NIED Hi-net data request and process scripts

This is a collection of scripts for Hi-net data request, download and process. It does not come with any warranties, nor is it guaranteed to work on your computer. The user assumes full responsibility for the use of all scripts. The author are **NOT** responsible for any damage that may follow from correct *or* incorrect use of these scripts.

Dependency

- Python 3.4 (Not work under 2.6, 2.7; Not Tested under 3.3)
- Python third-party modules
  - docopt
  - BeautifulSoup4
  - requests
  - clint
- catwin32 from Hi-net win32tools

How to Use

Modify Configure File

`Hinet.cfg` is the configure file you need to modify.

- `User` : User name.
- `Password` : Password.
- `Net` : Default net code to request data from.
- `Maxspan` : Maximum record length for one request.
- `catwin32` : Path to `catwin32` supplied by Hi-net win32tools.

Request and Download Data

`HinetContRequest.py` is used to request and download data from Hi-net server.

Usage

```
$ python HinetContRequest.py -h
Request continuous waveform data from NIED Hi-net.

Usage:
```

```
    HinetContRequest.py <year> <month> <day> <hour> <min> <span> [options]
    HinetContRequest.py -h

Options:
    -h, --help              Show this help.
    -c CODE --code=CODE     Select code for organization and network.
    -d DIR --directory=DIR  Output directory. Default: current directory.
    -o FILE --output=FILE   Output filename.
                            Default: CODE_YYYYMMDDHHMM_SPAN.cnt
    -t FILE --ctable=FILE   Channel table filename. Default: CODE_YYYYMMDD.ch
```

Examples

1. Request data of Hi-net start from 2010-10-01T15:00:00 (JST) with duration of 20 minutes

   ```
   python HinetContRequest.py 2010 10 01 15 00 20
   ```

2. Request data of F-net start from 2010-10-01T15:00:00 (JST) with duration of 20 minutes

   ```
   python HinetContRequest.py 2010 10 01 15 00 20 -c 0103
   ```

3. Request data of Hi-net, with customized output directory and filename

   ```
   python HinetContRequest.py 2010 10 01 15 00 20 -d aaa -o aaa.cnt -t aaa.ch
   ```

4. Request data of Hi-net, use default filename and customized output directory. (**Highly Recommended**)

   ```
   python HinetContRequest.py 2010 10 01 15 00 20 -d 201010010600
   ```

If you run `HinetContRequest.py` in the highly recommender way, you will get a directory `201010010600` with two file inside: `0101_201010011500_20.cnt` and `0101_20101001.ch`.

```
|-- 201010010600
    |-- 0101_201010011500_20.cnt
    `-- 0101_20101001.ch
```

Extract SAC files from WIN32 file

`rdhinet.py` is what you need.

Usage

```
Extract SAC data files from NIED Hi-net WIN32 files

Usage:
    rdhinet.py DIRNAME [-C <comps>] [-D <outdir>] [-S <suffix>] [-P <procs>]
    rdhinet.py -h

Options:
    -h          Show this help.
    -C <comps>  Components to extract, delimited using commas.
                Avaiable components are U, N, E, X, Y et. al.
                Default to extract all components.
    -D <outdir> Output directory for SAC files.
    -S <suffix> Suffix of output SAC files. Default: no suffix.
    -P <procs>  Parallel using multiple processes.
                Set number of CPUs to <procs> if <procs> equals 0. [default: 0]
```

Examples

1. Extract all channels

   ```
   python rdhinet.py 201010010600
   ```

2. Extract NEU components with suffix 'SAC'

   ```
   python rdhinet.py 201010010600 -C U,N,E -S SAC
   ```

In most cases, what you need is only `-C` option.

If you run `python rdhinet.py 201010010600 -C U`, you will get SAC files looks like `N.FRNH.U` under directory `201010010600`.

Extract SAC PZ files from Channel Table

`ch2pz.py` is the one.

Usage

```
$ python ch2pz.py -h
Convert NIED Hi-net Channel Table file to SAC PZ files

Usage:
```

```
    ch2pz.py DIRNAME [-C <comps>] [-D <outdir>] [-S <suffix>]

Options:
    -C <comps>    Channel Components to convert. Choose from U,N,E,X,Y et. al.
                  Default to convert all components.
    -D <outdir>   Output directory of SAC PZ files. Use the directory of
                  Channel Table file as default.
    -S <suffix>   Suffix for SAC PZ files. [default: SAC_PZ]
```

Examples

1. Extract all channels

   ```
   python ch2pz.py 201010010600
   ```

2. Extract NEU components

   ```
   python ch2pz.py 201010010600 -C U,N,E
   ```

In most cases, what you need is only `-C` option.

If you run `python ch2pz.py 201010010600 -C U`, you will get SAC PoleZero files looks like `N.FRNH.U.SAC_PZ` under directory `201010010600`.