

```

#!/bin/bash
# makemakeurls -- interactive bash shell script for building the makeurls
# script from user input.
#
# Marion Bates   mbates@ists.dartmouth.edu
# ISTS/Dartmouth College   Hanover, NH
#
# makemakeurls   Copyright (C) 2001 Marion Bates   GNU/GPL
# Last revision April 24, 2002. Slight change for directories with and
# without trailing slash, made by Vince.
#

####
# IMPORTANT: Change HOME to where you want the script to be created.
# If you're unfamiliar with the syntax, just comment out the line below
# (by putting a # in front of it) and add a line below that which says
# "HOME=/path/you/want/to/use" (no quotes though.) Make sure that this
# path does NOT end with a slash.
####

# Get HOME
test $1 && HOME=$1

# Default to N so we loop through the question at least once
CHECK1="N"
CHECK2="N"
CHECK3="N"
CHECK4="N"

echo ""
echo "This script will generate makeurls, which in turn will reside on the REAL
WEBSERVER and will produce a list of valid URLs whenever you run it. The proxy will
then fetch this URL list using another script."
echo ""
echo "My HOME variable is set to $HOME, so this is where I will create makeurls.
The makeurls script I generate will use this as its HOME, also, and it will make backups
of the URLs list there, too."
echo ""
echo "Is this correct? If so, please type Y or y and hit return.
If that's NOT correct, please answer N or n and edit this script such that
HOME is set correctly. (How about /usr/mrp?)"

read CHECK1
if [ "$CHECK1" = "Y" ] || [ "$CHECK1" = "y" ] ; then
echo ""
echo "OK here we go! Please answer the following questions, and I will generate the
makeurls script based on your answers."

while [ "$CHECK2" = "N" ] || [ "$CHECK2" = "n" ] ; do
echo ""
echo "What is the path to the html directory? Do NOT include a trailing slash.
Note: If you have images and html files in separate directories, you can include the
parent directory, but be careful to check that the final URL list does NOT inadvertently
include any cgi scripts that you don't want the world to see."
read MUHTML
echo "You entered $MUHTML. Is this correct? Y(y) or N(n)"
read CHECK2
done

if [ "$CHECK2" = "Y" ] || [ "$CHECK2" = "y" ] ; then

```

```

while [ "$CHECK3" = "N" ] || [ "$CHECK3" = "n" ] ; do
echo ""
echo "What's the fully-qualified hostname of the real webserver?
Something like \"wwwfiles.yourdomain.com\" perhaps. (Do NOT include the trailing slash
or the starting \"http://\".)"
read HOSTNAME
echo "You entered $HOSTNAME. Is this correct? Y(y) or N(n)"
read CHECK3
done

if [ "$CHECK3" = "Y" ] || [ "$CHECK3" = "y" ] ; then

while [ "$CHECK4" = "N" ] || [ "$CHECK4" = "n" ] ; do
echo ""
echo "Who should own the listing (the backup copy of the URLs list)? (Hint: webmaster?)"
read LISTOWNER
echo "You entered $LISTOWNER. Is this correct? Y(y) or N(n)"
read CHECK4
done

if [ "$CHECK4" = "Y" ] || [ "$CHECK4" = "y" ] ; then

echo ""
echo ""
echo "Okay. Now generating the makeurls script."

echo "#!/bin/bash
# makeurls -- bash shell script to create the URL list on the webserver.
#
# Vincent Berk      vberk@ists.dartmouth.edu
# ISTS/Dartmouth College      Hanover, NH
#
# makeurls      Copyright (C) 2001      Vincent Berk      GNU/GPL

HOME=$HOME
HTML=$MUHTML
TODAY=listing.\`date +%m-%d-%y\`
LISTING=\$HOME/listings/\$TODAY
CGIFILE=\$HOME/listings/cgis
URLSFILE=\$HTML/urls

# First, take out the urls file to prevent it from showing up recursively
# but only if it exists. If not, make listings directory and create anew.
if \`test -e \$URLSFILE\`
then
    rm -f \$URLSFILE
    rm -f \$LISTING
else
    mkdir \$HOME/listings
fi

# Now, go build the listings
cd \$HTML

# Create the file
touch \$LISTING

# Insert the CGI file
if \`test -f \$CGIFILE\`

```

```
then
    echo \"# CGI scripts & options:\" >> \${LISTING}
    cat \${CGIFILE} >> \${LISTING}
else
    echo \"# No CGI file found\" >> \${LISTING}
fi

# Directories need an extra /, so we add it with awk
echo \"# Directories:\" >> \${LISTING}
find \${MUHTML} -type d | sed -e \"s@\${MUHTML}@http://\${HOSTNAME}@\" |
    awk '{ print \${0}\"/\n\${0} }' >> \${LISTING}

# Regular files -- note the slashes
echo \"# Regular Files:\" >> \${LISTING}
find \${MUHTML} -type f | sed -e \"s@\${MUHTML}/@http://\${HOSTNAME}/@\" >> \${LISTING}

# Symlinks -- note the slashes
echo \"# Symlinks:\" >> \${LISTING}
find \${MUHTML} -type l | sed -e \"s@\${MUHTML}/@http://\${HOSTNAME}/@\" >> \${LISTING}

# Create the current symlink
rm -f \${HOME}/listings/current
ln -s \${LISTING} \${HOME}/listings/current

# Copy to the webdirectory
cp \${LISTING} \${URLSFILE}

# Make it (and its parent directory) belong to the listing owner
chown \${LISTOWNER} \${LISTING}
chown \${LISTOWNER} \${HOME}/listings/ " > \${HOME}/makeurls

# Make makeurls executable
chmod 755 \${HOME}/makeurls

echo ""
sleep 1
echo "All done."
echo ""
echo "Please note: You will probably need to be ROOT (or use sudo) to run makeurls
successfully."

fi
fi
fi
fi
```