gives the names used in queries to this GSAC using HTTP (RESTful) queries, with command line commands, with the GSAC client program, and from Federated GSAC servers.

Some examples of capabilities in a standard GSAC are

```
<capability  id="site.name"  type="string"  label="Site Name"
<capability  id="site.code"  type="string"  label="Code (4 character ID)"
<capability  id="bbox"  type="spatial_bounds"  label="Lat-Lon Bounding Box"
<capability  id="site.date.from"  type="date_range"  label="Site Includes Dates in Range"
<capability  id="site.group"  type="enumeration"  label="Network"
<capability  id="site.type"  type="enumeration"  label="Site Type"
<capability  id="site.antenna"  type="enumeration"  label="Antenna type"
<capability  id="site.dome"  type="enumeration"  label="Radome type"
<capability  id="file.type"  type="enumeration"  label="File Type"
```

These should be the same in all standard GSACs, so that interactions with all GSACs are consistent.
On the API Information page, the link Repository information xml begins like:



```
<repository url="http://swierd:8080/prototypegsac/gsacapi" name="The Prototype GSAC Repository">
  <description>A Prototype GSAC Repository</description>
 – <capabilities name="Site Query" id="site" url="http://swierd:8080/prototypegsac/gsacapi/site/search">
   – <capability id="site.code" type="string" label="Code (4 character ID)" group="Site Query" tooltip="Code (4 character ID) of the station. semi-colon
     !p123" columns="30" browse="true">
       <description>Code (4 character ID) of the station</description>
     </capability>
   – <capability id="site.name" type="string" label="Site Name" group="Site Query" tooltip="Name of site. Full name of the site, such as Marshall, or pa
     browse="true">
       <description>Name of the site</description>
     </capability>
   – <capability id="bbox" type="spatial_bounds" label="Lat-Lon Bounding Box" group="Site Query" columns="30">
       <description>Spatial bounds within which the site lies</description>
     </capability>
   – <capability id="site.date.from" type="date_range" label="Site Includes Dates in Range" group="Site Query" tooltip="Site date" columns="30">
       <description>The site operated between these dates</description>
     </capability>
   – <capability id="site.group" type="enumeration" label="Network" group="Advanced Site Query" columns="30" allowMultiples="true">
       <value id="NOANET GPS"/>
     </capability>
   – <capability id="site.type" type="enumeration" label="Site Type" group="Advanced Site Query" columns="30" allowMultiples="true">
       <value id="GPS/GNSS Continuous"/>
     </capability>
   – <capability id="site.antenna" type="enumeration" label="Antenna type" group="Advanced Site Query" columns="30" allowMultiples="true">
       <value id="AOAD/M_T "/>
       <value id="AS10"/>
       <value id="ASH111661 NON"/>
       <value id="ASH700936D_M "/>
```

In a row like

<capability id="site.state" type="enumeration" label="State" group="Advanced Site Query" columns="30"

allowMultiples="true">

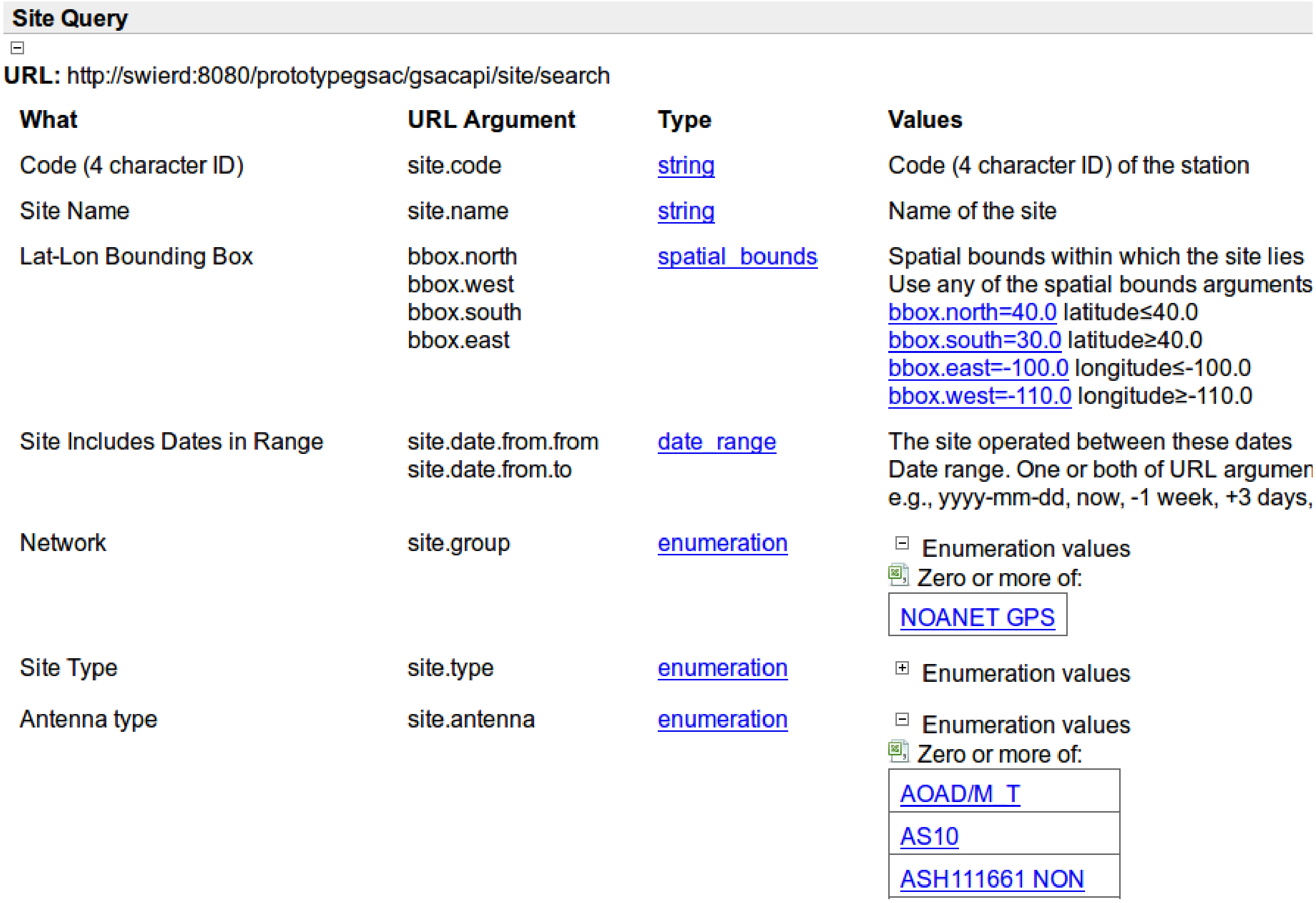
the value of the 'capability id' in the Repository information xml are GSAC's internal names for API parameters, and many names are (usually) common to all standard GSAC repositories. These are also listed as the "URL argument" in the **API Information** page sections Site Query, File Query, Misc. Arguments, and Output Types described just below. The Repository information xml also shows supplied (or meaningful) values for enumerated types of data in the database, such as the 'capability id=site.antenna' name values in the XML file above. You can search for any value of a capability; if the archive has no such data, then no results will be found. The enumerated values such as "AS10" show values where there are holdings in the archive.

You can see a complete Repository information xml page at any online GSAC, in the **API Information** page.

The information available under the **API Information** page sections Site Query, File Query, Misc. Arguments, and Output types (Results formats) is more suitable for human use, in composing API queries.

Note that *GSAC does not read data files or product files* to find metadata about data files. GSAC reads a database with information about data files and product files. This design approach allows GSAC to be generic or generalized. *GSAC does not know any file formats, does not read files, and does not check anything in the data files or in the product files it offers for download*. GSAC is not a system to control file processing of any kind, other than to provide links for downloads. This allows GSAC to, in principle, handle most any kind of data file from an Earth-located observing station. It is the responsibility of the GSAC operator to ensure that the information available to GSAC, in the database, is correct.

The sections Site Query, File Query, Misc. Arguments, and Output Types show the API parameter names ('URL argument'), value types ('Type'), the meaning of each parameter ('What'), and details of possible values ('Values') that you can use to query on in a GSAC implementation. Here is part of a **Site Query** section:



Site Query

⊟

URL: http://swierd:8080/prototypegsac/gsacapi/site/search

What	**URL Argument**	**Type**	**Values**
Code (4 character ID)	site.code	string	Code (4 character ID) of the station
Site Name	site.name	string	Name of the site

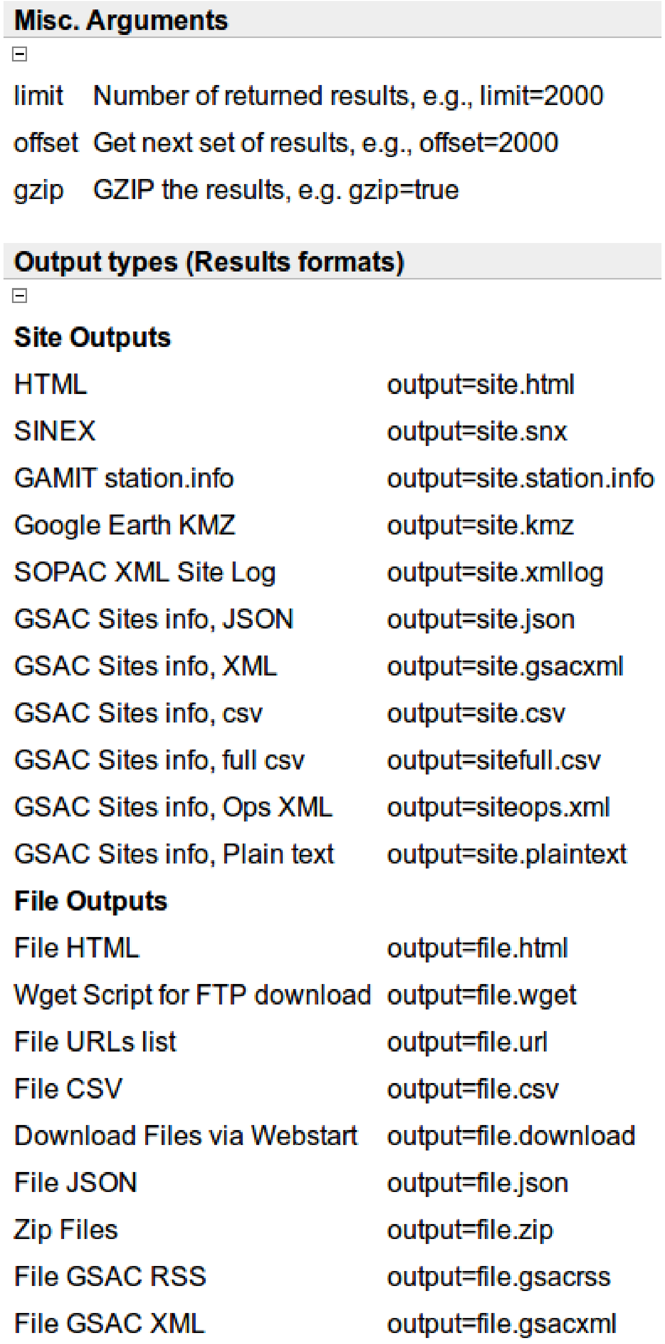
Lat-Lon Bounding Box	bbox.north
	bbox.west
	bbox.south
	bbox.east
		spatial_bounds	Spatial bounds within which the site lies
			Use any of the spatial bounds arguments
			bbox.north=40.0 latitude≤40.0
			bbox.south=30.0 latitude≥40.0
			bbox.east=-100.0 longitude≤-100.0
			bbox.west=-110.0 longitude≥-110.0

Site Includes Dates in Range	site.date.from.from site.date.from.to	date_range	The site operated between these dates Date range. One or both of URL argumen e.g., yyyy-mm-dd, now, -1 week, +3 days,
Network	site.group	enumeration	⊟ Enumeration values Zero or more of: NOANET GPS
Site Type	site.type	enumeration	⊞ Enumeration values
Antenna type	site.antenna	enumeration	⊟ Enumeration values Zero or more of: AOAD/M_T AS10 ASH111661 NON

See section 4, the API, below for how this is used.

Similarly, the parameters in the **API Information** sections File Query, Misc. Arguments, and Output Types are used in the GSAC API.

Here are complete examples of Misc. Arguments, and Output Types sections:



Misc. Arguments

⊟

limit	Number of returned results, e.g., limit=2000
offset	Get next set of results, e.g., offset=2000
gzip	GZIP the results, e.g. gzip=true

Output types (Results formats)

⊟

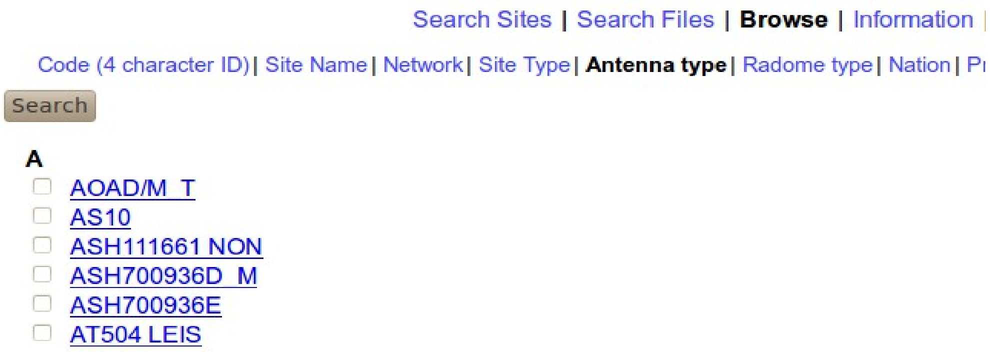
Site Outputs	
HTML	output=site.html
SINEX	output=site.snx
GAMIT station.info	output=site.station.info
Google Earth KMZ	output=site.kmz
SOPAC XML Site Log	output=site.xmllog
GSAC Sites info, JSON	output=site.json
GSAC Sites info, XML	output=site.gsacxml
GSAC Sites info, csv	output=site.csv
GSAC Sites info, full csv	output=sitefull.csv
GSAC Sites info, Ops XML	output=siteops.xml
GSAC Sites info, Plain text	output=site.plaintext
File Outputs	
File HTML	output=file.html
Wget Script for FTP download	output=file.wget
File URLs list	output=file.url
File CSV	output=file.csv
Download Files via Webstart	output=file.download
File JSON	output=file.json
Zip Files	output=file.zip
File GSAC RSS	output=file.gsacrss
File GSAC XML	output=file.gsacxml

The 'limit' parameter in Misc Arguments is useful when exploring a GSAC repository with the web forms, to avoid long tables of results, and to limit how many files to download from one file search.

The Output Types (Results formats) shows the many formats of results available from GSAC from searches about sites, and from file searches to retrieve instrumental data files and product files.

Each GSAC web page has a **Help** link which has a link to this User Guide document at UNAVCO, and to the UNAVCO GSAC web site (http://www.unavco.org/software/data-management/gsac/gsac.html).

Each GSAC web page has a **Browse** link which lets you find sites with one search choice, a single parameter value:


Search Sites | Search Files | Browse | Information |
Code (4 character ID) | Site Name | Network | Site Type | Antenna type | Radome type | Nation | Pr
Search

A

- □ AOAD/M_T
- □ AS10
- □ ASH111661 NON
- □ ASH700936D_M
- □ ASH700936E
- □ AT504 LEIS

4. The API

Any GSAC Repository supports a RESTful web API (application interface). You compose a URL to do queries and choose a format for the results, and to get the results, without using GSAC UI web pages. You can enter the API request in a web browser page address, or use computer utilities to make the API call. Results go to the web page, to a computer terminal screen, or to a file.

For use of the GSAC API, first see section 3.3 which introduces arguments to GSAC queries. The GSAC API uses the same arguments, as does the GSAC client (section 5 below).

Any GSAC API URL begins with the "Base URL " of the repository, supposedly listed in its web site API Information page, such as the "Base URL" http://www.unavco.org/gsacws.

You use that Base URL in composing API queries. If they fail due to a bad connection the "Base URL" value shown on the Information web page may be wrong. Use instead the URL in the browser's address line for any GSAC page. if the GSAC operator has hidden the actual GSAC URL by a web site redirection, then you can't use API queries to their GSAC, a major problem outside of the control of GSAC.

Examples here use the UNAVCO GSAC at "http://www.unavco.org/gsacws".

4.1 Site Queries

For GSAC queries you add options to the "Base URL". Foe example, for a site query by 4 character ID for site code P210

```
http://www.unavco.org/gsacws/gsacapi/site/search?site.code=P210
```

Enter that in a browser, which searches the UNAVCO GSAC archive for all sites with code P210 (there are 3).

This query results in the default format for a site query, HTML, which makes the new page in the browser.

To choose other output formats you append "`&output=`" such as

```
http://www.unavco.org/gsacws/gsacapi/site/search?site.code=P210&output=site.snx
```

which returns a SINEX file of site metadata. Check the Results Output list in API Information for other formats.

Note that the & character is glue between search name-value pairs. In a site name like "St Helena" use the string `St+Helena` in the `site.name=` search with + for the space in the name. Do not quote any character string.

All terms for an GSAC API query at a particular GSAC repository are listed on its web site API Information page, under **Site Query** for sites, and under **File Query** for files. File searches can also use all site selection choices. All the valid terms for return formats for site queries are listed on the web site's API Information page, under **Output types -> Site Outputs**.

4.1.1 Linux curl GSAC commands

GSAC API commands can be called with programs or utilities on your operating system. For example, GSAC can be invoked from a Linux command with the Linux '`curl`' utility. For example this command uses curl with a site searchh with SINEX output:

```
curl "http://www.unavco.org/gsacws/gsacapi/site/search?
site.code=P210&output=site.snx"
```

To see if you have curl, use the command `which curl`, normally returning something like `/usr/bin/curl`.

Note that the complete URL in a curl command is inside quotation marks " ".

Here are some more examples of site searches using Linux commands with the curl utility.)The URL in each command can be used in a browser for the same query and result.) The 'arguments' in the command follow the "?" and the "&".

The command:

```
curl "http://www.unavco.org/gsacws/gsacapi/site/search/sites?
output=site.snx&site.code=P30*"
```

will list on your terminal a SINEX file for all GNSS stations whose code (ID) begins with "P30" (ten stations in this case).

For several distinct stations by each code (four character ID):

```
curl "http://www.unavco.org/gsacws/gsacapi/site/search?
site.code=P123&site.code=MTWO&site.code=AB04&output=site.snx"
```

Or both ways, by individual names and with a wildcard search for IDs like AB0*:

```
curl "http://www.unavco.org/gsacws/gsacapi/site/search?
site.code=P120&site.code=MTWO&site.code=AB0*&output=site.snx"
```

(The "*" must be at the end of the wildcard; options like AB*5 will not work.) To request the UNAVCO GSAC for five stations by a site ID list separated with ';', with output as a SINEX site metadata file named five_sites.snx:

```
curl "http://www.unavco.org/gsacws/gsacapi/site/search/sites.snx?
output=site.snx&site.code=cn10;cn11;cn12;cn13;abmf" > five_sites.snx
```

To request the UNAVCO GSAC for all COCONet network stations, results as SINEX:

```
curl "http://www.unavco.org/gsacws/gsacapi/site/search/sites.snx?
output=site.snx&site.group=COCONet"    > COCONet_sites.snx
```

GSAC correctly merges the information from all stations found into one SINEX file.

To get results in the GAMIT "station.info" format, use argument `output=site.station.info`:

```
curl "http://www.unavco.org/gsacws/gsacapi/site/search/sites?
output=site.station.info\&site.code=p123"
```

To make results in the "SOPAC XML log file" format, use output=site.xmllog; for JSON use output=site.json; for a CSV file use output=site.csv.full; and for Google Earth KMZ use output=site.kmz.

To get stations inside a latitude-longitude bounding box (region), use these spatial bounding arguments:

`bbox.north=40.0`	for latitudes ≤40.0, the north side of the box is at 40 north latitude;
`bbox.south=30.0`	for latitudes ≥30.0
`bbox.east=-100.0`	for longitudes ≤-100.0; the east side of the box is at longitude -100 (100 west longitude)
`bbox.west=-110.0`	for longitudes ≥-110.0

Longitude is in positive degrees east.

Also you can limit the station count returned with the argument `limit=`, such as `limit=25`.

For example:

```
curl "http://www.unavco.org/gsacws/gsacapi/site/search/sites?
output=site.station.info\&bbox.north=41.0\&bb.east=-104.0\&bb.south=37.0\&bb.west=-
112.0\&limit=25"
```

Add "`> demo_station.info`" to the end of the command to put the results in a local file named demo_station.info. Lacking the `limit=25`, this command returns 575 lines of information in station.info format.

The same search, with results in SOPAC XML site log format, returns 31736 lines.

To learn more about how to compose the GSAC API URL for the site search you need, first use the GSAC's web page to make a site like you need. Use the default HTML output (web pages) . When the search finds the sites you want, in the table of sites found click on the **API Request** tag (as of any GSAC version after December 2104). This shows the API you used. You can use this in a programmatic request, like a curl request. For example a search with the UNAVCO GSAC for sites in the island of Hawaii is

```
/gsacws/gsacapi/site/search?
bbox.north=20.45&output=site.html&site.sortorder=ascending&site.name.searchtype=exac
t&site.code.searchtype=exact&bbox.south=18.01&bbox.west=-156.5&limit=500&bbox.east=-
154&site.interval=interval.normal
```

You add `http://www.unavco.org/` to the front for use with curl.

To find recent stations in the island of Hawaii, working in ther time interval 01-01-2014 to 01-01-2015, add a date range constraint:

```
/gsacws/gsacapi/site/search?site.date.from=2014-01-
01&bbox.north=20.45&output=site.html&site.sortorder=ascending&site.name.searchtype=e
xact&site.code.searchtype=exact&bbox.south=18.01&bbox.west=-
156.5&limit=500&bbox.east=-154&site.date.to=2015-01-01&site.interval=interval.normal
```

You can edit this slightly to get output as a JSON file, using `&output=site.json`. Likewise you can choose other "`output=`" choices from the GSAC web site API Information page, **Output Types** section.

To see only stations in the "Hawaii USGS HVO" network, add `site.group=Hawaii+USGS+HVO` to the end of the API command. Spaces in GSAC API arguments of strings must become "+"; spaces are not allowed in API commands.

To see only stations whose 4 character ID begins with K, add &site.code=k* to the API command.

4.2 File Queries

For querying files you use an API URL beginning with the Base URL, like for site searches. File queries are very much like site queries, but begin with, for example,

```
curl "http://www.unavco.org/gsacws/gsacapi/file/search?
```

before the API arguments.

Here is a URL to get metadata about data files in a given time range from stations in the network "COCONet" and whose sampling rate is "normal." The API choice `site.interval=interval.normal` is distinctive to the UNAVCO GSAC. The output is redirected with the Linux > to a file.

```
http://www.unavco.org/gsacws/gsacapi/file/search?
site.group=COCONet&output=file.csv&file.datadate.from=2013-11-
01&file.datadate.to=2014-01-31&site.interval=interval.normal&limit=50000" >
coconet_2013Nov-thru2014Jan_data_files_normrate_info.csv
```

The search is limited to 50,000 lines of file metadata. This query found information 29,942 files.

Some file queries return one or more data files. Be cautious, since a single GNSS file may be, say, 8 MB and some searched may find hundreds of files.

All the valid terms for a file query at a particular GSAC repository are listed on its web site API Information page, under **File Query**. The results can be returned in a variety of output formats using the "`&output=`" URL term plus a value, such `&output=file.zip.` All the valid output arguments for return formats for file queries are listed on the web site's API Information page, under **Output types -> File Outputs.**

4.3 More Query Help

In the **API Information** page you find additional API query terms, such as in sections **Misc. Arguments** and **Vocabularies.** See section 3 above.

To learn more about GSAC API use, try some site and file queries with GSAC API arguments using a browser and a working GSAC repository, such as the UNAVCO GSAC Repository at http://www.unavco.org /data/gsacws.

4.3.1 Repository Information

To get the GSAC API capabilities programatically, you can query any GSAC for its repository information. The "Repository Information" for a GSAC web site lists all of the API arguments. You can see this information in a GSAC API Information page, at the Repository information xml link, or retrieve it with a command like this:

```
http://www.unavco.org/gsacws/gsacapi/repository/view?output=xml
```

for the UNAVCO GSAC.

4.3.2 URL Argument Types

Here are more details about GSAC API arguments. We have tried to be flexible with how URL arguments are used. In this section "arg" stands for one of the API's URL arguments such as "site.code."

- string - There may be none, one, or more of a URL string argument:

  ```
      arg=p123&arg=p456
  ```

  Each argument may be a semi-colon delimited list of argument values:

  ```
      arg=p123;p456
  ```

  Wild cards are supported:

  ```
  arg=p12*  - find values that begin with P12
  arg=*123 - find values that end with P12
  arg=*12* - find values that contain P12
  ```

  As well as negation:

  ```
  arg=!p123 do not show results from station P123
  ```

  Also, one can specify the string search type with the <arg>.searchtype argument. This can take on the values:

  ```
  <arg>.searchtype=exact
  <arg>.searchtype=beginswith
  <arg>.searchtype=endswith
  <arg>.searchtype=contains
  ```

- enumeration - Just like the string type but without the wildcard capabilities.
- boolean - A boolean argument may be undefined, "true" or "false".
- date_range - There are 2 arguments that may specified: "<arg>.from" and "<arg>.to". There may be none, one or both of these. If specified the logic is:

  ```
  date value ≥ arg.from
  date value ≤ arg.to
  ```

A date string can be any of the following forms. See http://download.oracle.com/javase/1.4.2/docs/api/java/text/SimpleDateFormat.html for information.

```
yyyy-doy  - year and 3 digit day of year
yyyy-MM-ddTHH:mm:ss Z
yyyyMMddTHHmmss Z
yyyy/MM/dd HH:mm:ss Z
yyyy-MM-dd HH:mm:ss Z
EEE MMM dd HH:mm:ss Z yyyy
yyyy-MM-ddTHH:mm:ss
yyyyMMddTHHmmss
yyyy/MM/dd HH:mm:ss
yyyy-MM-dd HH:mm:ss
yyyy-MM-ddTHH:mm Z
yyyyMMddTHHmm Z
yyyy/MM/dd HH:mm Z
yyyy-MM-dd HH:mm Z
yyyy-MM-ddTHH:mm
yyyyMMddTHHmm
yyyy/MM/dd HH:mm
yyyy-MM-dd HH:mm
yyyy-MM-dd
yyyy/MM/dd
yyyyMMdd
yyyy-MM
yyyyMM
yyyy/MM
yyyy
now
```

If a timezone is not specified then the date/time is assumed to be in UTC.

You can also specify a date value that is relative to the other date:

-1 week, +3 days, -1 month, etc.

For example, to find the last 2 weeks of data from current time you would use:

arg.from=-2 weeks&arg.to=now

If you wanted to search 3 days into the future:

arg.from=now&arg.to=+3 days

- number_range - There are 2 arguments that may specified: "<arg>.min" and "<arg>.max". There may be none, one or both of these. If specified the logic is:

  value ≥ arg.min
  value ≤ arg.max

- spatial_bounds - There are 4 arguments that may specified:

  `<arg>.north` [most northerly]
  `<arg>.south` [most southerly]
  `<arg>.east` [most easterly]

`<arg>.west` [most westerly]

The values are assumed to be in degrees north (for latitude) and degrees east (for longitude). Not all arguments are needed. If specified the logic is:

```
site location latitude ≤ arg.north
site location latitude ≥ arg.south
site location longitude ≤ arg.east
site location longitude ≥ arg.west
```

Latitude and longitude values are in decimal degrees in GSAC.

5. The GSAC Client Program

There is a GSAC client program (gsacclient.jar) for programmatic searches of a GSAC repository and to download files. The GSAC client is available as part of the GSAC package. This program is an alternative on Windows for the use of 'curl' commands on Linux, and the client may be run on Linux as well as Windows.

First review sections 3 and 4 about the GSAC API for basic concepts and API information.

The GSAC client makes site and file queries of any GSAC repository, and gets the results returned in a variety of formats. The client uses the GSAC RESTful web API (application interface). You run the GSAC client program on your own terminal's command line to do programmatic searches of a GSAC repository, and download files. In some cases this is more efficient than using the web page search forms.

To run the GSAC client you need Java 1.6 or higher. If needed, install Java on your machine.

5.1 Download the Client

The GSAC command line client is available for download as part of the GSAC package at SourceForge.

Download the GSAC client code from SourceForge. On `http://sourceforge.net/projects/gsac/`

click "Download gsacclient.zip." The file is about 4 MB.

Once downloaded, simply unzip the file `gsacclient.zip` (Linux):

```
unzip gsacclient.zip
  Archive:  gsacclient.zip
   creating: gsacclient/
   creating: gsacclient/lib/
  inflating: gsacclient/lib/gsacclient.jar
  inflating: gsacclient/gsacclient.sh
  inflating: gsacclient/README
  inflating: gsacclient/gsac.properties
```

This will create a directory (folder) called `gsacclient,` which includes, among other items, the file `gsacclient.sh` and the Java file `gsacclient/lib/gsacclient.jar`, which is the executable client. Since it is in Java it should run on any platform with Java 1.6.

5.2 Running the Client

The client program is the Java archive file `gsacclient/lib/gsacclient.jar.` You can run the GSAC client on LINUX or Mac using either the shell file, `gsacclient.sh`:

```
cd gsacclient/

gsacclient.sh <arguments>
```

or directly using Java and the jar file:

```
java -jar gsacclient/lib/gsacclient.jar <arguments>
```

You first may need to change file permissions; for example, by:

```
chmod 744 gsacclient/gsacclient.sh
```

The examples shown here below should work if you try them in your terminal.

On Windows, since the gsacclient is a command line tool, it must be run from the terminal, running the "jar" file.

In Windows, the terminal is the Command Prompt application. Click on Start, and then Run, and in the Run window type cmd, and click on OK to bring up a DOS emulation window where you can enter command lines.

The client uses the GSAC API arguments and also its own arguments. To see the client's command line arguments run the client with -help, such as

```
gsacclient.sh -help
```

which shows

```
-properties <properties file to load>
-server  http://examplegsacrepository.edu/someprefixpath, e.g.
      http://www.unavco.org/gsacws
-info  fetch and print to stdout the remote GSAC repository's information,
      including available arguments
-download <destination directory> Do a file search and download the files to
      the given directory
-keep_paths <true|false> When doing the download, do we maintain the directory
      structure of the ftp urls?  Default is true
-query site  or -site means do a site query (and add other arguments).
-query file  or -file means do a file query (and add other arguments).
-out <outputfile>  Write the output to the specified file
-output  specify the format of the query results (such as 'site.xmllog' or
      'file.url' )
-url <url to fetch> <optional filename to write to> act like wget
 After above gsaclient arguments you can add any number of query arguments,
 e.g.:
-site.code P12*
-bbox west-longi south-lati east-longi north-lati, such as bbox -120.0 30.5
      -117.0 33.0
Note: for any of the arguments you can specify a file that contains extra
      arguments, e.g.:
      -site.code file:site_queries.txt
where sites_queries.txt contains site query codes and values, one per line.
```

The `-server` argument is essential to query a remote GSAC-enabled repository. For the server address of a particular GSAC installation, use the Base URL shown in the GSAC'S site API Information page.

5.3 Listing Repository Information

You can get information (including gsacclient arguments) about a GSAC repository with for example

```
gsacclient.sh -server http://www.unavco.org/gsacws -info
```

Similar and more complete information is on the GSAC repository's API Information page; click on Repository information xml . That list will also show allowed values for enumerated fields in the GSAC database.

5.4 Site searches

A sample query for one GPS site is:

```
gsacclient.sh -server http://www.unavco.org/gsacws -site.code ADE1
```

Spaces separate arguments in the gsacclient command line. Do *not* have a "/" at the end of the server URL.

The results are sent to your terminal, from the remote GSAC web service, in the short csv format with header line:

```
  http://www.unavco.org/gsacws/gsacapi/site/search?site.code=ADE1&out
  put=site.csv
  #site.id,site.code,site.name,site.latitude,site.longitude,site.elevation
  23478_ADE1_5650,ADE1,Australia NGA collocated,-34.729,138.6473,38.1548
```

The results sent to your screen begin with a line ("http...") showing the GSAC URL which is equivalent to the gsacclient query, in this case:

```
http://www.unavco.org/gsacws/gsacapi/site/search?
site.code=ADE1&output=site.csv
```

This helps to learn or demo use of the GSAC API in browser requests. You can try this URL query request in a browser. Note that "&" is the glue to connect different arguments in the URL query, and "?" begins the arguments after the address. This is typical use of "&" and "?" in web services. No spaces are used.

You can get results in other formats using the argument "-output,", such as SOPAC XML Site Log format:

```
gsacclient/gsacclient.sh -server http://www.unavco.org/gsacws -site.code
"ADE1" -output site.xmllog
```

All the possible choices of formats of results from site queries are listed in the GSAC'S site **API Information** page, in the Output Types -> Site Outputs URL section. For example, for HTML (web page) output use site.html; XML Site Log (SOPAC) output=site.xmllog; SINEX output=site.snx; GAMIT station.info output=site.station.info; and for csv file output=site.csv.

This command

```
gsacclient/gsacclient.sh -server http://www.unavco.org/gsacws -site.code ADE*
```

finds three sites, all beginning with "ADE"; a wild card search.

That command, plus `-output site.xmllog` will send three SOPAC XML Site Logs to your terminal.

You can direct any gsacclient results, such as all those site logs in XML format, to a file:

```
gsacclient/gsacclient.sh -server http://www.unavco.org/gsacws -site.code ADE*
-output site.xmllog -out ADE_site_logs.xml
```

The command argument format for a latitude-longitude bounding box search i,s for example:

```
-site -bbox -120.0 30.5 -117.0 33.0
```

The '`bbox'` values are longitudes and latitudes: the most westerly, the most southerly, the most easterly, and the most northerly. Be sure to note that `bbox` is preceded by `-site`.

5.5 GSAC Client Properties File

You can specify one or more gsacclient arguments to always use when running the client. Just edit the `gsac.properties` file in the gsacclient/ directory. For example, use the line:

```
server=http://www.unavco.org/gsacws
```

to query the UNAVCO GSAC with every gsacclient call. To make every command a site query, add the line

```
query=site
```

You can also create and use several different properties files (in the same format). For example:

```
gsacclient.sh -properties  gsac_client_3.properties.
```

Will do queries using the arguments in `gsac_client_3.properties`. Any gsacclient arguments may be entered in a properties file. For example, if you want to get more than the default 1000 results, add -limit <count>, e.g.:

```
gsacclient.sh  -site -limit 1500
```

5.6 File searches

A site.code search to get all the files for site P123 (caution, this returns hundreds of file names):

```
gsacclient.sh  -file -site.code p123
```

Results in this case are a list of information about all files available from site P123, like this, not data files' contents:

```
3233505,GNSS RINEX Observation (Hatanaka, Unix
Compressed),739030c019ab4169acc66a317daead0a,638221,2007-03-06,ftp://data-
out.unavco.org/pub/rinex/obs/2007/064/p1230640.07d.Z

3258729,GNSS Navigation,74787c121b5fb8344240ee05c3927a33,32805,2007-02-
15,ftp://data-out.unavco.org/pub/rinex/nav/2006/279/p1232790.06n.Z
```

Other example gsacclient file queries:

Absolute time range file search for p123

```
gsacclient.sh  -file site.code p123
               -file.datadate.from 2010-09-01  -file.datadate.to 2010-09-15
```

which gets back 46 files.

Absolute time range file search for p123, and return FTP URLs for data files:

```
gsacclient.sh  -file site.code p123
               -file.datadate.from 2010-09-01
               -file.datadate.to 2010-09-15    -output file.url
```

Time relative file search for P123 :

```
gsacclient.sh  -file site.code p123
               -file.datadate.from "-1 month"
               -file.datadate.to now
```

Both the date 'from' and 'to' arguments must be used together. You could use

```
-file.datadate.to "-1 month".
```

You can search on file *type*; for example, to get RINEX obs files :

```
gsacclient.sh -file -output url site.code p123
              -file.datadate.from 2010-09-01
              -file.datadate.to 2010-09-15
              -file.type gnss.data.rinex.observation
```

File type choices for a particular GSAC repository are listed in the GSAC'S site API Information page, in the File Query -> File Type section. Click on the [+] before "Enumeration values," if there is one.

File Output formats are also selectable, and listed in the GSAC'S site API Information page, in the Output Types -> File Outputs section. This choice can get information about a file, in several formats, or download the file itself (`-output file.download`). You can also find FTP URLs and wget scripts.

5.7 File Downloads with the GSAC Client

This query will do a file query and download some files into `destination_directory`. You need first to make the directory `destination_directory`. There is a limit of four files in this command. Some sites ahve hundreds of files so be sure to use the "limit" option.

```
gsacclient.sh -file -download destination_directory  -site.code p123  -limit 4
```

That `gsacclient.sh` command maintains the original directory file paths from the remote GSAC server, and cretes the same path on your machine. For example if your local destination directory is data-dir/, you may find the data files in a path such as `data-dir/pub/rinex/obs/`, where the /pub/rinex/obs/ part mimics the full FTP directory path to the files on the remote GSAC repository. In GSAC client terminology, this is argument

"-keep_paths true."

To have all of the files dowmloaded into a single directory, set the the -keep_paths argument to false:

```
gsacclient.sh  -file -download destination_directory
               -site.code p123  -limit 4   -keep_paths false
```

To search for metadata about instrument data files (using the the gsac client, the web page UI, or the API)

```
gsacclient.sh -file site.code Palm -limit 15
```

you can for example specify "output" (format of result) of "file.url,"

```
gsacclient.sh -file site.code Palm -limit 15 output file.url
```

and you will get a list of URLS like this:

```
ftp://data-out.unavco.org/pub/rinex/obs/2012/158/palm1580.12d.Z
```

This is of course a URL which can be used to download a file with FTP from the given server.

As always, the gscacclient shows you first the equivalent one-line API URL:

```
Processing file query:
http://www.unavco.org/gsacws/gsacapi/file/search?
site.code=Palm&limit=15&output=file.url
```

If you use "output" of "file.wget" (using the the gsac client, the web page UI, or the API)

```
gsacclient.sh -file site.code Palm -limit 15 output file.wget
```

the result returned is one or more lines such as

```
wget ftp://data-out.unavco.org/pub/rinex/obs/2012/158/palm1580.12d.Z
```

which are wget commands to download a file with "wget" from the given server.

5.8 Using Linux with the GSAC Client

You can use gsac client calls as part of Linux commands, such as

```
gsacclient.sh -server http://www.unavco.org/gsacws -site.code ADE* -output
site.xmllog | grep receiverType
```

which lists the reciever types for sites beginning with ADE:

```
<gnssReceiver><equip:receiverType><![CDATA[ASHTECH Z-XII3]]></equip:receiverType>

<gnssReceiver><equip:receiverType><![CDATA[ASHTECH Z-XII3]]></equip:receiverType>

<gnssReceiver><equip:receiverType><![CDATA[TRIMBLE 5700]]></equip:receiverType>
```

You can write results in a new file ADE_receiver_list:

```
gsacclient.sh -server http://www.unavco.org/gsacws -site.code ADE* -output
site.xmllog | grep receiverType >> ADE_receiver_list
```

You can get the GSAC API (URL) for any query submitted through the GSAC client. The API URL is the first line returned from any query. See examples of GSAC client use in the section titled The GSAC Command Line Client.

6. GSAC Resources and Contacts

GSAC Information online

UNAVCO GSAC Web Site	http://www.unavco.org/software/data-management/gsac/gsac.html
GSAC code repository	http://sourceforge.net/projects/gsac

Current GSAC Repositories

See list on the UNAVCO GSAC web site.

UNAVCO GSAC Support

UNAVCO supports the UNAVCO GSAC code. We may be able to assist with installation and / or operation of GSAC servers. Email gsac@unavco.org.

GSAC Development Personnel (NASA contract), 2010-2012:

PI

- Fran Boler, UNAVCO, boler@unavco.org

CDDIS

- Carey Noll, Co-I GSAC, carey.noll@nasa.gov

SOPAC

- Yehuda Bock, Co-I GSAC, ybock@ucsd.edu
- Mindy Squibb, Software Engineer, msquibb@ucsd.edu

UNAVCO

- Jeff McWhirter, Software Engineer

UNR

- Geoff Blewitt, Co-I GSAC, gblewitt@unr.edu
- Corne Kreemer, Co-I GSAC, kreemer@unr.edu

UNAVCO GSAC Development and Support Personnel (NSF contract), Oct. 2012- Dec. 2013:

- Fran Boler, UNAVCO, PI and UNAVCO GSAC Project Manager, boler@unavco.org
- Stuart Wier, UNAVCO, Science Software Engineer, wier@unavco.org

Acknowledgements

GSAC was a NASA ROSES ACCESS Program funded project (2010-2012), through the ROSES ACCESS program Cooperative Agreement NNX10AF07A, developed at UNAVCO, NASA CDDIS, and SOPAC UCSD. In 2012 and 2013 NSF funded GSAC development at UNAVCO in support of COOPEUS.

Most of the original GSAC software was developed by Jeff McWhirter at UNAVCO (Boulder, Colorado), during 2010-2011, with assistance from staff at CDDIS (NASA's Crustal Dynamics Data Information System) and at SOPAC (the Scripps Orbit and Permanent Array Center, at the University of California at San Diego).