Need data? The IRIS DMC can help

Chad Trabant, Tim Ahern, Rob Casey, Celso Reyes, Yazan Suleiman, Bruce Weertman, Rick Benson

IRIS Data Management Center, Seattle Washington

14-767

Summary

The IRIS Data Management Center (DMC) is the access point for a large, global archive of seismological data, including all past and future recordings from USArray. The DMC offers multiple mechanisms for accessing the archive allowing users to choose an the best fit for their needs. One of the newer mechanisms is a suite of web service interfaces with the following characteristics and features:

- Designed for use by scientific data users who can create their own programs/scripts to easily integrate data access into their processing
- Many useful data access software packages have been built on top of these interfaces to support command line access, access from Java, Python and MATLAB and more
- Access to all primary data and information repositories at the DMC with new interfaces to data quality metrics and other repositories soon.
- As programmatic interfaces the services are well suited for integration of data into arbitrary display and processing work flows
- Access to data holdings in a form usable by non-seismologists

Command Line access (service.iris.edu/clients)

Fetch scripts allow access to the DMC archive and are a comfortable fit for command line users. These scripts are written in Perl and are well suited for automation and integration into existing workflows on most operating systems. For metadata and event information, the Fetch scripts even parse the returned data into simple text summaries.

To demonstrate their use, below is an example of requesting data using the Fetch scripts for the 2002 Denali fault earthquake.

First find the event parameters using FetchEvent:

```
FetchEvent -s 2002-11-03 -e 2002-11-04 --radius 65:-155:10 --mag 7.0 --allmags
```

... resulting in output similar to:

```
Received 1.0 KB of event information in 0.9 seconds (1.1 KB/s)
Processed event information for 1 events, 1 origins in 1.0 seconds (1.0 KB/s)
1403125 |2002/11/03 22:12:41.1300| 63.6285|-147.6114| 10.8|ISC|ISC|ISC,6123395|mb,7.0,NEIC...MW,7.9,HRVD...|CENTRAL ALASKA
```

The use FetchData to retrieve GSN broadband seismic channels in miniSEED format:

```
FetchData --net _GSN --chan BH? -s 2002-11-03T22:12:41 -e 2002-11-03T23:12:41 -o denali.mseed -m denali.metadata
```

If you would rather have SAC files, the DMC's mseed2sac converter can be used:

```
mseed2sac -E "2002,305,22:12:41.13/63.6285/-147.6114/10.8/Central Alaska" -m denali.metadata denali.mseed
```

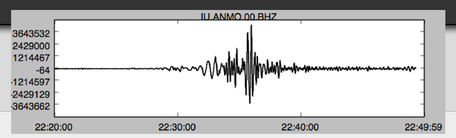
Optionally include event details into the SAC header

Python access: ObsPy (www.obspy.org)

Access DMC data directly from your Python scripts with ObsPy. Developed at LMU Munich, ObsPy is an open source project that can be used to process seismological data.

This example that retrieves a waveform from the DMC, and then plots it.

```
from obspy.fdsn import Client
from obspy import UTCDateTime
client = Client("IRIS")
t = UTCDateTime("2002-11-03T22:12:41.000")
st = client.get_waveforms("IU", "ANMO", "00", "BHZ", t, t + 3600)
st.plot()
```



MATLAB access: irisFetch (service.iris.edu/clients)

Routines within irisFetch.m allow seamless access to the DMC archives from within MATLAB scripts. The irisFetch tools are capable of retrieving time series data, station metadata, and event parameters as MATLAB structs, available for immediate use.

Channel metadata

Station metadata can be retrieved via a single line of code. Use additional name-value pairs to further refine the results. For example:

```
aChannel = irisFetch.Stations('response','IU','ANMO','00','BHZ') % includes response
someStas = irisFetch.Stations('station','_GSN','','','BHZ','starttime','1/1/2010','maximumlatitude',40)
```

Information is returned in a nested form, shown at right, but can be simplified by using flattenToChannel() or flattenToStation() .

```
sf = irisFetch.flattenToChannel( aChannel )
```

```
Network
└ Station
  └ Station Epoch
    └ Channel
      └ Channel Epoch
        Response
```

Retrieve seismic traces

Seismic traces can be retrieved with a line of code specifying what & when:

```
tr = irisFetch.Traces('IU','ANMO','00','BH?','2010-02-27 06:45:00','2010-02-27 07:45:00')
```

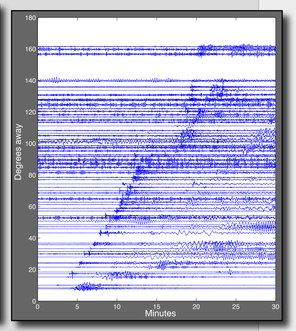
Traces are returned as an array of structs ready for further manipulation, an example struct is shown at right. Below, how to access groups of data.

```
allChannels = {tr.channel}
allElevations = [tr.elevation]
```

```
% Structure of the first trace, tr(1)
          network: 'IU'
          station: 'ANMO'
         location: '00'
          channel: 'BH1'
          quality: 'M'
         latitude: 34.9460
        longitude: -106.4571
        elevation: 1671
            depth: 145
          azimuth: 328
              dip: 0
      sensitivity: 3.4566e+09
sensitivityFrequency: 0.0200
       instrument: 'Geotech KS...
sensitivityUnits: 'M/S'
             data: [72000x1 single
      sampleCount: 72000
       sampleRate: 20
        startTime: 7.3420e+05
          endTime: 7.3420e+05
            sacpz: [1x1 struct]
```

Example: visualizing travel times

```
ev = irisFetch.Events('StartTime', '1/1/2010', 'MinimumMagnitude', 6.0)

[arclen, az] = distance([tr.latitude],[tr.longitude],...
                        [ev.PrimaryLatitude],[ev.PrimaryLongitude]);

% Although many events exist, use only one for each 2 degrees of distance
[~,i,~]      = unique(fix(arclen / 2),'first');

for idx = i
   tStart  = datenum(ev(idx).PrimaryTime);
   tEnd    = addtodate(startTime, 30, 'minute');
   tr          = irisFetch.Traces('IU','ANMO','00','BHZ', tStart, tEnd);
   data        = double(tr.data - mean(tr.data)); % demean
   scaledData  = data / max( data / 2);
   timeSteps   = [0:(tr.sampleCount-1)] / tr.sampleRate;
   plot(timeSteps / 60, scaledData + arclen(idx));
   hold('on');
end
xlabel('Minutes');
ylabel('Degrees away');
```



* *The irisFetch tools ares built upon the DMC's Java library (below); however, Java knowledge is not required*

Java access: IRIS Java WS Library (service.iris.edu/clients)

The IRIS Java WS Library provides an API to allow direct access to DMC data, including: time series data, station metadata and event parameters.

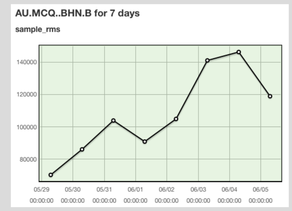
Example: retrieve waveforms from IRIS and display basic information from each:

```
import edu.iris.dmc.ws.extensions.entities.Trace;
import edu.iris.dmc.ws.extensions.fetch.TraceData;

// Retrieve traces associated with an earthquake from BHZ channels of the IU network
traces = TraceData.fetchTraces("IU","*","00","BHZ","2010-02-
27T06:30:00","2010-02-27T10:30:00",'B',false);

// Display some basic information about the trace
for (Trace tr: traces) {
   System.out.printf("Found %2s-%5s (%2s)  located at %8.4f lat, %8.4f lon\n",
             tr.getNetwork(), trace.getStation(), tr.getChannel(), tr.getLatitude(),
tr.getLongitude());
   System.out.printf("  This trace has %d samples, at %7.2f samples per second\n",
             tr.getSampleCount(), tr.getSampleRate());
 }
```

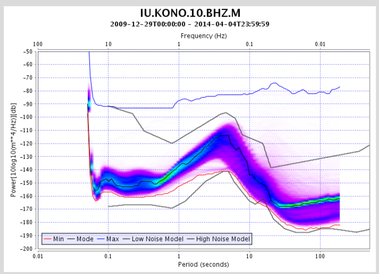
Data quality as a service

IRIS DMC has been building a database and web service suite, called MUSTANG, to provide data quality metrics of IRIS DMC's archival data, amounting to hundreds of gigabytes of detailed measurements. Access is provided to users via web services, with documentation and help information on web pages in the same style as the the DMC's other web services. Providing close to 50 separate measurements, MUSTANG provides network operators and seismologists actionable measurements that they can use to monitor the health of their stations, as well as filter their data sets for those seismograms that most fits their profile of interest.

The DMC will be developing methods to use the various MUSTANG metrics to filter data requests before data are returned to the end users, thus providing a valuable data culling capability to ease a researchers work to assemble a Research-Ready-Data-Set. Development of this capability will begin in FY16.



A plot of RMS by the LASSO tool using MUSTANG.



A PSD/PDF returned by a MUSTANG web service.

Though most of the data returns are raw numbers in tabular or XML format, we are providing a service that will allow users to generate aggregate PSD and PDF plots. In addition, we have visualization services for the web (in development) to allow for combining, manipulating, and drilling into metrics to derive meaning and insight into the state of health of the data we have archived, as well as the stations that recorded it.

MUSTANG is going to be a central component of improved automation for how the IRIS DMC monitors the quality of station data and makes characterizations of data problems to be resolved through investigation and contact with network operators.

The MUSTANG data set is still being constructed during 2014 as there is a lot of data to cover. A beta release of web service interfaces to MUSTANG data is available for evaluation at service.iris.edu/mustangbeta.

International standards for seismology

Over the last couple of years the IRIS DMC has worked with our partners in the International Federation of Digital Seismograph Networks (FDSN) to create two important new standards:

For a common data access:
FDSN Web Services - www.fdsn.org/webservices/

For a common, modern metadata exchange format:
FDSN StationXML - www.fdsn.org/xml/station/

These developments are an important evolutionary step and are already having a significant impact on how seismologists access and process data.

http://xkcd.com/949/