```
                step_dict = {}
                for tok in line[6:-1].split(' '):
                    key, value = tok.split('=')
                    step_dict[key] = float(value)

                if self.steps is None:
                    self.steps = [step_dict]
                else:
                    self.steps.append(step_dict)
        log.close()
        if not (self.steps is None):
            # Individual access to the Trace Classes, this information is stored in the Axis
            # which is always in position 0
            self._traces[0]._set_steps(self.steps)
            pass

    def __getitem__(self, item):
        """Helper function to access traces by using the [ ] operator."""
        return self.get_trace(item)

    def get_steps(self, **kwargs):
        if self.steps is None:
            return [0]  # returns an single step
        else:
            if len(kwargs) > 0:
                ret_steps = []  # Initializing an empty array
                i = 0
                for step_dict in self.steps:
                    for key in kwargs:
                        ll = step_dict.get(key, None)
                        if ll is None:
                            break
                        elif kwargs[key] != ll:
                            break
                    else:
                        ret_steps.append(i)  # All the step parameters match
                    i += 1
                return ret_steps
            else:
                return range(len(self.steps))  # Returns all the steps
```