

```

#!/usr/bin/perl
#####
#
# Document:      File Count by Extension
# Create Date:   October 25, 2000
# Agency:       Church of the Brethren Network
# Programmer:    Ronald J. Gordon
#
# This script is not copywrited because I am serving the Lord

#$| = 1; # Set output to flush directly (for troubleshooting)
$lib = "/library"; # Default path for loading libraries
require "cgi-lib.pl";

#####
# Obligatory declare

use CGI;
use cPanelUserConfig;
use CGI::Carp qw(fatalsToBrowser);
print "Content-type: text/html\n\n";

#####
# Declare variables

# unique variables to this script
$administrator = 'rongordon@cob-net.org';
$date = localtime(time);
$host = $ENV{"REMOTE_HOST"};
$basedir="/home/content/c/o/b/cobnetorg/html/cgi/";
$basedir="/home/mjlj0ngrv27f/public_html/cob-net.org/cgi";
$baseurl = "https://www.cob-net.org/cgi/";
$bg = 'https://www.cob-net.org/bg';
$root_web_path = "/home/content/c/o/b/cobnetorg/html";
$root_web_path = "/home/httpd/html/cob-net.org";
$root_web_path = "/home/mjlj0ngrv27f/public_html/cob-net.org";
$server_url = "https://www.cob-net.org";
$siteurl = "http://cob-net.org/";
$sitename = "Church of the Brethren Network";
$progname = 'https://www.cob-net.org/cgi/filecount.pl';
$domain = ("");
$domain = 'cob-net.org';

# unwanted files or directories
# @unwanted = ("wusage", "_xvti");

# The following outputs the CGI Header
#print &PrintHeader;

#####

```

```

# Create a loop to examine files in every directory
#
# We traverse the whole directory structure under $root_web_path
# and in doing so, we also parse the HTML files to see if they have
# the keywords and what their title is.
#
# The following sets up the initial variables
#
# @dirs is the array of directories as a placeholder for going back up
# the directory tree when we run out of files in a subdirectory.
#
# $cur_dir is the current directory number and is a reference to the
@dirs
# array.
#
# Directory Handles are straight ASCII and consist of "DIR" + $cur_dir

$number_of_hits = 0;
$cur_dir = 0;
@dirs = ($ROOT_WEB_PATH);
opendir("DIR$cur_dir", $dirs[$cur_dir]);

$end_of_all_files = 0;

#print "Start of Counting<BR><BR>";

while (!(($end_of_all_files)) {
    # The following is used to trace down the next file. If there is no
    # next file (the whole directory tree was traversed),
$end_of_all_files
    # is set to positive and then the whole process is ended.
    while (1) {
        $filename = &GetNextEntry("DIR$cur_dir", $dirs[$cur_dir]);
        $fullpath = "$dirs[$cur_dir]/$filename";

        &CountExts;

# CASE 1) File is null but we can still traverse back up the directory
#
        if (!(($filename) && $cur_dir > 0) {
            closedir("DIR$cur_dir");
            $cur_dir--;
            next;
        }

# CASE 2) File is null and no where else to go, so we end the searching
#
        if (!(($filename)) {
            closedir("DIR$cur_dir");
            $end_of_all_files = 1;
            last;
        }

# CASE 3) File is a directory so traverse down through it

```

```

#
    if (-d $fullpath) {
        if (-r $fullpath && -x $fullpath) {
            $cur_dir++;
            $dirs[$cur_dir] = $fullpath;
            opendir("DIR$cur_dir", $dirs[$cur_dir]);
            next;
        } else {
# Since the dir does not have r or x perms we go on
            next;
        }
    }
}

```

```

# CASE 4) The file is an unwanted file

```

```

#
#     $unwanted = 0;
#     foreach (@unwanted) {
#         if ($fullpath =~ /$_/) {
#             $unwanted = 1;
#         }
#     }
#     if ($unwanted) {
#         next;
#     }

```

```

# CASE 5) The file is really something to search

```

```

#

```

```

# So we break out of the while loop

```

```

    if (-r $fullpath) {
        last;
    } # Make sure the file is readable

```

```

    } # End of While (1)

```

```

} # End of While

```

```

print "<TT><BR>Church of the Brethren Network<BR>";
print "All Directory File Count By Extension<BR><BR>";

```

```

print "HTM = $htm<BR>";
print "TXT = $txt<BR>";
print "CGI = $cgi<BR>";
print "PL- = $pl<BR>";
print "GIF = $gif<BR>";
print "JPG = $jpg<BR>";
print "BMP = $bmp<BR>";
print "TIF = $tif<BR>";
print "PHP = $php<BR>";
print "RAM = $ram<BR>";
print "RA- = $ra<BR>";
print "MID = $mid<BR>";
print "WAV = $wav<BR>";
print "AU- = $au<BR>";

```

```
print "CLS = $cls<BR>";
print "JAR = $jar<BR>";
print "EXE = $exe<BR></TT>";
```

```
exit;
```

```
#####
#
#   subroutine: GetNextEntry
#
#   Usage:
#       &GetNextEntry(DIRECTORY_HANDLE, "directory_name");
#
#   Parameters:
#       DIRECTORY_HANDLE = handle to currently open directory
#       $directory = full path of directory
#
#   Output:
#       $filename = name of next file, null if no more files
#       in current directory
#
#####
```

```
sub GetNextEntry {

local($dirhandle, $directory) = @_ ;
while ($filename = readdir($dirhandle)) {
    if (($filename =~ /[a-zA-Z]/) || (!($filename =~ /^\.\.?$/)) && -d
"$directory/$filename")) {
        last;
    }
} # End of while still stuff to read

# filename will be valid if it is a directory or an HTML file

$filename;

} # End of GetNextEntry

#if (($filename =~ /htm.?/i) || (!($filename =~ /^\.\.?$/)) && -d
"$directory/$filename")) {
```

```
#####
#   Count the number of files by extensions
#
sub CountExts {

if (lc($fullpath) !~ /usage/) {
    if (lc($filename) !~ /^_/) {
        if ($filename =~ /\.htm/) { $htm++; }
        if ($filename =~ /\.txt/) { $txt++; }
        if ($filename =~ /\.cgi/) { $cgi++; }
    }
}
}
```

```

        if ($filename =~ /\.pl/) { $pl++; }
        if ($filename =~ /\.gif/) { $gif++; }
        if ($filename =~ /\.jpg/) { $jpg++; }
        if ($filename =~ /\.bmp/) { $bmp++; }
        if ($filename =~ /\.tif/) { $tif++; }
        if ($filename =~ /\.php/) { $php++; }
        if ($filename =~ /\.ram/) { $ram++; }
        if ($filename =~ /\.ra/) { $ra++; }
        if ($filename =~ /\.mid/) { $mid++; }
        if ($filename =~ /\.wav/) { $wav++; }
        if ($filename =~ /\.au/) { $au++; }
        if ($filename =~ /\.class/) { $cls++; }
        if ($filename =~ /\.jar/) { $jar++; }
#       if ($filename =~ /\.exe/) { $exe++; }
#       if ($filename =~ /\.exe/) { $exe++; print "$fullpath<BR>"; }
        if ($filename =~ /\.exe/) { $exe++; print ""; }
    } # End of check in 2nd filename
} # End of check in 1st filename

#print "$rg - $fullpath ---- $htm - $txt - $cgi - $gif - $jpg<BR>";
#print "$fullpath ---- $htm - $txt - $cgi - $gif - $jpg<BR>";

} # End of CountExts

#####
#   Validate the referring document

# validate the referring document
$referer = $ENV{'HTTP_REFERER'};
if ($referer =~ $domain -1) {
    print "<font size=5><B>Invalid referring
document</B></font><BR><BR>";
    print "<A HREF='http://www.cob-net.org/'>Church of the Brethren
Network</A>";
    print " library of proprietary CGI/PERL scripts<BR><BR><BR><BR>";
    print "We cannot validate the following unrecognized referring
document<BR>";
    print "[ <font color=BB0000>$referrer</font> ]<BR>\n";
    exit;
}

```