a Gateway to the U Mich Geography server

Jim Davis <davis@dri.cornell.edu>

- **Mail folder:** WWW Talk 1992 Archives
- **Next message:** Tom Libert: "Re: a Gateway to the U Mich Geography server"
- **Previous message:** Kevin Hoadley: "Smart server or smart client ?"

Date: Wed, 18 Nov 1992 08:17:00 -0500
From: Jim Davis <davis@dri.cornell.edu>
Message-id: <199211181317.AA19290@willow.tc.cornell.edu>
To: www-talk@nxoc01.cern.ch
Subject: a Gateway to the U Mich Geography server

Here is a quickly hacked up Gateway from WWW to the University
of Michigan Geography server. It expects one argument, a
WWW doc id. It ignores the "pathname", extracts the search words,
then passes those to the server. It does NOT parse the data
returned by the server (that is an improvment yet to be done)
but you can understand the output.

To use this, you would need to have an HTTP server running
someplace where you can attach this gateway. I can provide
the very simple HTTP server I use here, but this subject
is already documented in the WWW online documentation.

Best wishes

```
#!/usr/local/bin/perl
# Gateway from WWW to the Geography server at U Michigan
# Copyright 1992 Xerox.  All rights reserved.
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
# Jim Davis, Nov 17 1992

# To do:
# 1) Parse the returned info so as to display more neatly.
#    (See documentation at end of file its format.)
# 2) Handle case where multiple places match supplied name
# 3) Handle case where no place matches supplied name.

# This expects one argument, a document ID.  The search keywords
# are sent to the Geography server, the "pathname" is ignored.
# e.g. /geography/ignored/whocares?ithaca,ny => ithaca,ny
# As an extra bonus feature, if the first arg has no question marks
# then the entire arg line is passed to the server.  This is
# convenient when you call this from the shell rather than from WWW.

if ($#ARGV == -1) {
 die "Usage: geography-gateway [WWW-DOCUMENT-ID | location]\
An example WWW document id is /geography?washington,dc
If calling from shell, omit the /geography? part.\nStopped";}

$usage =  "<ISINDEX>\n\
<TITLE>Geography Server</TITLE>\n\
<h2>Type geographic name in Index.</h2>\n\
```

```
";

# for debugging
# print "<LISTING>\n";
# print "ARGV =  $#ARGV\n";
# for ($i = 0; $i < $#ARGV+1; $i++) {
#   print "Arg $i is $ARGV[$i]\n";}
# print "</LISTING>";
# exit;


@parts=split('/', $ARGV[0]);  # separate pathname from keywords.
$last = $parts[$#parts];      # the last path component has keys
$i=index($last,"?");          # are there any keywords in there?
if ($i == -1)
   {@keywords = @ARGV;}          # none, probably called from shell
 else
   {@keywords=split('\?', substr($last,$i));
    $parts[$#parts] = substr($last,0,$i);}


# if no keywords, tell user how to type them.

if ($#keywords == -1) {
  print $usage;
  exit;
  };


# debugging
# print "<LISTING>\n
# parts are @parts\n
# keywords are @keywords\n
# </LISTING>\n";
# exit;

# otherwise contact the server

$host = "martini.eecs.umich.edu";
$port = 3000;

$sockaddr = 'S n a4 x8';          # packing format

$hostname = `hostname`;           # where I am calling from
chop($hostname);                  # get rid of trailing CR
($name, $aliases, $type, $len, $myaddr) = gethostbyname($hostname);
$here = pack($sockaddr, 2, 0, $myaddr);

($name, $aliases, $type, $len, $destaddr) = gethostbyname($host);

#print ("Name = ", $name ," " addr = ", $destaddr, "\n");
if ($destaddr eq "") {die "No such host: $host. Stopped";}

$there= pack($sockaddr, 2, $port, $destaddr);

($name, $aliases, $protocol) = getprotobyname('tcp');
socket($S, 2, 1, $protocol) || die "socket: $!";
bind ($S, $here) || die "bind: $!";
connect ($S, $there) || die "connect: $!";
```

```
select($S);
$| = 1;           # force output
print (@keywords ,"\n");
print "quit\n";
select(STDOUT);
print "<LISTING>\n";
while (<$S>) { print;}
print "</LISTING>\n";
select($S);
exit;

# This is the format of info returned by the geo server.
# in case you decide you want to parse it.

# 0 <city name>
# 1 <county FIPS code> <county name>
# 2 <state/province abbreviation> <state/province name>
# 3 <nation abbreviation> <nation name>
# A <telephone area code>
# E <elevation in feet above mean sea level>
# F <feature code> <feature name>
# L <latitude DD MM SS X> <longitude DDD MM SS X>
# P <1980 census population>
# R <remark>
# T <time zone>
# Z <postal ("ZIP") code>

# Example
# 0 Ithaca
# 1 36109 Tompkins
# 2 NY New York
# 3 US United States
# R county seat
# F 45 Populated place
# L 42 26 26 N  76 29 49 W
# P 28732
# E 814
# Z 14850 14851 14852 14853 14882
#
# .
```