Design Specification – Bash Script

Shell Script

- Shell script located in ~/Documents/linux/config called "fix-hostfile.sh" with a soft-link in ~/bin.
- create-soft-links.sh is updated to create this soft link.

Arguments

- restore : restores original hosts file, displays output
- prep : creates copy of original hosts file, displays output, calls hblock(1)
- DNS name to add with -a switch

Switches

- -a : add IP entry to allow.list, delete it from hosts.
- -f : DNS cache and restart the mDNSResponder service.
- -h : Display usage.

Globals

- DNS_FLUSH = FALSE
- ADD_DNS = FALSE
- HOSTS=/etc/hosts
- HBLOCK=/etc/hblock
- ALLOW_LIST=/etc/$HBLOCK/allow.list

`main()`

- Parse args and switches via process_arguments()
- IF arg1 == restore, call restore_hosts_file()
- IF arg1 == prep, call prep_hosts_file()
- IF ADD_DNS = TRUE, call add_dns_name()
- IF DNS_FLUSH = TRUE, call dns_flush()
- exit

`usage()`

- echo `usage: basename $0 [ahf] <prep | restore> [DNS entry]`
- exit 1

`process_arguments()`

- IF malformed input, call usage(), THEN exit_failure
- IF -h switch, call usage(), THEN exit_success
- IF -f switch, set DNS_FLUSH = TRUE
- IF -a switch, set ADD_DNS = TRUE
- return

`prep-hosts-file()`

- cd /etc
- echo “Existing hosts files”
- sudo ls -las hosts*
- Check for existence of files: hosts and hosts-ORIG.
    - IF hosts file doesn’t exist, display error and exit_failure.
- sudo cp hosts{,-ORIG}
- echo “Running hblock to create new hosts file. . . ”
- hblock
- echo “Updated hosts files”
- sudo ls -las hosts*
- return

`restore_hosts_file()`

- cd /etc
- echo “Existing hosts files”
- sudo ls -las hosts*
- Check for existence of files hosts and hosts-ORIG.
    - IF either file doesn’t exist, display error and exit_failure.
- Check for existence of hosts-HBLOCK. IF it exists,
    - Warn that this action will delete this file.
        - Query user to continue or exit.

        - IF continue
        - sudo mv hosts{,-HBLOCK}
        - sudo cp hosts{-ORIG,}
        - echo “Updated hosts files”
        - sudo ls -las hosts*
- return

`add-dns-name(name)`

- Verify <name> has been passed to this function
- echo “Adding <name> to /etc/hblock/allow.list”
- cd /etc/hblock
- Check if <name> already exists in allow.list. IF present
    - echo “Entry <name> already exists in allow.list”
- ELSE
    - cat » allow.list <name>
        - echo “Updated allow.host”
- cat allow.list
- cd /etc
- Verify <name> is in hosts file. IF present,
    - echo “Removing <name> from /etc/hosts
        - sed command to delete <name> from hosts file
- ELSE

    - echo “Entry <name> is not present in hosts file”
- echo “Actions completed”
- return

`dns-flush()`

- echo “Flushing DNS cache. . . ”
- sudo dscacheutil -flushcache
- sleep
- echo “Restarting the mDNSResponder service. . . ”
- sudo killall -HUP mDNSResponder
- sleep
- echo “New mDNSResponder PID”
    - Display mDNSResponder PID
- return

TODO

- ☒ Research if there’s a way to validate proper DNS name syntax
- ☒ Allow multiple switches on command line at the same invocation
- ☒ Test script on real /etc/hosts files
- ☒ Test running hblock
- ☒ Create git repo
- ☒ Create README
- ☒ Create manpage
- ☒ Create makefile to make manpage, deploy script, create softlinks
- ☒ Export this spec to a markdown file

LESSONS & NOTES

- Bash doesn’t like variable names that contain a hyphen.
    - My original global variable name was DNS-FLUSH, but this results in bash returning “command not found”.
        - Bash *does* accept underscores in variable names, so DNS_FLUSH works fine.
- Using getopts or getopt — pro’s and con’s
    - I wanted to add long switches in addition to the short switches. Trying that was educational!
        - getopts only supports single letter switches. It is also a POSIX-compliant builtin.
        - getopt supports long-name switches (.e.,g —help along with -h), but it is not POSIX compliant, is not guaranteed to be on a user’s linux box, and proved to be a real PITA to use! It cost me hours trying to debug it. Further, the code suggested by the various AI’s failed — none of ’em worked!