How To Access Data With PyDAP

The PyDAP developers implemented a module for authenticating with Earthdata Login (it also works for any basic type authentication), but depending upon how you download PyDAP, the module may or may not be included. If the module is included, then to download files from an Earthdata Login enabled server, you can use code like the following:

```
from pydap.util.urs import install_basic_client

install_basic_client()

from pydap.client import open_url
dataset = open_url('https://goldsmr4.gesdisc.eosdis.nasa.gov/opendap/MERRA2/M2T1NXSLV.5.12.4/2016/06
/MERRA2_400.tavg1_2d_slv_Nx.20160601.nc4')
```

If the module is not included, you can manually add it to your code as follows:

```
# BEGIN BASIC AUTH MODULE CODE (Comments removed)

import cookielib

import netrc

import urllib2

import re


import pydap.lib

from pydap.exceptions import ClientError


import logging


log = logging.getLogger(__name__)

log.setLevel(logging.DEBUG)


# Set the debug level for urllib2.

debuglevel=1


def install_basic_client(uri='', user='', passwd='', use_netrc=True):

    # Create special opener with support for Cookies

    cj = cookielib.CookieJar()


    # Create the password manager and load with the credentials using

    pwMgr = urllib2.HTTPPasswordMgrWithDefaultRealm()


    # Get passwords from the .netrc file nless use_netrc is False

    if use_netrc:

        logins = netrc.netrc()

        accounts = logins.hosts # a dist of hosts and tuples

        for host, info in accounts.iteritems():

            login, account, password = info

            log.debug('Host: %s; login: %s; account: %s; password: %s' % (host, login, account, password))

            pwMgr.add_password(None, host, login, password)


    if uri and user and passwd:

        pwMgr.add_password(None, uri, user, passwd)
```

```
    opener = urllib2.build_opener(urllib2.HTTPBasicAuthHandler(pwMgr),
                                  urllib2.HTTPCookieProcessor(cj))

    opener.addheaders = [('User-agent', 'pydap/EL')]


    urllib2.install_opener(opener)


    def new_request(url):
        if url[-1] is '&': url = url[0:-1]
        log.debug('Opening %s (install_basic_client)' % url)
        r = urllib2.urlopen(url)

        resp = r.headers.dict
        resp['status'] = str(r.code)
        data = r.read()

        # When an error is returned, we parse the error message from the
        # server and return it in a ``ClientError`` exception.
        if resp.get("content-description") == "dods_error":
            m = re.search('code = (?P<code>\d+);\s*message = "(?P<msg>.*)"',
                    data, re.DOTALL | re.MULTILINE)
            msg = 'Server error %(code)s: "%(msg)s"' % m.groupdict()
            raise ClientError(msg)

        return resp, data

    from pydap.util import http
    http.request = new_request

# END BASIC AUTH MODULE CODE


install_basic_client()

from pydap.client import open_url
dataset = open_url('https://goldsmr4.gesdisc.eosdis.nasa.gov/opendap/MERRA2/M2T1NXSLV.5.12.4/2016/06
/MERRA2_400.tavg1_2d_slv_Nx.20160601.nc4')
```

Note that in both cases, user credentials are pulled from the .netrc file, so that must be set up in order for it to work.