

Adding Logging functionality to ConfigMaster

We want to add something like this to ConfigMaster:

```
import os
import datetime as dt
import argparse
import shutil
import logging
import sys

def main():
    parser = argparse.ArgumentParser(description='This program does ...', formatter_class=argparse.
ArgumentDefaultsHelpFormatter)
    parser.add_argument('-l', '--logPath', help="The full path to the log file. Use '-' to log to stdout.",
default="-")
    parser.add_argument('-d', '--debugLevel', choices=["VERBOSE", "DEBUG", "INFO", "WARNING", "ERROR", "CRITICAL"],
help="Control volume of log messages.", default="INFO")
    args = parser.parse_args()

#####

    # setup
    logging

    lPath = args.logPath

    # does this do anything, since I have a default value for --logPath?
    if lPath == None:
        lPath = os.path.join(sys_path.Log_dir, os.path.basename(__file__) + "." + datetime.datetime.today().
strftime('%Y%m%d') + ".log")

    # Add custom verbose logging
    level

    logging.VERBOSE = 5
    logging.addLevelName(logging.VERBOSE, "VERBOSE")
    logging.Logger.verbose = lambda inst, msg, *args, **kwargs: inst.log(logging.VERBOSE, msg, *args, **kwargs)
    logging.verbose = lambda msg, *args, **kwargs: logging.log(logging.VERBOSE, msg, *args, **kwargs)

    numeric_level = getattr(logging, args.debugLevel.upper(), None)
    if not isinstance(numeric_level, int):
        raise ValueError('Invalid log level: %s' % args.debugLevel)
    if args.logPath == "-":
        print "Logging to stdout"
        logging.basicConfig(format="%(levelname)s: %(asctime)s - %(message)s", level=numeric_level, datefmt='%m
/%d/%Y %H:%M:%S', stream=sys.stdout)
    else:
        print "Logging to {}".format(lPath)
        logging.basicConfig(format="%(levelname)s: %(asctime)s - %(message)s", level=numeric_level, datefmt='%m
/%d/%Y %H:%M:%S', filename=lPath)

    logging.verbose("test verbose")

if __name__ == "__main__":
    main()
```

1. Create Branches via branching model
2. create feature branch
3. etc.