

Detailed README for chado2GEO reporting code (the/my code hereafter)

Section 0 the code organization and requirement

the code is living at modencode svn. Check out the code by

```
svn co svn+ssh://youraccount@svn.modencode.org/modencode/private/reporter2 reporter
```

the code works with EO's validator code (the validator hereafter) version 620. I changed function `get_denormalized_protocol_slots` and added a function `get_full_denormalized_protocol_slots` in file `ModENCODE/Parser/Chado.pm` at version 597. these two functions are useful to read normalized applied protocol in chado and generate denormalized applied protocol (see EO's documents for detail on normalize/denormalize and protocol/applied protocol). I also mute a complain from `confess "What, no accession?"` unless `$_[1]->{'accession'}`; in `DBXref.pm`.

To check out the validator version that works with my code, you need:

```
svn co svn+ssh://youraccount@svn.modencode.org/modencode/private/validator_zz validator
```

the dependent cpan modules are:

`LWP::UserAgent`

`HTTP::Request::Common`

`HTTP::Response`

`HTTP::Cookies`

[File::Basename](#)

[File::Copy](#)

[File::Spec](#)

`Net::FTP`

`Mail::Mailer`

`Class::Std`

and all modules needed for validator. There might be overlapping requirements for modules.

Section 1 interface to pipeline

The pipeline needs to call `chado2GEO.pl` for the reporting process to start. This script lives in the `reporter` directory.

`Chado2GEO.pl` could accept 3 required and 2 optional command-line parameters. The three required parameters are `unique_id`, `out`, `config`. The two optional parameters are `make_tarball` and `send_to_geo`, both default is 0.

`unique_id` is the modencode unique id of the submission. this is an integer number.

`out` is the output directory where the generated GEO files will go.

`config` is a general configure file that contains several information that this code will need, such as:

1. the modencode chado databases,
2. the modencode geo submission account,
3. the ftp of geo, where the geo submission goes
4. email to geo, this is for notification to geo curators that a submission arrives.

5. wiki. the modencode private wiki, it is used for fetching protocol full text in case the brief description of DBfield of a protocol is blank.
6. prefix and suffix of flattened (unstructured) tarball download url.

After the code fetches necessary meta-data information, it will try to archive raw and normalization data to make a tarball to send to GEO. `make_tarball` and `send_to_geo` are two options to facilitate the data wrangler to view the geo submission and avoid unfortunate mistakes.

`make_tarball`, default 0, which means DO NOT archive raw/normalized data. If set to 1, the code will start to fetch raw and normalization data.

`send_to_geo`, default 0, which means DO NOT send crappy test result to GEO. If set to 1, the code will start to ftp the tarball to GEO ftp site and send a notification email to geo curators telling them a new submission arrives in ftp.

I also add an usage function. so you can simply type
perl chado2GEO.pl
to see the options and their meanings.