

Straightforward Shell Script Command Line Argument Parsing

Brendan Sechter

2017-01-27

This post covers a relatively straightforward method of parsing commandline arguments. All commandline arguments take the form of **-p=value** or **-parameter=value**. A safe delimiter is used internally so the value can contain an equals sign.

Software Versions

```
$ date -u "+%Y-%m-%d %H:%M:%S +0000"
2017-01-25 14:27:00 +0000
$ uname -vm
FreeBSD 12.0-CURRENT #18 4f888bf(drm-next): Wed Jan 18 14:31:26 UTC 2017    root@gauntlet:/usr
```

Instructions

The basic strategy looks like this, with a safe delimiter.

```
parse_args() {
  SAFE_DELIMITER="$(printf "\a)"
  while [ "$1" != "" ]
  do
    PARAM=$(echo $1 | cut -f1 -d=)
    VALUE=$(echo $1 | sed "s/=/${SAFE_DELIMITER}/" | cut -f2 "-d${SAFE_DELIMITER}")

    case $PARAM in
      -h | --help)
        usage
        exit
        ;;
      -v | --variable)
        G_VARIABLE="${VALUE}"
        ;;
      *)
        echo "ERROR: Unknown parameter ${PARAM}"
        usage
        exit 1
        ;;
    esac
    shift
  done
}
```

A complete template example looks something like this.

```

#!/bin/sh

set_defaults() {
    G_URL="https://www.google.com"
    G_CREDENTIALS="username:password"
}

usage() {
    set_defaults
    cat <<-EOF
./param.sh
Parameters:
-h | --help:
    Show this help.
-u | --url)
    URL
    Default: ${G_URL}
-l=username:password | --login=username:password)
    Username and password login credentials.
    Default: ${G_CREDENTIALS}
EOF
}

parse_args() {
    set_defaults
    SAFE_DELIMITER="$(printf "\a")"
    while [ "$1" != "" ]
    do
        PARAM=$(echo $1 | cut -f1 -d=)
        VALUE=$(echo $1 | sed "s/=/${SAFE_DELIMITER}/" | cut -f2 "-d${SAFE_DELIMITER}")

        case $PARAM in
            -h | --help)
                usage
                exit
                ;;
            -u | --url)
                G_URL="${VALUE}"
                ;;
            -l | --login)
                G_CREDENTIALS="${VALUE}"
                ;;
            *)
                echo "ERROR: Unknown parameter ${PARAM}"
                usage
                exit 1
                ;;
        esac
        shift
    done
}

main() {
    parse_args $@
}

```

```
    echo "URL=${G_URL}"
    echo "CREDENTIALS=$(echo "${G_CREDENTIALS}" | sed 's/:.*/:*****/')"
```



```
}

main $@
```