irisFetch - Feature #580
Add support for using the IRIS Federator

08/07/2014 03:12 PM - Chad Trabant

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
<th>Start date:</th>
<th>08/07/2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
<td>Due date:</td>
<td>09/11/2015</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Robert Weekly</td>
<td>% Done:</td>
<td>100%</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
<td>Estimated time:</td>
<td>20.00 hours</td>
</tr>
<tr>
<td>Target version:</td>
<td>2.0.7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description
Soon a beta version of the IRIS Federator will be released and support should be integrated into irisFetch.m

History

#1 - 08/15/2015 08:23 PM - Chad Trabant

The use of the Federator should follow this general pattern:

- User requests waveforms normally with Traces, invoking use of Federation by a keyword 'Federate' (akin to 'includePZ')
- irisFetch sends requested parameters to the irisws-fedcatalog service (http://service.iris.edu/irisws/fedcatalog/1/)
- irisFetch parses response from irisws-fedcatalog and fetches each identified channel from each identified data center as would normally be done with a Traces request.

The data center from which waveforms are fetched must be printed to the console (a name is identified in the response from the irisws-fedcatalog service), messages that cannot easily be turned off. Along the lines of:

"Fetching data from IRISDMC (http://ds.iris.edu)"
"Fetching data from ORFEUS (http://www.orfeus-eu.org)"

Somehow the data center from which a waveform was fetched must be identified in the trace structure, ideally hard to ignore for the user. I see two possibilities for this:

1. Return a structure where the first level key is the data center label returned by irisws-fedcatalog, e.g. traces("IRISDMC") contains all the traces returned from the IRIS DMC. This definitely requires the user to see the data centers from which they got data, but it changes the normal response from the Traces function.

2. Return a list of Traces including a named element of the data center from which they were fetched. This, of course, is easy to ignore.

Other suggestions on how to identify the data centers is welcome.

Note that the DMC's FetchData command line script does Federation (the -F option), it would be preferable to mimic the behavior (e.g. console output) whenever possible. Latest version: https://seiscode.iris.washington.edu/projects/ws-fetch-scripts/files
I have added simple support for the IRS federator.
To use it, use irisFetch.Traces and add a 'federated' option.
example usage:

```python
tr = irisFetch.Traces(['R','A','*','BHZ','2010-02-27 6:30:00', '2010-02-27 6:31:00','federated'])
```

**usage**

This function first queries the federator service to determine the holdings from each datacenter that match the request. Then, irisFetch retrieves the traces from the datacenter. See additional FEDERATED note below.

**Additional help info**

```plaintext
% ABOUT FEDERATED DATA
% When traces are received via the "federated" catalog, they
% are grouped by datacenter. The result will be a structure,
% containing fields with the name of the data center.
% Using a concrete example, if I request:
%    tr = irisFetch.Traces('R','A','*','BHZ','2010-02-27 6:30:00', '2010-02-27 6:31:00','federated')
% irisFetch first queries the federator service located at:
%  http://service.iris.edu/irisws/fedcatalog/1/
% which returns matches at three datacenters:
%  3 matches at BGR (http://eida.bgr.de)
%  9 matches at IRISDMC (http://ds.iris.edu)
%  4 matches at RESIF (http://www.resif.fr)
% it then retrieves each trace, one after the other. Placing
% them into a final structure which contains the data that was
% successfully retrieved:
%    tr =
%        BGR: [1x3 struct]
%        IRISDMC: [1x7 struct]
%        RESIF: [1x2 struct]
% * Side effect of retrieving data from other datacenters:
% The java library used by irisfetch remembers the last
% datacenter. So if you then try to retrieve data without the
% 'federated' method, it might search in the wrong center. To
% fix, either clear java or send a federated request that finds
% data at your specific datacenter.
```
Another question:
Should I include a way for users to send the results of an arbitrary federator request into Trace? If so, any thoughts?

What about Station? Events?

Hi Robert,
I think it's good to go. But there'll always be more I could add to it. Here's the irisFetch.m file for you to tear apart (hopefully ASAP since contract is nearing end).

Thanks!
Celso

Federation capability confirmed.

After fetching some traces I tried writing them out with the new Traces2SACfile() capability and ran into this:

```matlab
>> t = irisFetch.Traces('IU','ANMO','10','BHZ','2010-02-27 06:30:00','2010-02-27 10:30:00');
>> irisFetch.Traces2SACfile(t,'.')
Undefined function or variable 'dbPrint'.
```

And:

```matlab
Error in irisFetch/write_traces_to_sac/write_trace (line 2499)
   dbPrint('Opening : [%s]\n', outputFileName)
Error in irisFetch/write_traces_to_sac (line 2272)
   write_trace(writeDirectory, traces(n))
Error in irisFetch.Traces2SACfile (line 108)
   irisFetch.write_traces_to_sac(writeDirectory, traces, verbosity);
```
#6 - 09/14/2015 06:10 PM - Chad Trabant
It would appear that irisFetch.dbPrint() is a new function in this version. But it's not declared?

#7 - 10/19/2015 01:13 PM - Robert Weekly
- Target version set to 2.0.7

#8 - 10/29/2015 02:05 PM - Chad Trabant
- Status changed from Resolved to Closed

Confirmed function.

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>irisFetch.m</td>
<td>137 KB</td>
<td>09/05/2015</td>
<td>Celso Reyes</td>
</tr>
</tbody>
</table>