ESDS-RFC-014
Category: Technical Note
Updates: n/a

Lynnes, Yang, Hu, Domenico and Enloe
March 2009
Interoperability between OGC CS/W and WCS Protocols

Interoperability between OGC CS/W and WCS Protocols

Status of this RFC

This RFC Technical Note describes a project to provide a catalog search service for the Thematic Realtime Environmental Data Distribution System (THREDDS). Specifically, the project revealed lessons regarding the interoperability between two standards of the Open Geospatial Consortium (OGC), the Catalog Services for Web (CS/W) and Web Coverage Services (WCS). This RFC does not specify an Earth Science Data Systems (ESDS) standard. Distribution of this memo is unlimited.

Change Explanation

This document is not a revision to an earlier version.

Copyright Notice

Abstract

This document presents lessons related to Open Geospatial Consortium (OGC) protocols that were learned in the course of developing the OGC-Geoscience Gateway. The OGC-Geoscience Gateway is a NASA ACCESS project that provides a gateway between the OGC protocols and technologies widely used within the geosciences community, in this case THREDDS. The gateway allows a user to query a THREDDS catalog using the OGC CS/W protocol, making THREDDS-served data accessible to CS/W-aware GIS clients. Since THREDDS provides OGC WCS access, our end goal was to search THREDDS catalogs for WCS coverages in a GIS Client that could then seamlessly acquire the WCS coverage for display. Although this goal was only partially realized, several valuable lessons were learned with respect to CS/W interoperability, particularly with its sibling WCS protocol. We hope these lessons will be useful to OGC client developers, as well as OGC interoperability architects.

1 Introduction

In the past several years, interoperability gaps have made cross-protocol and cross-community data access a challenge within the Earth science community. One such gap is between two protocol families developed within the geospatial and Earth science communities. The Earth science community has developed a family of related geoscience protocols that includes Open-source Project for a Network Data Access Protocol (OPeNDAP) for data access and the Thematic Real-time Environmental Distributed Data Services (THREDDS) catalog capability. The corresponding protocols in the geospatial community are the Open Geospatial Consortium (OGC) protocols Web Coverage Service (WCS) for geospatial data access and Catalog Services for Web (CS/W) for metadata search. We have developed a catalog gateway to mediate client/server interactions between OGC catalog clients and THREDDS servers. Since THREDDS provides OGC WCS access, our end goal was to search THREDDS catalogs for WCS coverages in a geospatial client that could then seamlessly acquire the WCS coverage for display. In the course of (partially) reaching this goal, some lessons were learned with respect to CS/W interoperability, particularly with its sibling WCS protocol. Specifically, the CS/W protocol seems ambiguous in the type of WCS resource that should be referenced in the result: whether it is the server (exposed via GetCapabilities) or the coverage. Neither solution seems to fit well with supporting a subsequent WCS request. We hope these lessons will be useful to those developing CS/W and WCS clients. In addition, OGC interoperability architects may find the experiences documented below useful in future revisions to the WCS and/or CS/W protocols.

2 CS/W Server Implementation

CS/W provides catalog services for clients to find needed data and data-related services. CS/W specifies the interfaces, bindings, and a framework for defining application profiles required to publish and access digital catalogues for geospatial data and services. The CS/W specification does not require the use of a specific catalog schema. However, it encourages the adoption of standard schemas for maximum interoperability. Specifically, OGC developed two application profiles for CS/W: the ISO19115/19119 profiles and the electronic business Registry Information Model (ebRIM) profile. The ISO19115/19119 profile explains how catalog services based on this profile are organized and implemented for the discovery and management of geospatial data and service metadata that are compliant with the ISO19115 and 19119 standards. The ebRIM profile explains how services based on the more general Organization for the Advancement of Structured Information Standards (OASIS) ebXML Registry Information Model are organized and implemented. Our CS/W server is compliant with the OpenGIS Catalog Services Specification 2.0.2 -ISO Metadata Application Profile[1]. It specifies an application profile for ISO 19115/ISO 19119 metadata with support for XML encoding per ISO/TS19139 and HyperText Transfer Protocol (HTTP) binding. Currently, it supports OGC_Service.GetCapabilities, CSW Discovery.GetRecords, CSW Discovery.DescribeRecord, and CSW Discovery.GetRecordById. A CS/W client starts by sending a *GetCapabilities* request to the server and getting a response that describes the capability of CS/W server. The client then constructs a GetRecords request, following the specification, based on the inputs of the user who is using the client, and sends the request to the server. Based on the GetRecords request, the server sends a response XML back to the client.

For performance reasons, our implementation of the CS/W server for THREDDS database relied on ingesting the THREDDS catalog into a relational database for efficient querying. This process also allowed a mapping of the THREDDS metadata to ISO 19115, using a metadata mapping scheme. That is, the ingestor reads the THREDDS catalog and converts related metadata items into ISO19115 counterparts based on this mapping scheme and ingests them into the CS/W server database. The CS/W server provides either real-time or pre-stored THREDDS catalog information to the clients, using the following process:

1. The ingestor ingests THREDDS catalog information into the CS/W server on a pre-configured schedule (or on demand, as required).
2. The server receives a valid CS/W request from a CS/W client.
3. The server searches the CS/W database which has been pre-populated with ingested THREDDS catalog information.
4. The server sends the translated response to the requesting CS/W client.

The Ingestor includes four components: Ingesting, Parsing, Mapping and Registration (Fig 1).

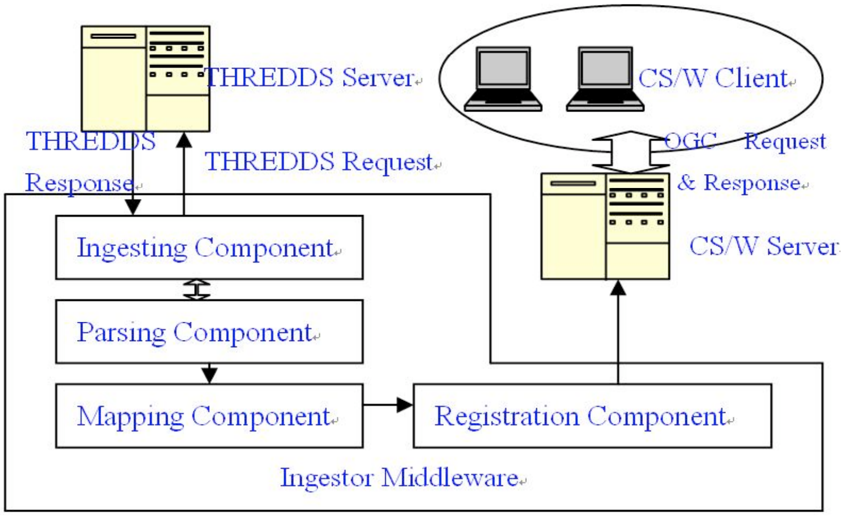


Fig 1. Diagram of CS/W server components.

The Ingesting component is itself a THREDDS client: it sends a request to the THREDDS server and hands the response off to the Parsing component. The Parsing component is responsible for parsing the catalog content. Because the THREDDS catalog is hierarchical, there are two types of datasets encountered: direct datasets and dataset collections. A dataset without a nested *dataset* is a direct dataset, whereas a *catalogRef* element is recognized as a dataset collection. THREDDS *catalogRef* elements are used to implicitly include other catalogs simply by pointing to that catalog's URL, allowing for both nested catalogs as well as highly distributed catalogs. If a *catalogRef* element is found in the catalog document, a new catalog XML URL reference is generated, which is handed back to Ingesting component, triggering a recursive request to the server for the corresponding new catalog XML document. If a *dataset* element is found, the metadata are passed to the Mapping component. The Mapping component translates the metadata from the schema of THREDDS InvCatalog Markup Language (describing THREDDS inventory) to the schema that compliant with ISO 19115. Finally, the Registration component registers the metadata it into the CS/W database.

One THREDDS catalog may, and often does, contain many thousands of individual datasets. The Ingestor was designed to do hierarchical ingestion. It ingests a small number of more stable data collections on a pre-configured, regular and relatively longer time period schedule, such as daily and weekly. The more frequently updated datasets are ingested on shorter cycles, such as hours or minutes. In our current integration with the Unidata THREDDS catalog, we also included on-demand ingesting for datasets, i.e., ingesting only when a client wants to search inside a particular data collection, at which point the datasets under the needed data collection are ingested in real-time.

3 Interoperability Challenges

The OGC CS/W interface can work with a variety of metadata profiles. Thus, a catalog system implementing the OGC CS/W interface standard must also implement a metadata profile. The common metadata profiles being used with the CS/W include the ebRIM family and the ISO 19115 family. Because more than one metadata profile can be used with the OGC CS/W standard, interoperability between a CS/W Server and an independently developed Client is difficult. For meaningful, basic searches to be performed, the CS/W client and server must comply with the same mandatory set of metadata attributes. For comprehensive, detailed searches to be performed, the CS/W client and server must agree on a similar set of optional metadata attributes. In practice, this requires coordination and agreement between the CS/W client and server developers.

4 Deep Dataset Hierarchies

THREDDS servers nearly always provide access to collections of datasets, allowing for catalogs of catalogs. For example, the Unidata “**motherlode**” prototype server serves real-time output of several different weather forecast model runs from the National Centers for Environmental Prediction (NCEP), among them the Rapid Update Cycle (RUC), North American Model (NAM), and Global Forecast System (GFS) models. Each of these models is run at regular intervals ranging from every hour to twice a day. THREDDS catalogs are available at various levels:

- Top level catalogs the different types of datasets available (e.g., NCEP model output, radar, station obs, radar scans, satellite images) at http://motherlode.ucar.edu:8080/thredds/catalog.html
- In the NCEP branch, the next level catalogs different models (e.g., RUC, NAM, GFS) at http://motherlode.ucar.edu:8080/thredds/idd/models.html
- The next level catalogs the output options for a specific model such as the NCEP-NAM-CONUS_12km-conduit (e.g. forecast model run or individual "file access") at http://motherlode.ucar.edu:8080/thredds/catalog/fmrc/NCEP/NAM/CONUS_12km/conduit/catalog.html

- For the "file access," there is an inventory list of the Gridded Binary (GRIB) files with about 90 such files on **motherlode** for NCEP-NAM-CONUS_12km-conduit runs over about 3 weeks at http://motherlode.ucar.edu:8080/thredds/catalog/fmrc/NCEP/NAM/CONUS_12km/conduit/files/catalog.html
- For any one of those NCEP-NAM-CONUS_12km-conduit runs, there are four access method options (OPeNDAP, HTTPServer, WCS, NetcdfSubset) at http://motherlode.ucar.edu:8080/thredds/catalog/fmrc/NCEP/NAM/CONUS_12km/conduit/files/catalog.html?dataset=fmrc/NCEP/NAM/CONUS_12km/conduit/files/NAM_CONUS_12km_20080820_0600.grib2

This hierarchy of catalogs begs the question of which level should be harvested for discovery in the CS/W search facility. At the top level, there are only seven categories of data in the catalog on **motherlode**. However, these branch out into thousands of inventory leaf node possibilities. If one chooses to list the catalogs at one of the higher collection levels for searching in the CS/W in order to avoid returns with hundreds of hits, one must be able to drill down from that level to the data access level where a data access interface such as WCS is available. Furthermore, even if this capability is provided in the CS/W server, it is not useful unless clients are able to take advantage of it and make it possible for the user to drill down via the client interface.

As noted below in the "Last Mile Problem" section, there is the added challenge that users may actually be searching for specific fields (e.g., vorticity) in any given forecast. This compounds the granularity issue in that the various models output many dozens of fields in each data file. If each such field in each dataset is viewed as a separate coverage, one can envision a simple CS/W query that would result in thousands or tens of thousands of hits in terms of individual coverages for different model runs and forecast times on a given server.

5 Freetext Search

Freetext searches are by far the most common search types on the Internet today (cf. Google). One of the keys to their popularity is the utter simplicity: there is no data model to learn in order to use it, just a single field. However, the CS/W search/response protocol is highly structured, with a number of fields containing potentially searchable text. As a result, there is much ambiguity as to which of these many fields should be searched by the client when presenting a simple freetext field to the user. Some clients (e.g. ESRI) search only the title; others search the abstract, and still others may search multiple fields with the same keywords. As a result, the same keywords in two separate clients may generate different results even when submitted to the same server.

6 The “Last Mile” Problem

The ultimate goal is to return a search result that a client can use to issue a Web Coverage Service request, thus taking advantage of the WCS capabilities of the THREDDS Data Server. This should be relatively transparent to the user, without the need to cut and paste URLs. The THREDDS WCS server is implemented at the dataset level, which means for each THREDDS dataset, a WCS script is provided. Because a THREDDS dataset may contain many variables each of which is offered as a coverage, a single WCS request URL using the dataset name as coverage name cannot be directed to retrieve a specific variable. (Restructuring the catalog to inventory individual variables would have increased the catalog size manyfold, producing scalability problems and would have been out of scope of this project.) Rather, the THREDDS server refers such a WCS URL to the *GetCapabilities* request of the server script. Therefore in the initial CS/W server implementation, the THREDDS metadata were inventoried at a server level, meaning that the CS/W server provided enough information for the client to issue a WCS *GetCapabilities* request, followed eventually by a *GetCoverage* request. However, the *GetCapabilities* response overlaps significantly with the CS/W response in information content, forcing the client to essentially repeat a significant amount of search work. For example, if the user searches on, say, "*vorticity*", this response would force the user or the client to search through the *GetCapabilities* response (or a subsequent *DescribeCoverage* response) in order to find the *vorticity* coverage included, (most likely together with many other coverages), in a THREDDS dataset. One alternative is to instead inventory at a Coverage level, so that a client could immediately issue a *GetCoverage* request to a specific coverage. One possibility, suggested by Enrico Boldrini and Lorenzo Bigagli of the Earth and Space Science Informatics Laboratory (developers of the GI-go client), is to include sufficient information in the fields of the ISO 19115 *CI_OnlineResource* element to allow the client to issue a specific direct *GetCoverage* request without having to use additional information, as the following:

```
-linkage -> the WCS endpoint (without request parameters)

-protocol -> something like "OGC:WCS-1.0.0" or "OGC:WCS-1.1.1-get-
coverage", following the "style" NAMESPACE:PROTOCOL-VERSION-OPERATION used
in the examples from OGC:
http://schemas.opengis.net/csw/2.0.2/profiles/apiso/1.0.0/examples-
ISO19139/collectiondata4ESA.xml and in GeoNetwork:
http://trac.osgeo.org/geonetwork/wiki/ISO19119impl .

-name -> the name of the coverage

-description -> brief description of the coverage

-function -> download
```

Currently, the WCS access URLs in the THREDDS catalog include *GetCapabilities* but not *GetCoverage* requests. Thus a WCS client component was implemented in the CS/W server To act as a gateway. This WCS client component returns the coverage information using the following steps:

a) send a *GetCapabilities* request for a specific dataset using the WCS URL provided by THREDDS and parse the *GetCapabilities* response;

b) based on the *GetCapabilities* response, construct a *DescribeCoverage* request and parse the response

c) match the clients CS/W search criteria to the metadata of the coverages to filter out those coverages that are not needed by the client; and finally

d) construct one or more *CI_OnlineResources* element, each containing a direct *GetCoverage* request to the matched coverage.

With this information, the user of a CS/W client can directly download specific variable(s) in a multi-variable dataset. This idea was verified through the GI-go client, which has the ability to make both CS/W and WCS requests. Using the GI-go client, we were able to conduct a CS/W query and use the results, together with user-input spatial constraints, to issue a WCS *GetCoverage* request. In this case, the resulting data were saved as a file and visualized in an external tool (the Integrated Data Viewer). Screenshots of this process are included in Appendix B. Ultimately, the goal would be to issue the catalog query, acquire the coverage, and visualize (or even operate on) the data all in a single client.

7 Conclusion

Ultimately, the OGC Geoscience gateway was successful at providing a CS/W interface to the THREDDS catalogs, which, in addition to protocol interoperability, enable a user to both browse and search THREDDS data holdings. Perhaps even more important, however, the gateway was able to provide both catalog (CS/W) and coverage (WCS) access through a single client (the GI-go client), making the critical link from catalog discovery to data and service access. In the process, this surfaced a discontinuity or overlap between our implementation of the CS/W catalog search and the THREDDS Data Server implementation of the WCS *GetCapabilities* protocols. This is rooted in an apparent ambiguity in the CS/W specification with respect to how WCS service access points should be returned in results. Returning simply the service endpoint forces the client to essentially “repeat” the search by issuing subsequent GetCapabilities and DescribeCoverage requests and then searching within them. Returning the Coverage name in addition would short-circuit this, but there does not appear to be a documented standard location for this. In the meantime, clients must make ad hoc arrangements, with potential divergence and suboptimal results. On the other hand, solving this mismatch problem within the OGC protocol family could provide a boost to both CS/W and WCS clients, which are not quite so abundant as one might expect at this stage.

8 Informative References

[1] OpenGIS Catalogue Services Specification 2.0.2 -ISO Metadata Application Profile

9 Authors

The OGC Geoscience Gateway Project Team:

Christopher Lynnes, NASA GSFC, chris.lynnes@nasa.gov

Wenli Yang, George Mason University, wyang1@gmu.edu

Chengfang Hu, George Mason University, chu11@gmu.edu

Ben Domenico, NCAR/UCAR, ben@unidata.ucar.edu

Yonsook Enloe, SGT, yonsook@mindspring.com

Appendix A - Glossary

ACCESS - Advancing Collaborative Connections for Earth System Science

CS/W – Catalog Services for the Web

CEOS – Committee on Earth Observation Satellites

ebRIM – electronic business Registry Information Model

ebXML – electronic business eXtensible Markup Language

ESDS – Earth Science Data Systems

GFS – Global Forecast System

GRIB – Gridded Binary

HTTP – HyperText Transfer Protocol

NAM – North American Model

NCEP – National Centers for Environmental Prediction

OASIS – Organization for the Advancement of Structured Information Standards

OGC – Open Geospatial Consortium

OPeNDAP – Open-source Project for a Network Data Access Protocol

RUC – Rapid Update Cycle

THREDDS – Thematic Realtime Environmental Distributed Data Services

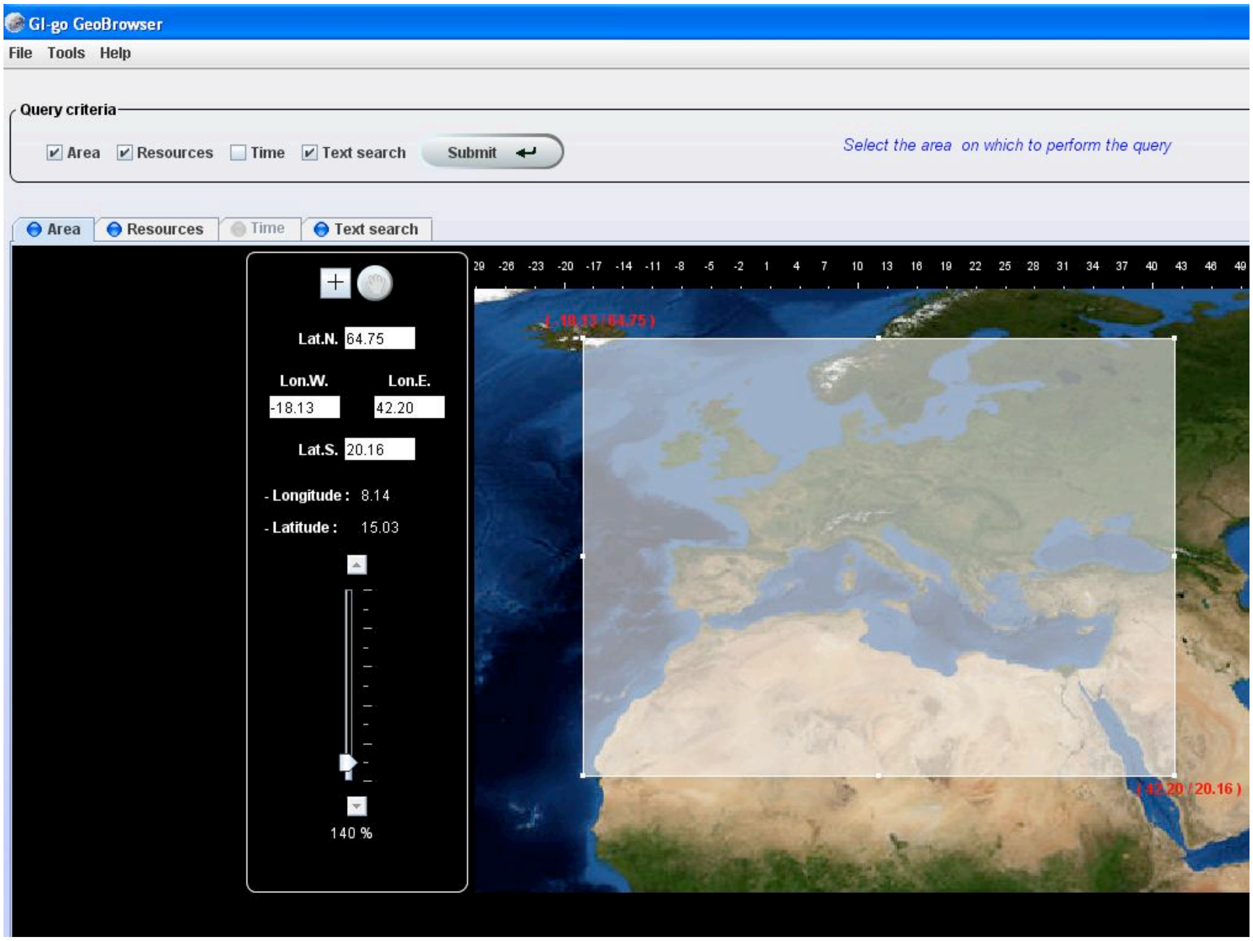
URL – Universal Reference Locator

WCS – Web Coverage Service

WGISS – Working Group on Information Systems and Services

XML – eXtensible Markup Language

Appendix B – Screenshot sequence showing CS/W search and WCS download in the GI-go client.



☑ Area ☑ Resources □ Time ☑ Text search

Select the area on which to perform the query

Lat.N. 64.75

Lon.W. -18.13 Lon.E. 42.20

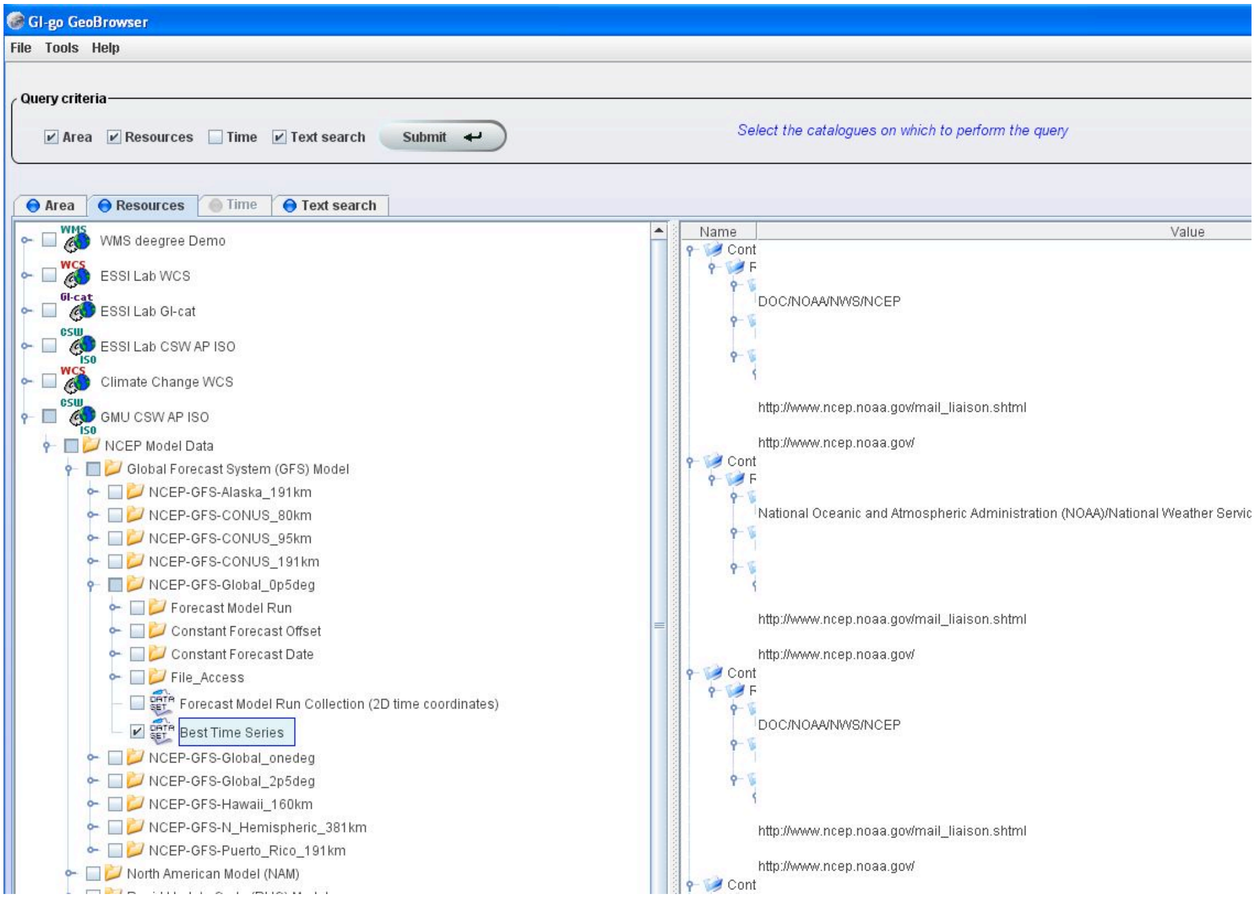
Lat.S. 20.16

- Longitude : 8.14

- Latitude : 15.03

140 %

Fig 1 Area (first tab) is specified on a map.

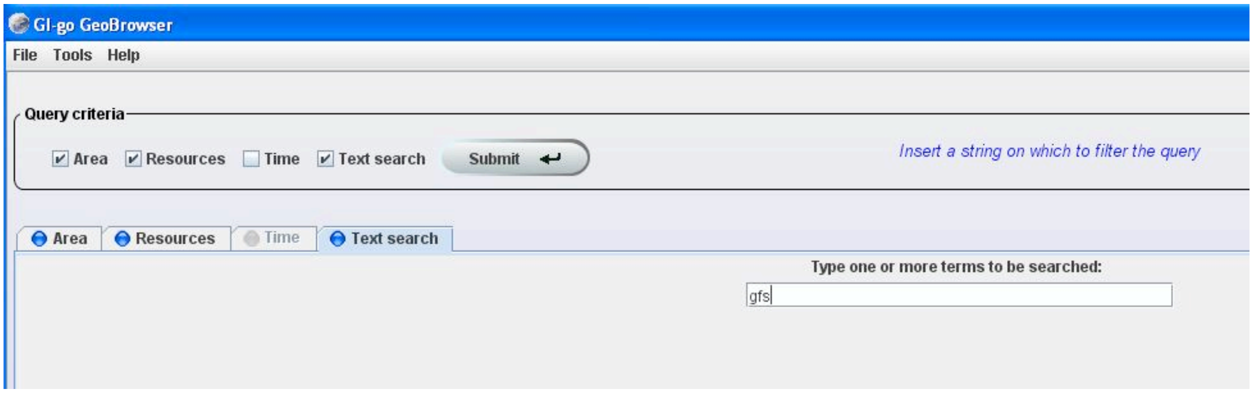


Query criteria

☑ Area ☑ Resources □ Time ☑ Text search

GI-go GeoBrowser
File Tools Help
Submit
Select the catalogues on which to perform the query
Area
Resources
Time
Text search
WMS deegree Demo
ESSI Lab WCS
ESSI Lab GI-cat
ESSI Lab CSW AP ISO
Climate Change WCS
GMU CSW AP ISO
NCEP Model Data
Global Forecast System (GFS) Model
NCEP-GFS-Alaska_191km
NCEP-GFS-CONUS_80km
NCEP-GFS-CONUS_95km
NCEP-GFS-CONUS_191km
NCEP-GFS-Global_0p5deg
Forecast Model Run
Constant Forecast Offset
Constant Forecast Date
File_Access
Forecast Model Run Collection (2D time coordinates)
☑ Best Time Series
NCEP-GFS-Global_onedeg
NCEP-GFS-Global_2p5deg
NCEP-GFS-Hawaii_160km
NCEP-GFS-N_Hemispheric_381km
NCEP-GFS-Puerto_Rico_191km
North American Model (NAM)
Name
Value
Cont
DOC/NOAA/NWS/NCEP
http://www.ncep.noaa.gov/mail_liaison.shtml
http://www.ncep.noaa.gov/
Cont
National Oceanic and Atmospheric Administration (NOAA)/National Weather Servic
http://www.ncep.noaa.gov/mail_liaison.shtml
http://www.ncep.noaa.gov/
Cont
DOC/NOAA/NWS/NCEP
http://www.ncep.noaa.gov/mail_liaison.shtml
http://www.ncep.noaa.gov/
Cont

Fig 2. Resource (i.e., dataset grouping) is specified. This is a drilldown into the THREDDS hierarchy.

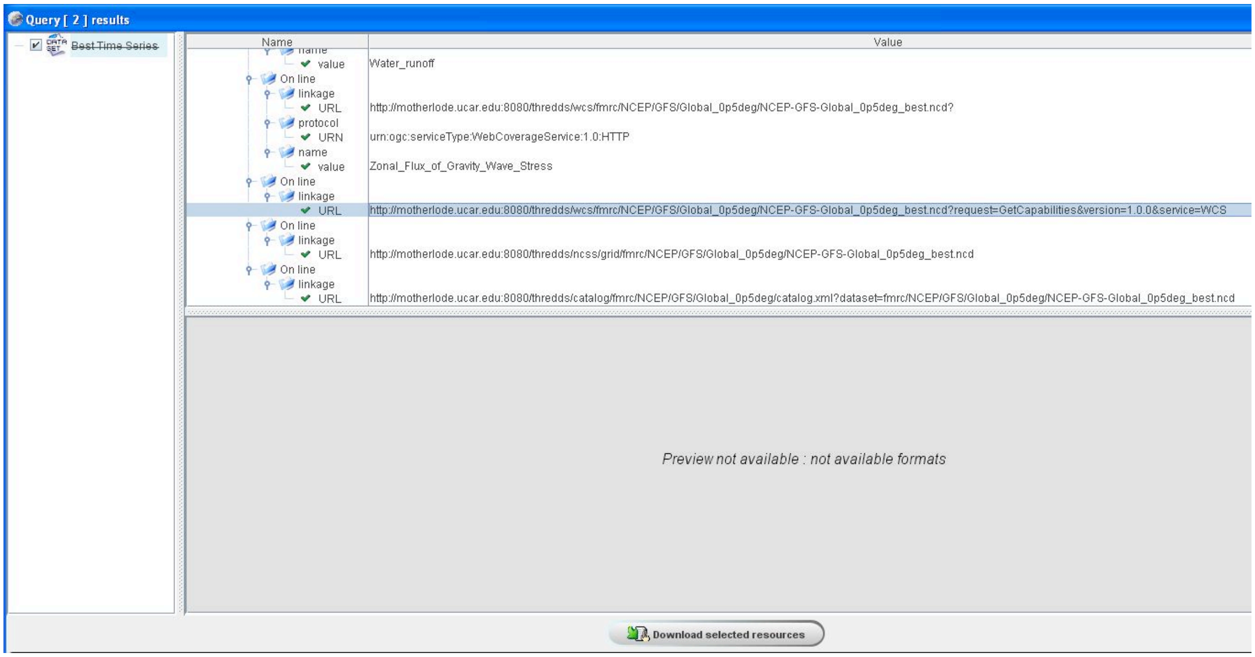


Query criteria

☑ Area ☑ Resources □ Time ☑ Text search

Type one or more terms to be searched: gfs

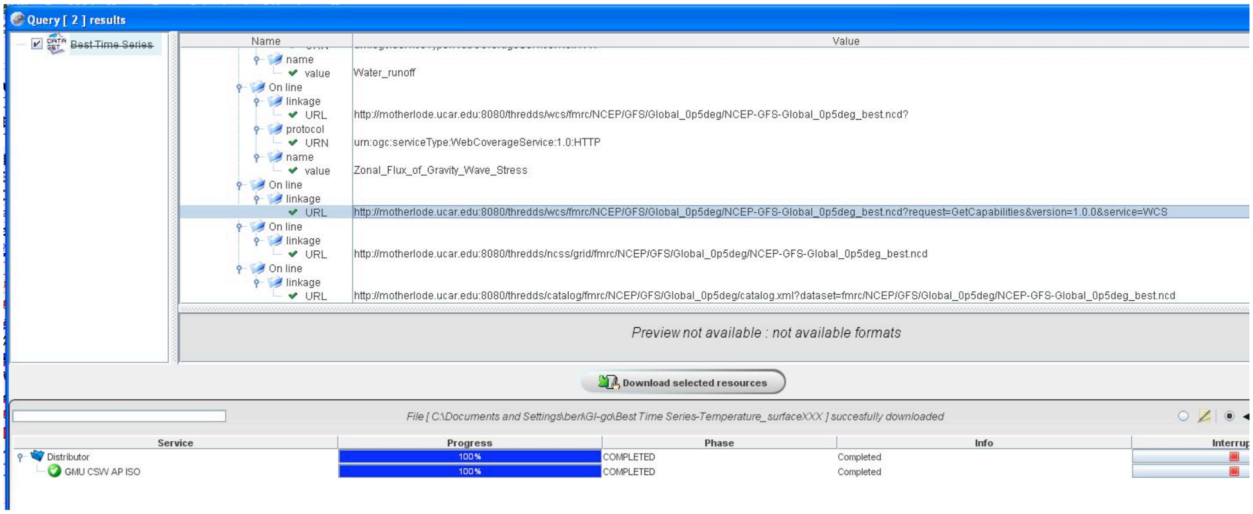
Fig 3 Search keyword ("gfs") is specified.



Name	Value
name	
value	Water_runoff
On line	
linkage	
URL	http://motherlode.ucar.edu:8080/thredds/wcs/fmrc/NCEP/GFS/Global_0p5deg/NCEP-GFS-Global_0p5deg_best.ncd?
protocol	
URN	urn:ogc:serviceType:WebCoverageService:1.0:HTTP
name	
value	Zonal_Flux_of_Gravity_Wave_Stress
On line	
linkage	
URL	http://motherlode.ucar.edu:8080/thredds/wcs/fmrc/NCEP/GFS/Global_0p5deg/NCEP-GFS-Global_0p5deg_best.ncd?request=GetCapabilities&version=1.0.0&service=WCS
On line	
linkage	
URL	http://motherlode.ucar.edu:8080/thredds/ncss/grid/fmrc/NCEP/GFS/Global_0p5deg/NCEP-GFS-Global_0p5deg_best.ncd
On line	
linkage	
URL	http://motherlode.ucar.edu:8080/thredds/catalog/fmrc/NCEP/GFS/Global_0p5deg/catalog.xml?dataset=fmrc/NCEP/GFS/Global_0p5deg/NCEP-GFS-Global_0p5deg_best.ncd

Preview not available : not available formats

Fig 4 Query results, showing URLs for online WCS resource.



Name	Value
name	
value	Water_runoff
On line	
linkage	
URL	http://motherlode.ucar.edu:8080/thredds/wcs/fmrc/NCEP/GFS/Global_0p5deg/NCEP-GFS-Global_0p5deg_best.ncd?
protocol	
URN	urn:ogc:serviceType:WebCoverageService:1.0:HTTP
name	
value	Zonal_Flux_of_Gravity_Wave_Stress
On line	
linkage	
URL	http://motherlode.ucar.edu:8080/thredds/wcs/fmrc/NCEP/GFS/Global_0p5deg/NCEP-GFS-Global_0p5deg_best.ncd?request=GetCapabilities&version=1.0.0&service=WCS
On line	
linkage	
URL	http://motherlode.ucar.edu:8080/thredds/ncss/grid/fmrc/NCEP/GFS/Global_0p5deg/NCEP-GFS-Global_0p5deg_best.ncd
On line	
linkage	
URL	http://motherlode.ucar.edu:8080/thredds/catalog/fmrc/NCEP/GFS/Global_0p5deg/catalog.xml?dataset=fmrc/NCEP/GFS/Global_0p5deg/NCEP-GFS-Global_0p5deg_best.ncd

Preview not available : not available formats

File [C:\Documents and Settings\ben\GI-go\Best Time Series-Temperature_surfaceXXX] succesfully downloaded

Service	Progress	Phase	Info	Interrup
Distributor	100%	COMPLETED	Completed	
GMU CSW AP ISO	100%	COMPLETED	Completed	

Fig 5 The client obtains coverage information from the selected WCS resource.

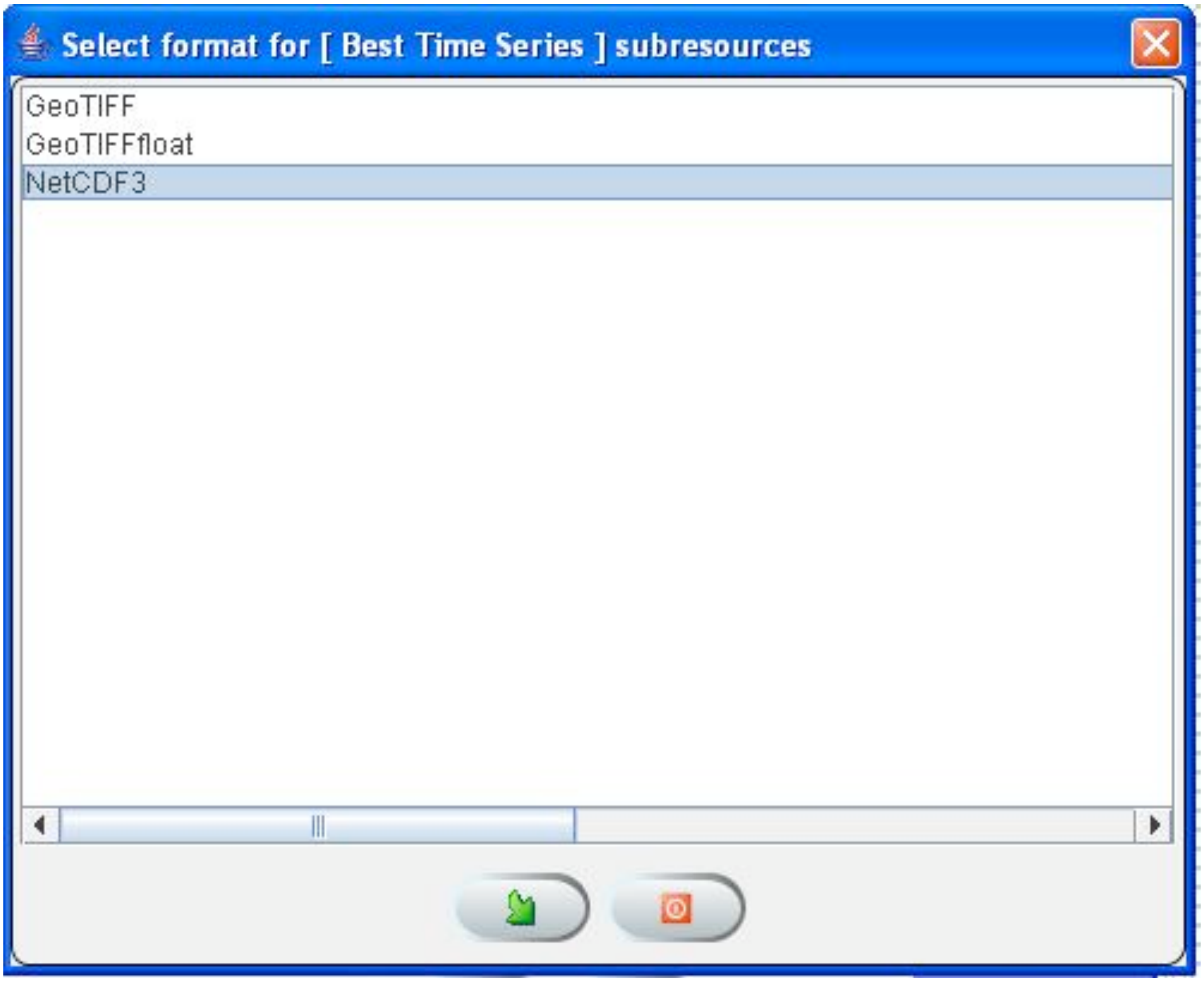
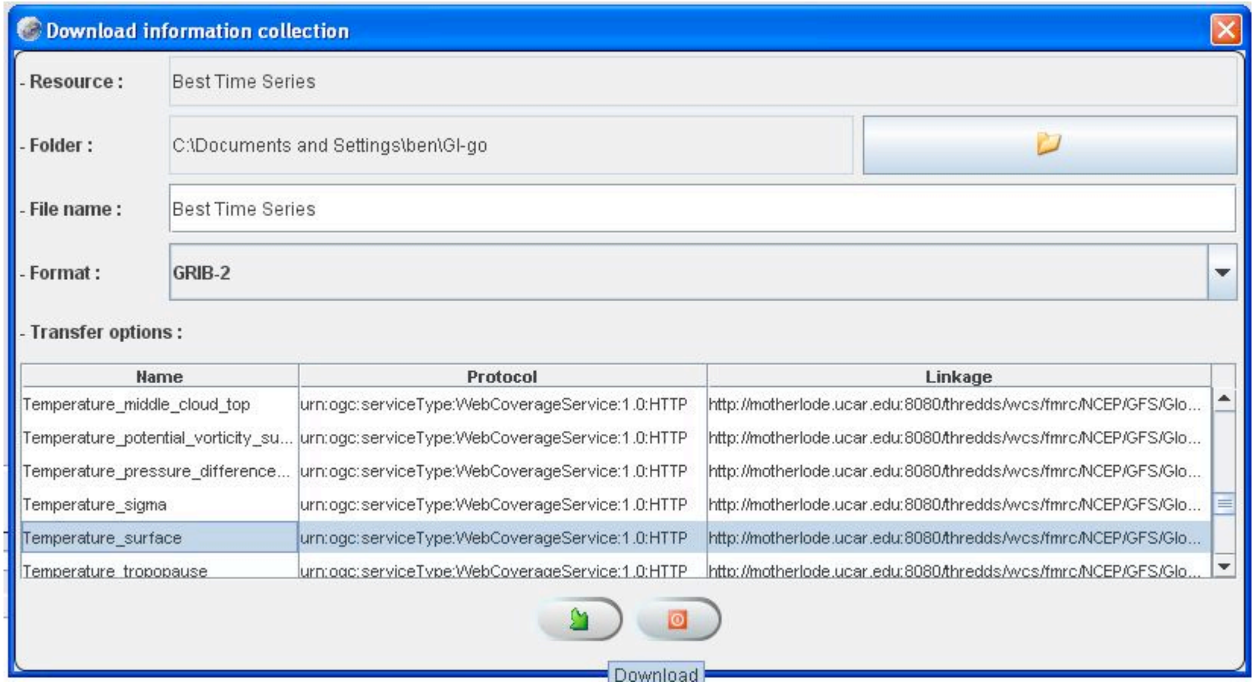


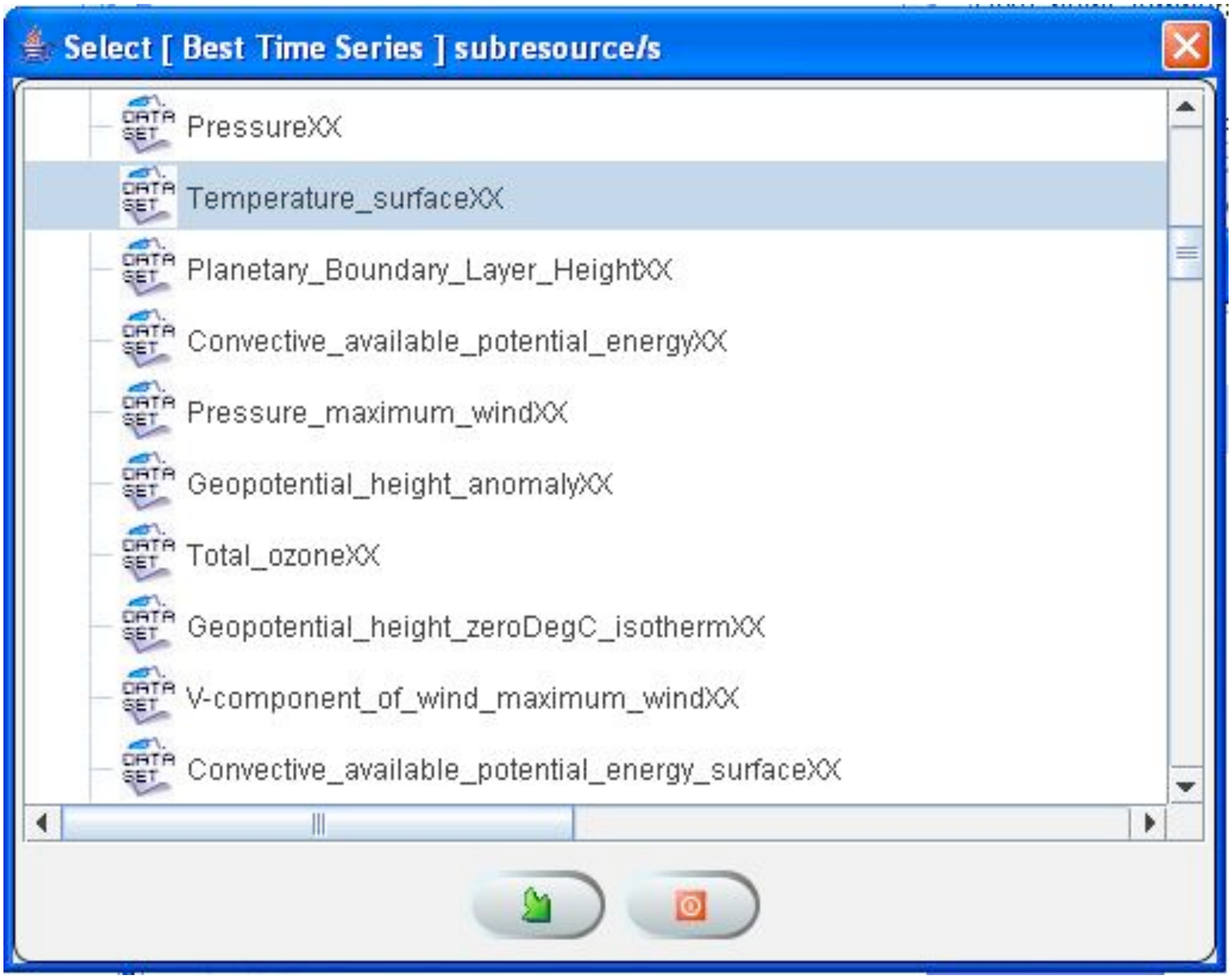
Fig 6. Various WCS profile choices are displayed from the WCS server referenced in the above screen.



- Resource : Best Time Series
- Folder : C:\Documents and Settings\ben\GI-go
- File name : Best Time Series
- Format : GRIB-2
- Transfer options :

Name	Protocol	Linkage
Temperature_middle_cloud_top	urn:ogc:serviceType:WebCoverageService:1.0:HTTP	http://motherlode.ucar.edu:8080/thredds/wcs/fmrc/NCEP/GFS/Glo...
Temperature_potential_vorticity_su...	urn:ogc:serviceType:WebCoverageService:1.0:HTTP	http://motherlode.ucar.edu:8080/thredds/wcs/fmrc/NCEP/GFS/Glo...
Temperature_pressure_difference...	urn:ogc:serviceType:WebCoverageService:1.0:HTTP	http://motherlode.ucar.edu:8080/thredds/wcs/fmrc/NCEP/GFS/Glo...
Temperature_sigma	urn:ogc:serviceType:WebCoverageService:1.0:HTTP	http://motherlode.ucar.edu:8080/thredds/wcs/fmrc/NCEP/GFS/Glo...
Temperature_surface	urn:ogc:serviceType:WebCoverageService:1.0:HTTP	http://motherlode.ucar.edu:8080/thredds/wcs/fmrc/NCEP/GFS/Glo...
Temperature_tropopause	urn:ogc:serviceType:WebCoverageService:1.0:HTTP	http://motherlode.ucar.edu:8080/thredds/wcs/fmrc/NCEP/GFS/Glo...

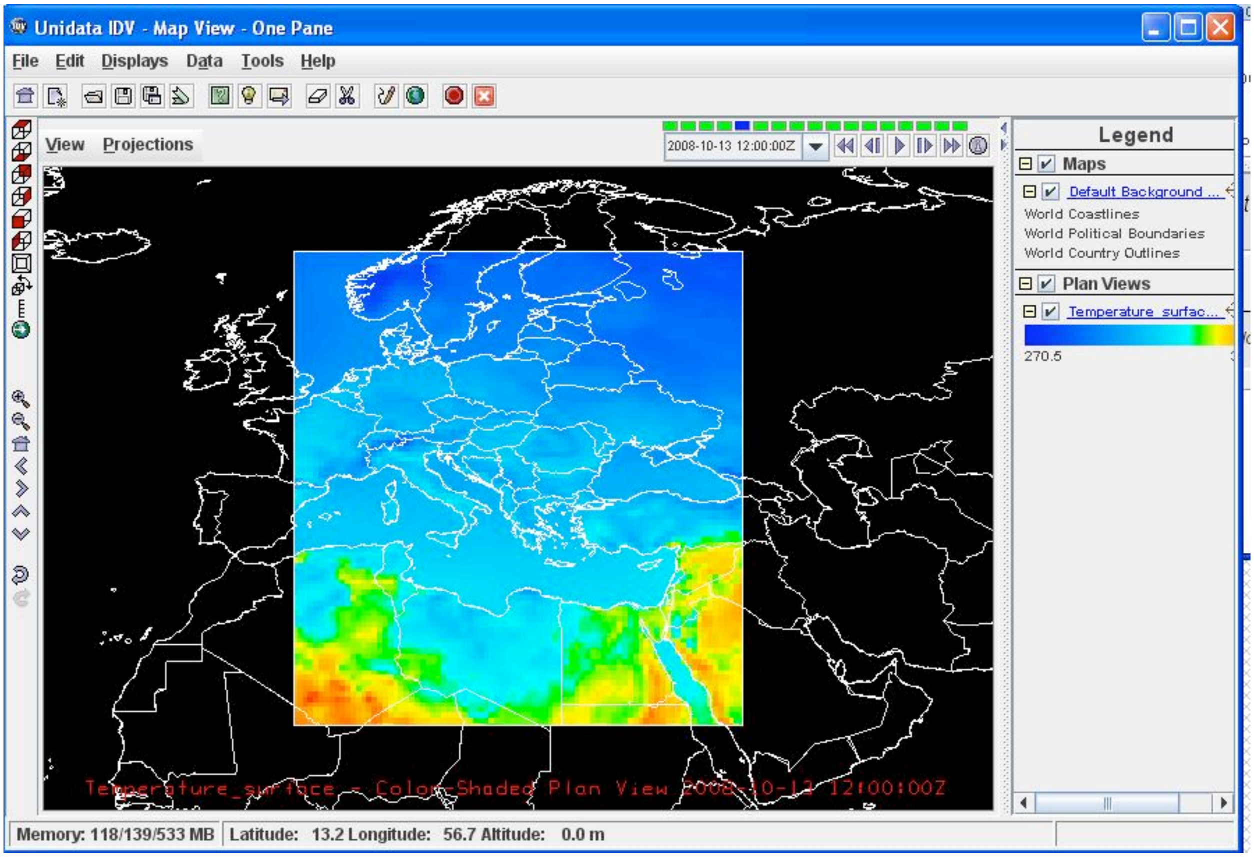
Fig 7 The user chooses from multiple parameters, all available from the same WCS server for this data file.



Select [Best Time Series] subresource/s

- PressureXX
- Temperature_surfaceXX
- Planetary_Boundary_Layer_HeightXX
- Convective_available_potential_energyXX
- Pressure_maximum_windXX
- Geopotential_height_anomalyXX
- Total_ozoneXX
- Geopotential_height_zeroDegC_isothermXX
- V-component_of_wind_maximum_windXX
- Convective_available_potential_energy_surfaceXX

Fig 8. The screen to execute a WCS request to download the actual coverage data.



Memory: 118/139/533 MB Latitude: 13.2 Longitude: 56.7 Altitude: 0.0 m

Fig 9 Integrated Data Viewer display of the coverage results obtained from the THREDDS WCS server.