Adding Extractors

<u>Exercise 1 (/home/bd/bd-tutorial/AddingExtractors/):</u>

1. Start BrownDog base:

```
cd /home/bd/bd-base
./bd
```

2. In a second terminal set up a basic development environment for conversions:

```
cd /home/bd/bd-base
./extractor
git clone
https://opensource.ncsa.illinois.edu/bitbucket/scm/bd/bd-tutorial.git
cd /home/clowder/bd-tutorial/AddingExtractors
```

3. Write a Clowder script:

```
nano extactor.py
```

2. Write extractor description:

```
nano extractor_info.json
```

3. Start your new extractor:

```
./extractor.py
```

4. Test it with the Brown Dog Command Line Interface:

```
bd –b http://fence:8080 –o /home/bd/bd-tutorial/AddingExtractors/gchn.csv
```

Appendix

wordcount.py:

```
#!/usr/bin/env python

"""Example Clowder script."""

import logging
import subprocess

from pyclowder.extractors import Extractor
import pyclowder.files

class WordCount(Extractor):
    """Count the number of characters, words and lines in a text file."""
    def __init__(self):
        Extractor.__init__(self)

        # add any additional arguments to parser
        # self.parser.add_argument('--max', '-m', type=int, nargs='?', default=-1,
        #                          help='maximum number (default=-1)')

        # parse command line and load default logging configuration
        self.setup()

        # setup logging for the exctractor
        logging.getLogger('pyclowder').setLevel(logging.DEBUG)
        logging.getLogger('__main__').setLevel(logging.DEBUG)

    def process_message(self, connector, host, secret_key, resource, parameters):
        # Process the file and upload the results

        logger = logging.getLogger(__name__)
        inputfile = resource["local_paths"][0]
        file_id = resource['id']

        # call actual program
        result = subprocess.check_output(['wc', inputfile], stderr=subprocess.STDOUT)
        (lines, words, characters, _) = result.split()

        # store results as metadata
        result = {
            'lines': lines,
            'words': words,
            'characters': characters
        }

        metadata = self.get_metadata(result, 'file', file_id, host)
        logger.debug(metadata)

        # upload metadata
        pyclowder.files.upload_metadata(connector, host, secret_key, file_id,
metadata)

if __name__ == "__main__":
    extractor = WordCount()
    extractor.start()
```

extractor_info.json:

```
{
  "@context": "http://clowder.ncsa.illinois.edu/contexts/extractors.jsonld",
  "name": "ncsa.wordcount",
  "version": "2.0",
  "description": "WordCount extractor. Counts characters, words, and lines in text file.",
  "author": "Rob Kooper <kooper@illinois.edu>",
  "contributors": [],
  "contexts": [
    {
      "lines": "http://clowder.ncsa.illinois.edu/metadata/ncsa.wordcount#lines",
      "words": "http://clowder.ncsa.illinois.edu/metadata/ncsa.wordcount#words",
      "characters": "http://clowder.ncsa.illinois.edu/metadata/ncsa.wordcount#characters"
    }
  ],
  "repository": {
    "repType": "git",
    "repUrl": "https://opensource.ncsa.illinois.edu/stash/scm/cats/pyclowder.git"
  },
  "process": {
    "file": [
      "text/*",
      "application/json"
    ]
  },
  "external_services": [],
  "dependencies": [],
  "bibtex": []
}
```