NAME

shell−quote − quote arguments for safe use, unmodified in a shell command

SYNOPSIS

shell-quote [*switch*]... *arg*...

DESCRIPTION

shell-quote lets you pass arbitrary strings through the shell so that they won't be changed by the shell. This lets you process commands or files with embedded white space or shell globbing characters safely. Here are a few examples.

EXAMPLES

ssh preserving args

When running a remote command with ssh, ssh doesn't preserve the separate arguments it receives. It just joins them with spaces and passes them to `$SHELL −c`. This doesn't work as intended:

```
ssh host touch 'hi there'           # fails
```

It creates 2 files, *hi* and *there*. Instead, do this:

```
cmd=`shell-quote touch 'hi there'`
ssh host "$cmd"
```

This gives you just 1 file, *hi there*.

process find output

It's not ordinarily possible to process an arbitrary list of files output by **find** with a shell script. Anything you put in `$IFS` to split up the output could legitimately be in a file's name. Here's how you can do it using **shell-quote**:

```
eval set -- `find -type f -print0 | xargs -0 shell-quote --`
```

debug shell scripts

shell-quote is better than **echo** for debugging shell scripts.

```
debug() {
    [ -z "$debug" ] || shell-quote "debug:" "$@"
}
```

With **echo** you can't tell the difference between `debug 'foo bar'` and `debug foo bar`, but with **shell-quote** you can.

save a command for later

shell-quote can be used to build up a shell command to run later. Say you want the user to be able to give you switches for a command you're going to run. If you don't want the switches to be re-evaluated by the shell (which is usually a good idea, else there are things the user can't pass through), you can do something like this:

```
user_switches=
while [ $# != 0 ]
do
    case x$1 in
        x--pass-through)
            [ $# -gt 1 ] || die "need an argument for $1"
            user_switches="$user_switches "`shell-quote -- "$2"`
            shift;;
        # process other switches
    esac
    shift
done
# later
eval "shell-quote some-command $user_switches my args"
```

OPTIONS

−−debug
Turn debugging on.

−−help
Show the usage message and die.

−−version
Show the version number and exit.

AVAILABILITY

The code is licensed under the GNU GPL. Check http://www.argon.org/˜roderick/ or CPAN for updated versions.

AUTHOR

Roderick Schertler <roderick@argon.org>