

Our initial exploration of the problem space lead us to chose to make use of the Ruby programming language as a basis for our solution. This lead to a number of efficiencies, however, there were some drawbacks that will be reported on later. We anticipate that our solution will run satisfactorily on recent MacOS and linux systems, however, we believe that there would need to be some modifications to enable it to run under Windows.

We started by creating an application structure that we feel is based upon common architecture. In our repository, in the bin folder, we wanted to have a single executable that provided an entry point to all functionality of our test agent. We chose to use the *thor*<sup>1</sup> ruby gem to handle most of the user interaction. Our initial framework setup a command line program allowing the following tasks:

- delete\_file
- make\_file
- modify\_file
- network\_connection
- start\_process

Each action was to be logged, we formatted our logfile with a JSON entry for each line in the logfile. A number of the log entry items required information about our process and other processes. We discovered while working on this requirement that there the ruby language standard library would not be sufficient for all of the logging data that we needed. We had several options available to us. We could have tried to access procs for our solution, however, this would limit our solution to only working with linux and windows systems running cygwin. Our alternative was to utilize a wrapper around the ps command <sup>2</sup>

---

<sup>1</sup><https://github.com/rails/thor>

<sup>2</sup><https://github.com/hungtatai/ps-ruby>