Variable names are always directly quoted when expanded. “$VARIABLE”.

Exported variables are always all caps. Variables used within the script are CamelCase. This convention prevents collisions.

$HOME is inherited from the environment. This is a security problem because it could be set to anything. ~ is just a shorthand for $HOME so using that doesn’t help. To plug this hole use the ‘home’ command in /usr/local/bin. That takes the home directory listing from the password file. If you want to use the HOME variables then set it at the top of the script:

HOME=/usr/local/bin/home

The unix date command ignores the locale. Use the /usr/local/bin/Z command. It also ignores locale, but it outputs in iso8601 date format, which is the company convention.

Wish I had a fix for this, rather be using exec -c than env -i:

exec -c and env -i, where INSIDE\_EMACS is set in the calling shell and is also set in

the init script of the newly launched shell, does not work. In both cases emacs stops

tracking directory changes in the newly launched shell. Emacs will continue to track

directory changes if we call bash directly, or if we use env -i with INSIDE\_EMACS defined as a parameter.

PATH is also inherited from the environment. Scripts make use of this to change which programs get executed, for example pythons ‘activate’. So we can’t reset it. This is a bigger security problem than HOME, but we can’t always reset it. The best we can do is to echo it to the transcript so the user can see it. We will echo it again in a script after changing it.