

EGI Programmatic Access Documentation

Programmatic Access of EOSDIS DAAC Hosted Services

User Documentation

- [Overview](#)
 - [What is EGI Programmatic Access?](#)
 - [The EGI Request](#)
 - [The EGI Response](#)
- [Using curl to Communicate With EGI](#)
 - [The Curl Program](#)
 - [Example Shell Script](#)
- [Earthdata Login Integration](#)
 - [Establishing an Earthdata Login Session](#)
- [Programmatic Access Parameters](#)
 - [CMR Parameters](#)
 - [EGI Parameters](#)
 - [Administrative and Formatting Parameters](#)
- [EGI Programmatic Access Usage Examples](#)
 - [Scenario 1 SPL3SMP Spatial and Parameter Subsetting and Reformatting to GeoTIFF](#)
 - [Scenario 2 SPL2SMA Reformatting to GeoTIFF](#)
 - [Scenario 3 GLAH12 Spatial and Parametric Subsetting](#)
 - [Scenario 4 MOD10A1 Reformatting to GeoTIFF and spatial and parameter subsetting](#)
 - [Scenario 5 MOD10A1 No Processing returning raw data with no customization services applied](#)
 - [Scenario 6 ICESat-2 ATL08 Temporal Subset](#)
 - [Scenario 7 ICESat-2 ATL06 Polygon Subset - KML](#)
 - [Scenario 8 ICESat-2 ATL06 Polygon Subset - Shapefile](#)
 - [Scenario 9 ICESat-2 ATL03 Polygon Subset - GeoJSON](#)
 - [Scenario 10 SMAP SPL4SMAU Polygon Subset - "CMR-like" Coordinate Pairs](#)
- [Retrieving Request Results](#)
- [Error Responses](#)
- [EGI Programmatic Access Configuration and Hardening](#)
- [References](#)

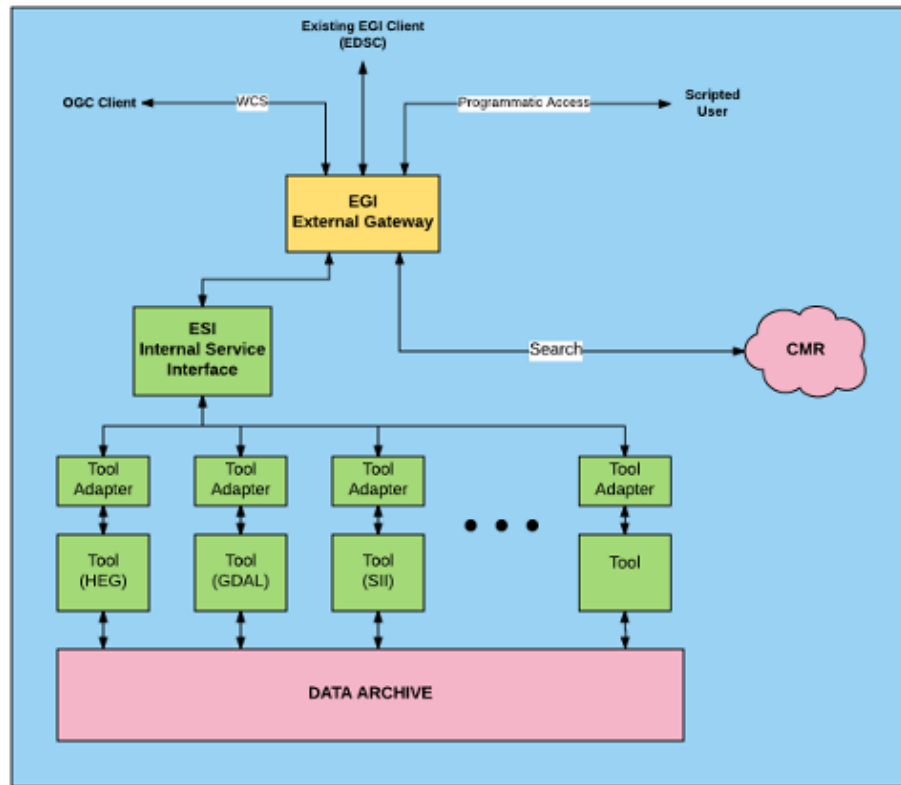
Overview

What Is EGI Programmatic Access ?

Programmatic Access is a capability enhancement to the Data Access services at EOSDIS Service Interface (**ESI**) enabled DAACs. (The ESI enabled DAACs are NSIDC, LP-DAAC and ASDC.)

As seen in the diagram below, it adds an ability for **EGI** (the *ESI Gateway Interface*) to access **CMR** (the *Common Metadata Repository*), and extends the accessible protocols to include **WCS** (*Web Coverage Service*) compatibility. This improves the scriptability of the EGI component, which is the exposed user program interface into the data access services.

The Programmatic Access interface is used to locate and access DAAC hosted data, optionally performing services on the data. It combines into one interface functions that formerly required multiple interfaces: searching CMR for science granules of interest, and then obtaining the science data files with optional services applied. The interface supports synchronous, streaming synchronous, and asynchronous modes of operation. **The Programmatic Access Interface requires** [client authentication through Earthdata Login](#).



The EOSDIS Services Gateway Interface Context

The EGI Request

EGI Programmatic Access is exposed as a REST interface using the HTTPS protocol. It takes the form of an HTTPS URL, containing a series of key-value-pairs (KVPs) as query parameters that specify the operands and the operations performed. The URL incorporates these elements:

- the EGI endpoint
- KVPs that control a CMR search for granules (collection, time, spatial constraints)
- KVPs that identify the services to be performed for each granule (subsetting, reformatting, etc)
- KVPs to provide administrative control and information (token, paging)

Example URL:

```
https://n5eil02u.ecs.nsidc.org/egi/request?
  short_name=MOD10A1&version=006
  &time=2016-07-01,2016-07-02
  &format=GeoTIFF
```

This URL generates a request that accesses the egi services at the NSIDC DAAC (<https://n5eil02u.ecs.nsidc.org/egi/>) as follows:

```
short_name=MOD10A1&version=006
```

These two parameters tell EGI to call CMR to perform a query and search for granules from the MOD10A1 version 6 collection.

```
time=2016-07-01,2016-07-02
```

This adds a temporal clause to the CMR query, restricting it to granules obtained on July 1, 2016.

```
format=GeoTIFF
```

This specifies that the data should be returned in GeoTIFF format.

This request and its response can be communicated with the DAAC using any HTTP client program, including browsers, command line utilities, and custom programs. A flexible command line utility named “curl” is commonly used for this purpose. The examples on this page show how to use curl to request programmatic access services and receive the resulting output files. Alternatively, a user can write a custom program in any language to implement their desired workflow, using a curl library or the native http communication functions of the language. An example demonstrating programmatic access of CMR over HTTP written in Python can be found here: <https://git.earthdata.nasa.gov/projects/HDS/repos/cmr/>. This can be readily extended to encompass the EGI Programmatic Access functionality described herein.

The EGI Response

The response returned by the EGI depends upon the request parameters. By default, a request is submitted in 'streaming synchronous' mode. This means that a response will not be returned to the client until the request has finished processing, and the results of the request will be returned directly to the client - usually a zip file containing all the resulting granules.

As with the streaming synchronous mode, the regular 'synchronous' mode will only return a response to the client once the request has completed processing, and the response will be an XML formatted status that includes URLs that can be used to download the resulting data,

In 'synchronous' mode, a response is returned to the client immediately upon receipt of the request. An email will be sent upon completion of the request to the user's email account, providing URLs that can be used to download the resulting data files.

Requests that invoke CMR queries but do not otherwise specify any form of granule processing or reformatting will return the XML formatted CMR response directly to the client.

Using curl to Communicate With EGI

The Curl Program

Curl is an extensive program (and library) that supports many internet protocols and options. Here we are focusing only on the HTTP protocol and options that are useful for EGI programmatic access. For more documentation, see the curl man page on your system.

Some of the program options used below require curl version 7.20.0 (2010) or later. Note that Red Hat Release 6.8 includes curl 7.19.7. A newer version is required. We recommend the latest stable version. We have developed these examples using curl 7.49.1 on Red Hat and curl 7.43.0 on Mac OS X 10.11.6.

Curl usage for programmatic access is:

```
curl <options> <endpoint>/request?<parameters separated with &>
```

The following options have proven useful in developing these examples.

```
-v      verbose, prints internal details of performing the curl request, used for debugging
-s      silent, suppresses all unnecessary information output, used when not debugging
-L      follow the referral to a new location, automatically handles the Code 303 redirect

We do not use -L in the example script because the CMR-only query does
not provide a file name, causing curl to provide a default file name
built from the query url. Instead, we handle the redirect in the script logic.

-O      save the result in a local file
```

Example Shell Script

The example shell script implements this logic sequence:

- ## Earthdata Login Integration

A typical EGI/EDL request performing EDL user authentication [using curl](#) :

Establishing an Earthdata Login Session

When Earthdata Login is required for EGI access, EGI requests with very long URLs, for example, those using large shapefiles, can fail if your client program (e.g. curl, wget) has not already logged in and established a session. This is due to the nature of the OAuth2 authentication process when credentials must be provided. In this case, it is easy to establish a session prior to sending your request simply by sending an empty request as follows:

```
> curl -nL -b ~/.cookies -c ~/.cookies 'https://<host>/egi' -o /dev/null
```

If a session has not already been established, this will force curl to provide credentials to Earthdata Login and then save the session cookie. Subsequent requests will submit the session cookie and thus avoid the need for further authentication until the session expires. Session expiration is specific to each EGI instance but in most cases should be at least a few hours.

Note: The use of CMR tokens eliminates the need to authenticate via Earthdata Login, so the same URL length limitations do not exist.

Programmatic Access Parameters

Programmatic Access parameters, or key-value pairs (KVPs) are represented in a request as URL query parameters, separated with an ampersand '&'. KVPs can be EGI parameters, CMR parameters, or a mix of both. Any parameter not recognized as a valid EGI parameter is passed to CMR.

Additionally, EGI also provides an [OGC Web Coverage Service \(WCS\)](#) compliant service interface that allows WCS compatible clients access to DAAC hosted ECS data.

The following table gives some useful CMR parameters; however any [CMR supported parameter](#) can be used.

Useful CMR parameters: (WCS Endpoint Parameters)

Short_name=aaaa	Specifies the short name of the collection used to find granules for the coverage requested. Can be used multiple times to return granules from multiple collections
Version=nnn	<p>Specifies collection version. The version is treated like a string and must match the version field for that collection in CMR. Multiple versions can be specified.</p> <p>Note: the Version parameter is also used by WCS clients to specify the version of WCS. When version=1.0.0, EGI understands it to be specifying the WCS versions. Otherwise Version is passed to CMR as a search parameter.</p> <p>Discussion: The format of the version parameter depends on the the metadata that was provided to CMR when the collection was created. See the FAQ.</p>
Updated_since=<datetime>	<p>Can be used to find granules recently updated in CMR.</p> <p>Example datetime: 2016-09-01T12:00:00Z</p>
Bounding_box=nn,nn,nn	<p>This specifies a search filter to find only granules having a spatial extent that overlaps this bounding box, specified in decimal degrees of latitude and longitude.</p> <p>Order is lower left long, lower left lat, upper right long, upper right lat.</p> <p>This order is referred to as WSEN and is the same order as used for the EGI subsetting Bbox.</p>
Time=<datetime>,<datetime>	<p>Specify data datetime range filter for the CMR query. "Time" is the WCS compatible equivalent of the CMR "Temporal" parameter.</p> <p>Note, see the Time keyword for EGI described below for temporal subsetting usage.</p> <p>The start and end time values should be specified as a compound date and time: year-month-dayHours:minutes:secondsZ. Year-month-day is mandatory, the time part is optional. Year-month-day must use 4 digits for the year, two digits for the month and two digits for the day, separated by hyphens. If the time part is used, it must start with T and it must include three subfields for hours, minutes and seconds, separated with colons. The trailing Z is optional. It is standardly used to indicate GMT time, however all time parameters used in EGI Programmatic Access are GMT times. If the time part is not used, it is equivalent to T00:00:00.</p> <p>The "time=" KVP must always contain two datetime values to specify start and end of a time window, separated with a comma. Here are some valid examples:</p> <p>time=2016-01-01,2016-01-02</p> <p>time=2016-01-01T12:00:00Z,2016-01-01T18:00:00Z</p> <p>time=2016-01-01T00:00:00,2016-01-01T23:59:59</p>

sort_key [] =<sort-option>	This is a CMR parameter to control the sort order of the returned results. Sort options are described here in the CMR documentation . Example: sort granules by data coverage date in reverse order (newest first): sort_key[]=-start_date
----------------------------------	--

Useful EGI parameters: (WCS GetCoverage Parameters)

Coverage=/group/sub-group/sub-sub-group/dataset	WCS: Used to specify the coverage to be processed. Specifies the subset data layer or group for Parameter Subsetting. Multiple datasets can be specified separated by commas. The dataset value always starts with a slash and the group-subgroup hierarchy are separated with slashes. If only a group or subgroup is specified, all lower level datasets are included in the processing.
Bbox=<W>,<S>,<E>,<N>	WCS: Bounding Box used for spatial subsetting. Coordinates are in decimal degrees. This is the same order as used for the CMR spatial filter (Bounding_box) parameter
Time=<start_datetime>,<end_datetime>	WCS Keyword: Specify data datetime range filter for the CMR query. Note, the Time keyword is a shared CMR/egi keyword that also invokes temporal subsetting/stitching for applicable data sets. WCS: Used for Temporal subsetting. Note: The Time keyword also is used as a CMR temporal filter.
SUBSET=time("<datetime>") SUBSET=lat(<min>,<max>) SUBSET=lon(<min>,<max>) SUBSET=x(<min>,<max>) SUBSET=y(<min>,<max>)	WCS: SUBSET=time("<datetime>") is used for Temporal subsetting with the datetime as the specified start time. WCS: Minimum and Maximum values to be used for the Bounding Box for spatial subsetting. Coordinates are in decimal degrees(lat/lon) or in meters(x/y).
Format=<format>	WCS: Optional output file format specifier used for re-formatting. Supported values vary by data type. [GeoTIFF, HDF-EOS5, NetCDF4-CF, NetCDF-3, ASCII, HDF-EOS, KML, Shapefile, TABULAR_ASCII] If this parameter is not used, then the output format is the same as the input format (no reformatting).
BoundingShape=<GeoJSON content> or <"CMR-like" coordinate pairs>	Optional, used for polygon subsetting. GeoJSON content string or "CMR-like" coordinate pairs (BoundingShape=lon1,lat1,lon2,lat2,...lonx,latx,lon1,lat1) Note: When drawing a polygon that crosses Anti-Meridian, need to redefine negative longitude values to > 180, vise versa, positive longitude values to < -180 in order for it to render correctly and for oge to convert it correctly to geojson format.
shapefile=@<KML/Shapefile>	Optional, used for polygon subsetting. [KML, Shapefile] Note: When drawing a polygon that crosses Anti-Meridian, need to redefine negative longitude values to > 180, vise versa, positive longitude values to < -180 in order for it to render correctly and for oge to convert it correctly to geojson format.
EMAIL	Specifies the email address to which any notifications will be sent. The special values yes and true are used to indicate that the email address should be obtained from the Earthdata Login user profile for the user submitting the request. These special values can only be used if the EGI is configured behind EDL and user credentials are provided OR a CMR token is submitted with the request.
FILE_IDS=<granule_id> <granule_UR>	Use of the FILE_IDS parameter will bypass CMR for the granule search and directly call EGI services using the SDPS granule id or granule UR (SC:<ESDT shortname>.<version_id>.<granule_id>)

Administrative and Formatting Parameters:

page_size=<n>	This is a CMR parameter to control the number of granules in the page of returned results. Discussion: When there are multiple granules returned from the CMR query, they are returned in sets called pages. The default page size is 10 granules. The page_size KVP allows the user to change the page size. Multi-page results can be accessed one page at a time using the page_num KVP described below. Note that the system configuration parameter MAX_GRANS_FOR_SYNC_REQUEST limits the size of a request in EGI. Exceeding that number returns an error. Page_size should always be set to less than or equal to this request limit. See the FAQ for more discussion.
---------------	--

page_num=<n>	This is a CMR parameter to select the page of results to be processed. The page contains the number of granules selected in the page_size KVP.
Version=1.0.0	WCS Clients Only: Indicates WCS Version 1.0.0 compatibility (optional)
Request=GetCoverage	WCS Clients Only: Identifies the type of WCS request (optional)
token=<token>	Allows the user to provide an Earthdata Login token. This token is used as proof that the user has been authenticated by the Earthdata Login system. It is used for: <ul style="list-style-type: none"> enabling user access to ACL protected granule results in CMR enabling Programmatic Access delivery of results only to authenticated users metrics collection
request_mode=synch async stream (default)	Request mode indicates how the user will receive their results: <ul style="list-style-type: none"> synch = response returned to the client once request has completed processing; response will be XML formatted status that includes result links async = response returned to client immediately; email will be sent upon completion of the request providing result links stream = response will not be returned to the client until request has completed and results are sent directly to the client

EGI Programmatic Access Usage Examples

The following examples address the set of scenarios identified in the NSIDC request for the Programmatic Access capability. Note that Programmatic Access can generally be used to search CMR and access any data sets configured for ESI processing at any of the ECS DAACs that use ESI. These examples show the URL that can be used directly with curl, as well as the command line to invoke the sample script. The following code block shows an example curl command that could be used with an example URL.

```
curl -s -O -J -w "%{http_code}\n%{url_effective}\n%{redirect_url}\n%{filename_effective}\n" \
-nL -b ~/.cookies -c ~/.cookies \
-dump-header response-header.txt \
"http://n5e1l02u.ecs.nsidc.org/ops/egi/request?KVP&KVP&... " \
> HTTP-response-code.txt
```

These examples are only notional and are dependent on available data in particular environments. Parameter details may need to be adjusted to work in other modes.

Scenario 1 SPL3SMP Spatial and Parameter Subsetting and Reformatting to GeoTIFF

SPL3SMP Characteristics

- Name: SMAP L3 Radiometer Global Daily 36 km EASE-Grid Soil Moisture V003
- Format: HDF-5
- Spatial Extent: Daily Global Composite, Bounding Rectangle: (85.0445°, -180°, -85.0445°, 180°)
- Organization: 31 data sets in one group

Service Request Description

- find SPL3SMP granules from March 30 thru April 20, 2015
- select soil moisture parameter
- spatially subset over region of interest
- reformat into GeoTIFF

Request Examples

```
curl -s -O -J -w "%{http_code}\n%{url_effective}\n%{redirect_url}\n%{filename_effective}\n" \
-nL -b ~/.cookies -c ~/.cookies \
-dump-header response-header.txt \
"https://n5eil02u.ecs.nsidc.org/egi/request?short_name=SPL3SMP \
&version=003 \
&time=2015-03-30,2015-04-20 \
&Subset_Data_Layers=/Soil_Moisture_Retrieval_Data/soil_moisture \
&Bbox=100,-20,140,20 \
&format=GeoTIFF"
```

```
pa.sh short_name=spl3smp \
version=003 \
time=2015-03-30,2015-04-20 \
Subset_Data_Layers=/Soil_Moisture_Retrieval_Data/soil_moisture \
Bbox=100,-20,140,20 \
format=GeoTIFF
```

Scenario 2 SPL3SMA Reformatting to GeoTIFF

SPL2SMA Characteristics

- Name: SMAP L3 Radar Global Daily 3 km EASE-Grid Soil Moisture V003
- Format: HDF-5
- Spatial Extent: Single Orbit Swath, Bounding Rectangle: (85.0445°, -180°, -85.0445°, 180°)
- Organization: 80 data sets in four groups

Service Request Description

- find SPL3SMA granules within a temporal window
- select the Radar_Data group
- spatially subset to region of interest
- reformat to GeoTIFF

Request Examples

```
curl -s -O -J -w "%{http_code}\n%{url_effective}\n%{redirect_url}\n%{filename_effective}\n" \
-nL -b ~/.cookies -c ~/.cookies \
-dump-header response-header.txt \
"https://n5eil02u.ecs.nsidc.org/egi/request?short_name=SPL3SMA \
&version=003 \
&time=2015-04-20,2015-04-27 \
&Subset_Data_Layers=/Radar_Data \
&Bbox=60,10,100,30 \
&format=GeoTIFF"
```

```
pa.sh short_name=spl3sma \
version=003 \
time=2015-04-20,2015-04-27 \
Subset_Data_Layers=/Radar_Data \
Bbox=60,10,100,30 \
Format=GeoTIFF
```

Scenario 3 GLAH12 Spatial and Parametric Subsetting

GLAH12 Characteristics

- Name: GLAS/ICESat L2 Global Antarctic and Greenland Ice Sheet Altimetry Data (HDF5) V034
- Format: HDF5
- Spatial Coverage: Global Extent Bounding Rectangle: (90.0°, -180.0°, -90.0°, 180.0°)

- Organization: 173 Datasets in 5 major groups; Each granule contains 14 Orbits

Service Request Description

- find GLAH12 version 034 granules with data date of April 12-13 2007
- select only the 1HZ data group
- spatially subset to a lat-lon box

Request Examples

```
curl -s -O -J -w "%{http_code}\n%{url_effective}\n%{redirect_url}\n%{filename_effective}\n" \
-nL -b ~/.cookies -c ~/.cookies \
-dump-header response-header.txt \
"https://n5e102u.ecs.nsidc.org/egi/request?short_name=GLAH12 \
&version=034 \
&time=2007-04-12T00:00:00,2007-04-14T00:00:00 \
&Coverage=/Data_1HZ \
&bbox=0,-80,100,80"
```

```
pa.sh short_name=glah12 \
version=034 \
time=2007-04-12T00:00:00,2007-04-14T00:00:00 \
Coverage=/Data_1HZ \
Bbox=0,-80,100,80
```

Scenario 4 MOD10A1 reformatting to GeoTIFF and spatial and parameter subsetting

MOD10A1 Characteristics

- Name: MODIS/Terra Snow Cover Daily L3 Global 500m SIN Grid V006
- Format: HDF4
- Spatial Extent: MODIS Sinusoidal Tile Grid

Service Request Description

- Find MOD10A1 version 6 granules from January 1 2011 between 00:00 and 02:00
- Over Eastern Asia (130,30,140,85)
- Select the NDSI_Snow_Cover dataset
- Spatially subset to the desired area (130,30,140,85)
- Convert to GeoTIFF

Request Examples

```
curl -s -O -J -w "%{http_code}\n%{url_effective}\n%{redirect_url}\n%{filename_effective}\n" \
-nL -b ~/.cookies -c ~/.cookies \
-dump-header response-header.txt \
"https://n5e102u.ecs.nsidc.org/egi/request?short_name=MOD10A1 \
&version=006 \
&time=2011-01-01,2011-01-01T02:00:00 \
&bounding_box=130,30,140,85 \
&Subset_Data_Layers=/MOD_Grid_Snow_500m/NDSI_Snow_Cover \
&Bbox=130,30,140,85 \
&format=GeoTIFF"
```

```
pa.sh short_name=MOD10A1 \
version=006 \
time=2011-01-01,2011-01-01T02:00:00 \
bounding_box=130,30,140,85 \
Subset_Data_Layers=/MOD_Grid_Snow_500m/NDSI_Snow_Cover \
Bbox=130,30,140,85 \
format=GeoTIFF
```

Scenario 5 MOD10A1 No Processing, returning raw data with no customization services (subsetting, reformatting or reprojection) applied

MOD10A1 Characteristics

- Name: MODIS/Terra Snow Cover Daily L3 Global 500m SIN Grid V006
- Format: HDF4
- Spatial Extent: MODIS Sinusoidal Tile Grid

Service Request Description

- Find MOD10A1 version 6 granules from December 8, 2002 between 00:00 and 11:00
- Over Guam (144.0, 13.0, 146.0, 14.0)
- Return the data without any customization services (subsetting, reformatting or reprojection) applied

Request Examples

```
curl -s -O -J -w "%{http_code}\n%{url_effective}\n%{redirect_url}\n%{filename_effective}\n" \  
-nL -b ~/.cookies -c ~/.cookies \  
-dump-header response-header.txt \  
"https://n5e1l02u.ecs.nsidc.org/egi/request?short_name=MOD10A1 \  
&version=006 \  
&time=2002-12-08T00:00:00,2002-12-08T11:00:00 \  
&bounding_box=144.0,13.0,146.0,14.0 \  
&AGENT=NO"
```

```
pa.sh short_name=MOD10A1 \  
version=006 \  
time=2002-12-08T00:00:00,2002-12-08T11:00:00 \  
bounding_box=144.0,13.0,146.0,14.0 \  
AGENT=NO
```

Scenario 6 ICESat-2 ATL08 Temporal Subset

ICESat-2 ATL08 Characteristics

- Name: ICESat-2 L3A Land and Vegetation Height V001
- Format: HDF5
- Spatial Extent: Along Ground-Track
- Organization: 6 ground track groups

Service Request Description

- Find ATL08 granules from December 21, 2020 between 10:00 and 12:00
- Select the ground track 1 land_segments group
- Temporally subset to the desired time frame (2020-12-21T10:00:00 to 2020-12-21T12:00:00)

Request Examples

```
curl -s -O -J -w "%{http_code}\n%{url_effective}\n%{redirect_url}\n%{filename_effective}\n" \  
-nL -b ~/.cookies -c ~/.cookies \  
-dump-header response-header.txt \  
"https://n5e1l02u.ecs.nsidc.org/egi/request?short_name=ATL08 \  
&version=001 \  
&time=2020-12-21T10:00:00,2020-12-21T12:00:00 \  
&Subset_Data_Layers=/gt1l/land_segments"
```

```
pa.sh short_name=ATL08 \  
version=001 \  
time=2020-12-21T10:00:00,2020-12-21T12:00:00 \  
Subset_Data_Layers=/gt1l/land_segments
```

Scenario 7 ICESat-2 ATL06 V001 Polygon Subset - KML

ICESat-2 ATL06 Characteristics

- Name: ICESat-2 L3A Land Ice Height version 001
- Format: HDF5
- Spatial Extent: Along Ground-Track
- Organization: 6 ground track groups

Service Request Description

- Find an ATL06 granule
- Select the ground track 1 land_ice_segments group
- Using KML passed to subset to the desired area
- Convert to Shapefile

Request Examples

```
curl -s -O -J -w "%{http_code}\n%{url_effective}\n%{redirect_url}\n%{filename_effective}\n" \  
-nL -b ~/.cookies -c ~/.cookies \  
-dump-header response-header.txt \  
-F "shapefile=@glacier.kml" \  
"https://n5e1l02u.ecs.nsidc.org/egi/request?short_name=ATL06 \  
&version=001 \  
&Subset_Data_Layers=/gt1l/land_ice_segments \  
&Format=Shapefile \  
&time=2015-10-27T04:59:53,2015-10-27T05:00:07"
```

```
pa.sh short_name=ATL06 \  
version=001 \  
Subset_Data_Layers=/gt1l/land_ice_segments \  
time=2015-10-27T04:59:53,2015-10-27T05:00:07 \  
Format=Shapefile
```

Scenario 8 ICESat-2 ATL06 V001 Polygon Subset - Shapefile

ICESat-2 ATL06 Characteristics

- Name: ICESat-2 L3A Land Ice Height version 001
- Format: HDF5
- Spatial Extent: Along Ground-Track
- Organization: 6 ground track groups

Service Request Description

- Find an ATL06 granule, 158875
- Select ground track 1 land_ice_height
- Using Shapefile(.zip) passed to subset to the desired area
- Convert to Shapefile

Request Examples

```
curl -s -O -J -w "%{http_code}\n%{url_effective}\n%{redirect_url}\n%{filename_effective}\n" \  
-nL -b ~/.cookies -c ~/.cookies \  
-dump-header response-header.txt \  
-F "shapefile=@new_crossing.zip" \  
"https://n5e1l02u.ecs.nsidc.org/egi/request?short_name=ATL06 \  
&version=001 \  
&FILE_IDS=158875 \  
&Subset_Data_Layers=/gt1l/land_ice_height \  
&Format=Shapefile"
```

```
pa.sh short_name=ATL06 \  
  version=001 \  
  Subset_Data_Layers=/gt11/land_ice_segments \  
  time=2015-10-27T04:59:53,2015-10-27T05:00:07 \  
  Format=Shapefile
```

Scenario 9 ICESat-2 ATL03 V001 Polygon Subset - GeoJSON

ICESat-2 ATL03 Characteristics

- Name: ICESat-2 L2A Global Geolocated Photon Data version 001
- Format: HDF5
- Spatial Extent: Along Ground-Track
- Organization: 6 ground track groups

Service Request Description

- Find an ATL03 granule
- Select the ground track 1 left group
- Using GeoJSON content passed to subset to the desired area
- Convert to Shapefile

Request Examples

```
curl -s -O -J -w "%{http_code}\n%{url_effective}\n%{redirect_url}\n%{filename_effective}\n" \  
  -nL -b ~/.cookies -c ~/.cookies \  
  -dump-header response-header.txt \  
  "https://n5eil02u.ecs.nsidc.org/egi_DEV06/request? \  
  short_name=ATL03 \  
  &BoundingShape={"type":"FeatureCollection","crs":{"type":"name","properties":{"name":"urn:ogc:def:crs:OGC:1.3:CRS84"}},"features":[{"type":"Feature","properties":{"Name":null,"description":null,"timestamp":null,"begin":null,"end":null,"altitudeMode":"clampToGround","tessellate":-1,"extrude":0,"visibility":-1,"drawOrder":null,"icon":null},"geometry":{"type":"Polygon","coordinates":[[[-147.25640727423652,52.95206360966622],  
  [-145.9181782282926,53.93605555521322],[-143.83211530373293,54.05413458867886],  
  [-142.9268427138297,52.991423287488104],[-142.5332459356109,51.968071664119215],  
  [-144.18635240412988,48.504420015793755],[-144.3044314375955,46.61515548034351],  
  [-143.63531691462353,45.19820707875582],[-141.43117495659826,43.93869738845565],  
  [-139.3844717098605,42.99406512073053],[-139.58127009896987,41.57711671914284],  
  [-143.43851852551416,39.215536049830035],[-147.25640727423652,52.95206360966622]]]}}] \  
  &version=001 \  
  &producer_granule_id=ATL03_20151027T044311_00810302_942_01.h5 \  
  &Subset_Data_Layers=/gt11 \  
  &format=Shapefile \  
  &subagent_id=ICESAT2"
```

```
pa.sh short_name=ATL03 \  
  version=001 \  
  Subset_Data_Layers=/gt11 \  
  BoundingShape={"type":"FeatureCollection","crs":{"type":"name","properties":{"name":"urn:ogc:def:crs:OGC:1.3:CRS84"}},"features":[{"type":"Feature","properties":{"Name":null,"description":null,"timestamp":null,"begin":null,"end":null,"altitudeMode":"clampToGround","tessellate":-1,"extrude":0,"visibility":-1,"drawOrder":null,"icon":null},"geometry":{"type":"Polygon","coordinates":[[[-147.25640727423652,52.95206360966622],  
  [-145.9181782282926,53.93605555521322],[-143.83211530373293,54.05413458867886],  
  [-142.9268427138297,52.991423287488104],[-142.5332459356109,51.968071664119215],  
  [-144.18635240412988,48.504420015793755],[-144.3044314375955,46.61515548034351],  
  [-143.63531691462353,45.19820707875582],[-141.43117495659826,43.93869738845565],  
  [-139.3844717098605,42.99406512073053],[-139.58127009896987,41.57711671914284],  
  [-143.43851852551416,39.215536049830035],[-147.25640727423652,52.95206360966622]]]}}] \  
  Format=Shapefile
```

Scenario 10 SMAP SPL4SMAU V005 Polygon Subset - "CMR-like" coordinate pairs

```
curl -s -O -J -w "%{http_code}\n%{url_effective}\n%{redirect_url}\n%{filename_effective}\n" \  
  -nL -b ~/.cookies -c ~/.cookies \  
  -dump-header response-header.txt \  
  "https://n5eil02u.ecs.nsidc.org/egi_DEV06/request? "
```

```
"https://n5eil02u.ecs.nsidc.org/egi_TSI/request? \
short_name=SPL4SMAU \
&BoundingShape=2.95313,43.37738,-9,43.37738,-8.57813,39.99999,-9,38.02984,-8.71875,36.48187,
-6.1875,36.90405,-4.92188,36.0597,-3.375,37.1855,
-0.28125,37.32622,0.5625,39.71854,2.39063,41.12579,3.51563,42.81448,6.60938,42.95521,9.5625,43.23666,8.29688,54.
21318,3.375,51.82086,0.70313,49.99144,-5.20313,48.30274,-1.40625,45.62898,-2.95313,43.37738 \
&version=005 \
&Subset_Data_Layers=/SPL4SMAU/Forecast_Data/surface_temp_forecast \
&format=GeoTIFF \
&subagent_id=HEG"
https://n5eil12u.ecs.nsidc.org/egi_TSI/request?
short_name=SPL4SMAU&version=004&FILE_IDS=9215717&format=GeoTIFF&Coverage=/Forecast_Data
/surface_temp_forecast&boundingshape=2.95313,43.37738,-9,43.37738,-8.57813,39.99999,-9,38.02984,
-8.71875,36.48187,-6.1875,36.90405,-4.92188,36.0597,-3.375,37.1855,
-0.28125,37.32622,0.5625,39.71854,2.39063,41.12579,3.51563,42.81448,6.60938,42.95521,9.5625,43.23666,8.29688,54.
21318,3.375,51.82086,0.70313,49.99144,-5.20313,48.30274,-1.40625,45.62898,-2.95313,43.37738
&REQUEST_MODE=async&email=afitzger@nsidc.org
```

Retrieving Request Results

After successful completion of a submitted request, the results can be downloaded from the EGI (ESIR) using standard HTTPS requests, either via a web browser or through a command line tool such as `wget` or `curl`. The request response (emails in the case of an asynchronous request) will list the URLs available to download the results. In the case of a multi-file request, this will include a URL for a ZIP file. It is also possible for a DAAC to configure limits on the maximum number of files or the total file volume permitted for a single zip file. In this case, a request that exceeds the configured limits will be split across multiple ZIP files, and URLs will be provided for each ZIP file necessary to download the complete results set.

The following request example submits a request to the EDF DEV01 mode.

```
> wget -O - --server-response --post-data 'FILE_IDS=302942%
2C302943&SUBAGENT_ID=GDAL_CMD&INTERPOLATION=BI&DATASET_ID=SMAP L3 Radar%2FRadiometer Global Daily 9 km EASE-
Grid Soil Moisture V003&FORMAT=GIF&EMAIL=bob@example.com&CLIENT=ESI&REQUEST_MODE=async' https://f5eil01.edn.ecs.
nasa.gov/egi_DEV01/request

> curl -k -d 'FILE_IDS=302942%2C302943&SUBAGENT_ID=GDAL_CMD&INTERPOLATION=BI&DATASET_ID=SMAP L3 Radar%
2FRadiometer Global Daily 9 km EASE-Grid Soil Moisture V003&FORMAT=GIF&EMAIL=peterlsmith@hotmail.
com&CLIENT=ESI&REQUEST_MODE=async' https://f5eil01.edn.ecs.nasa.gov/egi_DEV01/request
```

(don't forget to set the email query parameter). This should take just a minute to complete, and will send an email upon completion of similar to this:

Status update for ECS data processing request 23002

Your request is currently complete. Your request has completed processing. You may retrieve the results from the download URLs until 2018-12-17 13:28:33.289

Note from Client: To view the status of your request, please see: <http://search.sit.earthdata.nasa.gov/data/retrieve/0435908906>

The output of this request can be downloaded from the following URLs:

- https://f5eil01.edn.ecs.nasa.gov/esir_DEV06/23002.html (Listing of individual files)
- https://f5eil01.edn.ecs.nasa.gov/esir_DEV06/23002.zip?1 (ZIP file containing output files)
- https://f5eil01.edn.ecs.nasa.gov/esir_DEV06/23002.zip?2 (ZIP file containing output files)
- https://f5eil01.edn.ecs.nasa.gov/esir_DEV06/23002.zip?3 (ZIP file containing output files)
- https://f5eil01.edn.ecs.nasa.gov/esir_DEV06/23002.zip?4 (ZIP file containing output files)
- https://f5eil01.edn.ecs.nasa.gov/esir_DEV06/23002.zip?5 (ZIP file containing output files)
- https://f5eil01.edn.ecs.nasa.gov/esir_DEV06/23002.zip?6 (ZIP file containing output files)
- https://f5eil01.edn.ecs.nasa.gov/esir_DEV06/23002.zip?7 (ZIP file containing output files)
- https://f5eil01.edn.ecs.nasa.gov/esir_DEV06/23002.zip?8 (ZIP file containing output files)
- https://f5eil01.edn.ecs.nasa.gov/esir_DEV06/23002.zip?9 (ZIP file containing output files)
- https://f5eil01.edn.ecs.nasa.gov/esir_DEV06/23002.zip?10 (ZIP file containing output files)
- https://f5eil01.edn.ecs.nasa.gov/esir_DEV06/23002.zip?11 (ZIP file containing output files)
- https://f5eil01.edn.ecs.nasa.gov/esir_DEV06/23002.zip?12 (ZIP file containing output files)
- https://f5eil01.edn.ecs.nasa.gov/esir_DEV06/23002.zip?13 (ZIP file containing output files)
- https://f5eil01.edn.ecs.nasa.gov/esir_DEV06/23002.zip?14 (ZIP file containing output files)

Due to the size and/or the number of files in your request, the output has been split across multiple ZIP files. You will need to download all ZIP files to retrieve all request output files.

Please contact John Doe at cmshared@f5eil01.edn.ecs.nasa.gov with any questions about this request. Be sure to reference the request ID 23002 in any correspondence.

In this case, DEV01 has been configured for no more than 10 files in a ZIP, so multiple ZIP files are required in order to download the entire result. You can use wget or curl (or a web browser) to download these zip files as follows:

```
> wget --no-check-certificate --load-cookies ~/.cookies --save-cookies ~/.cookies --keep-session-cookies
'https://f5eil01.edn.ecs.nasa.gov/esir_DEV01/143846.zip?1'

> curl -o 143846-1.zip -nkL -b ~/.cookies -c ~/.cookies 'https://f5eil01.edn.ecs.nasa.gov/esir_DEV01/143846.zip?
1'
```

Note that in order to retrieve the data, you must configure wget or curl for URS authentication. When using a web browser, the browser will prompt the user for their credentials if it has not already done so.

Error Responses

Two types of errors can be encountered: errors encountered by the operating system when executing the curl program, and errors encountered by the curl program when communicating using the HTTP protocol. To detect either kind of error, the return codes should always be checked.

1. curl execution errors

This type of error is returned to the script from the curl program through the shell exit code. A complete list of curl error codes can be found here: <https://curl.haxx.se/libcurl/c/libcurl-errors.html>. The following are most likely causes.

- * curl return code 7 timeout - could not establish connection to remote http server
- * curl return code 23, unable to write output files, file already exists

2. HTTP protocol return codes

- HTTP Return Code 200 "OK", indicates success
- HTTP Return Code 201 "Resource Created", indicates xml error response file was provided. This happens when the service request successfully executed, but no output was generated. See details in the response file. Common causes are:
 - no data found in the requested spatial or temporal subset
 - requested data layer was not found
- HTTP Return Code 303 "See Other", this indicates a redirect to another url.
- HTTP Return Code 302 "Moved Temporarily" This response is only generated when the EGI is configured behind Earthdata Login. Your client must be configured to handle the required user authentication. See [Earthdata Login Integration](#).
- HTTP Return Code 400 "Bad Request".
- HTTP Return Code 404 "Not Found".
 - a CMR-only query
 - a request that matched no granules in the CMR query
 - Collection not configured for that operation
 - Request exceeds configured limits
 - Parameter not recognized by CMR
 - Requested processing produced no output data

EGI Programmatic Access Configuration and Hardening

The Programmatic Access capability is configured by the DAAC staff specifically for each execution environment (mode).

The Programmatic Access capability also includes features for hardening the interface to mitigate the risk of user activities affecting DAAC operations.

EGI Programmatic Access System Configuration

Each DAAC operational mode is configured as a distinct data provider to a particular instance of CMR.

The programmatic Access components are configured so that the query communicates with the correct instance of CMR (identified by the endpoint) and searches the correct provider metadata (identified by the provider= parameter in the url).

This information is seen in the 303 redirect url when a CMR-only query is performed. For example, this CMR-only query redirect url is indicated in the http response header, showing that the CMR endpoint is the sit instance of CMR, and the search is for metadata from provider DEV07.

```
REDIR='https://cmr.sit.earthdata.nasa.gov/search/granules?provider=DEV07&short_name=glah12&version=34'
```

The EGI endpoint to be used for the Programmatic Access request is also configured for the operational mode. This is the endpoint that must be used in the HTTP request url. For example:

```
EDF DEV02 endpoint: http://f5eil01v.edn.ecs.nasa.gov/dev02/egi/request?
```

Hardening Considerations

The EGI components that implement Programmatic Access have been hardened to protect the DAAC systems from excessive load.

Here are items that the Programmatic Access user should be aware of.

1. Transaction Size Limit

A single request can result in a large number of granules returned from CMR. A configuration parameter allows DAAC operators to set a limit on how many granules can be processed in a single request. An end user who needs to process more than the limit can submit multiple requests, using the CMR `page_size` and `page_num` parameters.

2. Cancel Request

If a Programmatic Access request for numerous granules is consuming too many resources (such as CPU or memory), the DAAC operations staff can cancel the request. This takes effect after the current granule finishes processing.

References:

CMR search API

https://cmr.earthdata.nasa.gov/search/site/search_api_docs.html

curl Documentation

<https://curl.haxx.se/>