hserveng

COLLABORATORS

	TITLE : hserveng	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 8, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

hserveng

1.1 Index

1.2 guide

This guide contains links with inline ARexx macros wich
use rmh.library/OpenURL. They will work if and only if
you have both openurl.library and rmh.ibrary installed.

rmh.library is by me, and it is included in this archive.

openURL.library is by Troels Walsted Hansen

"This library was created to make it easier for application programmers to
include clickable URLs in their applications, about windows, etc. ..."

You can find it on aminet at comm/www/OpenURL20.lha

1.3 warning

THIS SOFTWARE AND INFORMATION ARE PROVIDED "AS IS".
ALL USE IS AT YOUR OWN RISK, AND NO LIABILITY OR
RESPONSIBILITY IS ASSUMED. NO WARRANTIES ARE MADE.

1.4 introduction

hserv is a HTTP 1.0 compliant server with some features from HTTP 1.1 .

hserv is very configurable and funny and.

The main features are:
- run alone or as inetd service in multiple istance listening on different ports;
- accepts GET, HEAD and POST methods;
- basic file resuming;
- Virtual hosts implemented;
- "If-Modified-Since" supported, with many date formats;
- "rfc 19452 Auth supports;
- rejected-ips;
- mime types read from file, so you can had your mime;
- text parsing;
- in line ARexx functions in html documents;
- ARexx, REBOL, perl and exe cgi;
- may be set to require ident service running at client host;
- preferences and "QuickStart" GUI.

1.5 installation

First of all run the installation script.

Let's suppose you installed hserv in PATH: (complete PATH: to a drawer).
Follow this basic installation:

To install hserv as an inetd service:

- If running on AmiTCP
  open AmiTCP:db/services and (if not already present) add the line:
      http 80/tcp

```
  open AmiTCP:db/inetd.conf and add the line
      http stream tcp nowait root C:rxs rxs PATH:main/hserv.rexx

- If running on Miami
  open Miami/DataBase/services window
  press "Add" gadget
  (if not already present) add the entry:
      Name: http
      ID: 80
      Protocol: tcp
  open Miami/DataBase/InetD window
  press "Add" gadget
  add the entry:
      Service: http
      Socket: stream
      Protocol: tcp
      Wait: nowait
      User: root
      Server: c:rxs
      Name: rxs
      Args: PATH:main/hserv.rexx

To install it as stand alone:
    run >NIL: <NIL: rx PATH:main/hserv
    (or just rx hserv PATH:main/hserv)

Check if you have a services in inetd listening on 80/tcp or
hserv will fail with "can't bind socket" error.
```

1.6 requirements

```
hserv needs:

- AmigaOS 2.0 or higher
- ARexx
- Miami or AmiTCP/Genesis. hserv DOESN'T work on Termite.
- rxsocket.library
- rmh.library
- rxasl.library
- rxwiz.library
- wizard.library

NOTA BENE
wizard.library is not included in this archive,
but you can download it now.
```

1.7 author


```
I am Alfonso Ranieri .

My e-mail address is alfier@iol.it .
```

You can find me on:

- #amigaita ircnet ;
- #amyita   ircnet .

My home page is at http://users.iol.it/alfier/ .

1.8 distribution

hserv is FreeWare.

You are free to detribute it as long as the original archive is kept intact.
Commercial use or its inclusion in other software package is prohibited
without prior consens from the Author.

1.9 configuration

Configuration options are read from a file, which is the first argument of hserv .
If no file name is given, hserv tries to open /conf/hserv.conf

Each line of the configuration file is in the form:
<option> <value>

Empty lines, lines beginning with # or ;
lines after the 1024th, chars after the 256th
are ignored during parsing.

Words can be separated by space(s) and tab(s) .

The options are:

HostName	the host name, if none given, hserv tries to obtain it from the socket name Don't set it if you don't know what you are doing
Port	the port to use (of course, only valid if running as stand alone) default 80
Admin	any occurence of <!admin> in a text is converted to <A HREF="mailto:admin>admin</A>
DocumentDir	the dir where documents are
DocumentIndex	any request with a null file name is replaced with this default index.html
CGIDir	the directory where CGI macros are. It is used if and only if a "/cgi-bin/<file>" is received
ErrorLog	a flag that indicates if errors must be logged

```
                values are ON OFF SYS
                deafult SYS

ErrorFile       if ErrorLog is ON, log error in this file

TransferLog     a flag that indicates if requests must be logged
                values are ON OFF SYS
                default OFF

TransferFile    if TransferLog is ON, log requests int this file


            Auth
                       the auth file


            RejectedIP
                 the rejected ip - forces HostNamelookups ON


            MimeFile
                   the mime file


            Specials
                   the specials file


            Handlers
                   the handlers file


            VirtualHosts
               the virtual hosts file


            Errors
                     special errors file

Timeout         timeout in seconds for the receive request and send phase
                default 300

HostnameLookups a flags that indicates if the server should do a
                GetHostByName() on the client ip
                values are ON OFF
                default OFF

Ident           a flag that indicates if the client must have the
                ident service running; the server will connect it
                and get client ident
                values are ON OFF
                default OFF

OnlyAmigaClient only Amiga client accepted; clients without the word
                "Amiga" in User-Agent field are rejected
                values are ON OFF
                default OFF
```

```
Status          status of the server
                value are OPENED PAUSED CLOSED
                default OPENED

Pri             exec priority of the server process
                default 0

DefImage        send image on error
```

1.10 auth

```
The Auth file consists of lines in the form:
<pattern> <realm> <login> <pass>

- pattern is an AmigaDOS pattern, that will be applied to the complete
  file name requested.

- realm is a symbolic name for the protected space the file(s) belongs to

- login is the login

- pass is the password

Due ARexx limitation I set the max length of login":"pass to max 20 chars.

Anytime a file is requested, after it was parsed in a complete PATH: to a file ↩
   ,
and matches a pattern in a line of Auth file, an autentication request is send
to the client.

Try to join TheSecretGarden
you'll need to login with l:secret p:garden
```

1.11 rejectedip

```
The RejectedIP  file consists of lines in the form:
<pattern> <reasone>

- pattern is an AmigaDOS pattern, that will be applied to the ip
  of the client

- reasone is the reasone for the k-line . It is send to the client

If the pattern matches the ip, the connection is not accepted.

If a non empty Rejectedip file is given, HostNameLookups is set.
```

1.12 mime

```
The Mimefile consists of lines in the form:
<mime> <ext> {SP ext}

- mime is a mime type, e.g. <text/html>
- ext is the extension of the file.

Any file with an extension present in the left part of a mime line
is considered to be of that mime.

Default mime is <text/html>.
```

1.13 specials

```
The Special file consists of lines in the form:
<pattern> <code> <file> [args]

- pattern is an AmigaDOS pattern, that will be applied to the complete
  file name requested.

- code is on of:

  - CODE http error code, a http head is created and sent back
         with code "file" (must be integer) without any further
         action - args is discarded

  - CALL "file" is called and the current socket is pass to it
         ala inedt

- file it depends on the code value

- args arguments for CALL

An examples is

/#?/documentation.html CODE 301 Location: http://localhost/doc.html

that redirects documentation.html to the specified url .
```

1.14 handlers

```
The Handlers file consists of lines in the form:
<pattern> <handler>

- pattern is an AmigaDOS pattern, that will be applied to the complete
  file name requested.

- handler is one of
  - REXX   ARexx macro
  - REBOLE rebol script
  - CGI    perl cgi
  - EXE    executable
```

```
  - SEND   send the file

If the pattern matches the file name, the file is considered to be of
one of the above type . E.g. a line like

/CGI-BIN/counter REXX

will make hserv call the macro counter as it was a REXX macro rather
than an EXE macro .
```

1.15 virtualhosts

```
hserv implementation of virtual host is really siimple and
not even so much tested.

The VirualHosts  file consists of lines in the form:
<hostname> <DocumentDir>

On connection, the VirtualHosts file is opened and if the host
name of the request is found in a line, than the DocumentDir
specified is used; if the host name is not found, the default
DocumentDir is used. It also means, that for each virtual
host the default configuration is used, only the DocumentDir
can be changed.

To test virtual hosts, add in the host database of the stack you
are using, an entry like
127.0.0.1 www.alfie.org
Add in the VirtualHosts file a line like
www.alfie.org <path-to-somewhere>
and try to connect http://www.alfie.org with your browser.
```

1.16 errors

```
The Errors  file consists of lines in the form:
<httpCode/N> <macro> [newHttpCode]

If a http "httpCode" code is going to be returned by hserv,
e.g. the requeste document was not found and so hserv is
going to returns code 400, the Errors file is opened to
find a line with that code.
If the code is found, the macro specified is called as a
cgi (but the macro must be specified with complete path,
and doesn't have to be in the cgi-dir) and the output of
the macro is sent, with the originale code, if no newHttpCode
is specified, or with newHttpCode if present.
This is usefull to overwrite standard hserv answers.
```

1.17 runtime

```
hserv has a standard AmigaDOS template:
FILE,PORT/N
- FILE is the configuration file
- PORT is the port to listen to
       it only works if hserv is started as a stand alone service
       it can be 0 < port < 65535
       - if it is not given, it is read from the config file
       - if the config file has no PORT options it is set to 80 by default
       - if port is 0, the port is searched in the services database

If the server runs as stand alone:
- hserv opens a port called HSERV.PORT where port is the port it is listening ↩
   at
  this port accepts the commands:
  - QUIT         quit the server
  - CONF [file]  reload the old file, or a new one
  - SHOW         open HserPrefs with the current configuration file.

If the config file is modified, the server auto reloads it.

hserv (or better hs.rexx) does these controlls before sending a file:
- good request
- peer info
- Ident
- OnlyAmigaClient
- k-lined ip
- file contains "//" or ":"
- file exists
- Auth
- good request method
```

1.18 parsing

```
Any time a file of mime <text/*> is encontered, the server makes this parsing
on its' LINES:

changes the string <!ip> with the peer name

changes the string <!user> with the user name, if ident is ON, or unknown
if ident is OFF

changes the string <!userat> with the user user@ip, if ident is ON, or
ip if ident is OFF

changes the string <!admin> with the string
<A HREF="mailto:admin>admin</A>

changes the string <!power> with the string "Powered up with rxsocket.library"

changes the string <!ver> with the string "hserv 13.1"

changes the string <!this> with the current file name

changes the string <!InetDate> with the current date in GMT Internet format
```

changes the string <!include filename> and <!--#INCLUDE FILE="filename")
with the content of "filename" due ARexx limitation filename size must be
less then 65536 characters

changes the string <!REXX fun> with the output of the function fun, an Arexx
function called on the fly

changes the string <!CGI fun> with the result of the file created by the CGI
fun with the first 2 lines discarded

The parsing is made for all but !CGI then for !CGI, but never recursively for ↩
!CGI.

1.19 cgi

The term CGI refers to:
- ARexx macros
- perl macros
- REBOL script
- AmigaOS exe

The type of the macro is parsed from its file name extension:
- ARexx macros must have the extension "rexx"
- perl CGI macros must have the extension "cgi"
- REBOL script must have the extension "r"
- AmigaOS execs must have no extension

You can overwrite the "type" of a macro, specifying a line in the

Handlers
file.

A CGI macro of type perl will work if and only if you have perl
in your PATH , e.g. you have GeekGadgets installed.

A CGI macro of type REBOL will work if and only if you have rebol
in your PATH .

An exe macro can be an AmigaDOS script with the s bit set .

The directory CGIDir contains all the macro that are called in the form
"/cgi-bin/macro". It is a default directory for general porpouse macros.

CGI are called with the same arguments in the request.

They must write to stdout:
- first line: "Content-Type:" mime
- second line: (empty)
- rest: data (mime specific)

Macros have their real directory as CurrentDir.

If they are called from a <!CGI fun> the first 2 lines are discarded and
the rest is inserted.

ONLY file with mime text/* are parsed.

1.20 bug

- No way to pass cgi macros local vars, like the ip of the client,
  the arguments of a get or a post and so on, e.g. like apache does.

- That's not a bug:
  "If-Modified-Since" works just on AWeb, ask V and IB why, not me.

1.21 gui

hserv cames with a preferences editor GUI.

The GUI can be run via HservPrefs icon/macro.

It's template is:
FILE,PUBSCREEN/K
- FILE   the file to load
- PUBSCREEN the screen where to open
(- SERVER   internal, do not use)

The GUI is really easy to use, so there isn't so much to say.

At start it searches for:
- a file given as first argument
- conf/hserv.conf if present it exists

If the gui is started by hserv itself via the ARexx command SHOW
an optional gadget "Close server" is present: it closes that istance
of hserv.

QuickStart is a little gui used to quickly run different istances of hserv.