wdv-notes 180
Stand: 15.JUN.1993 (4.)
Wiss.Datenverarbeitung
FREIE UNIVERSITÄT BERLIN

Das Internet: Benutzung der Archiv-Server.

© 1991–1993 *Karl-Heinz Dittberner*

Net

Für Software-Recherchen werden im internationalen Com-puternetz für die Wissenschaften – dem Internet [4] – mehrere Archiv-Server betrieben, die u. a. mit einer interaktiven TELNET-Verbindung (wdv-notes Nr. 25) [1] zu erreichen sind.

In diesem Merkblatt wird die Benutzung dieses kostenlosen Internet-Services beschrieben, der„Archie“ genannt wird. Die Angaben beziehen sich auf die Version 3 der Archie-Software der Bunyip Information Systems Inc. (Montreal).

1. Description:

This Internet information service allows the user to query a database containing a list of files which are available on hosts connected to the Internet. Software located through this service can be obtained by means of ftp (wdv-notes No. 26) [2]. Other Internet users who are not directly connected may use the services of various ftp-by-mail servers including ftpmail-@decwrl.dec.com.

Some archie systems track archive sites globally, others only track the archive sites in their country, or continent in order to reduce the load on trans-oceanic links.

The Internet Archive Servers

archie.ans.net	**147.225.1.10**	**ANSnet, New York, USA**
archie.au	**139.130.4.6**	**Deakin University, Geelong, Australia**
archie.cs.huji.ac.il	**132.65.6.5 (?)**	**Hebrew University, Jerusalem, Israel**
archie.doc.ic.ac.uk	**146.169.11.3**	**Imperial College, London, UK**
archie.funet.fi	**128.214.6.102**	**FUnet, Helsinki, Finland**
archie.kuis.kyoto-u.ac.jp	**130.54.20.1**	**University of Kyoto, Japan**
archie.luth.se	**130.240.18.4**	**Sweden**
archie.ncu.edu.tw	**140.115.19.24**	**Taiwan**
archie.nz	**130.195.9.4**	**New Zealand**
archie.rutgers.edu	**128.6.18.15**	**Rutgers University, New Jersey, USA**
archie.sogang.ac.kr	**163.239.1.11**	**Korea**
archie.sura.net	**128.167.254.194**	**SURAnet, College Park, Maryland, USA**
archie.th-darmstadt.de	**130.83.128.111**	**TH Darmstadt, Germany**
archie.unl.edu	**129.93.1.14**	**University of Nebraska, Lincoln, USA**
archie.wide.ad.jp	**133.4.3.6**	**Japan**

2. Using the Interactive (TELNET) Interface:

In order to use the interactive system you should use the following procedure:

1. Telnet [1] to the archie system closest to you. Do not use ftp for this, it will not work.
2. Login as user “archie” (no capitals, no password is required). The system should print a banner message and status report before presenting you with the command prompt.
3. Type help for complete information on the system. For full details, see next section.

3. Archie Commands:

Note that some archie server sites may disable some of the commands for reasons particular to their site. Arguments to commands shown in square brackets ‘[…]’ are optional; all others are mandatory.

find *pattern* or **prog** *pattern* — This command produces a list of files matching the pattern *pattern*. The *pattern* may be interpreted as a simple substring, a case sensitive substring, an exact string or a regular expression (see section 5), depending on the value of the search variable. The output normally contains such information as the file name that was matched, the directory path leading to it, the site containing it and the time at which that site was last updated.

The format of the output can be selected through the output format variable. The results are sorted according to the value of the sortby variable, and are limited in number by the maxhits variable. The command prog is identical to find. It is included for backward compatibility with older versions of the archie system.

help [*topic* [*subtopic*] …] — Invokes the help system and presents help on the specified topic. A list of words is considered to be one topic, not a list of individual topics. Thus, “help set maxhits” requests help on the subtopic maxhits of topic set, not on two separate topics.

After help is presented the user is placed in the help system at the deepest level containing subtopics. For example, after typing “help set maxhits” and being shown the information for that topic the user is placed at the level set in the help hierarchy.

list [*pattern*] — Produce a list of sites whose contents are contained in the archie database. With no argument all the sites are listed. If given, the *pattern* argument is interpreted as a regular expression (see below) against which to match site names: only those names matching are printed. The format of the output can be selected through the output format variable.
Example: > list lists all sites in the database, while
> list .de$ lists all German sites.

mail [*address*] — Mail the output of the previous successful command (i.e. an invocation of find, list or whatis that produced output) is mailed to the specified electronic mail address.

If no *address* is given the contents of the mailto variable are used. If this variable is not set then an error occurs, and nothing is mailed, although the output is still available to be mailed. Conventional Internet addressing styles are used.
Example: > mail user@hello.edu

mailFP *address* — Mail the result of the last command that produced output (e.g. find, whatis, list) to *address*. This must be a valid email address.

manpage [roff | ascii] — Display the archie manual page. The optional arguments specify the format of the returned document. The roff specifies Unix troff(1) format while ascii specifies plain, preformatted ASCII output. Default is ascii.

motd — Re-display the “message of the day”, which is normally printed when the user initially logs on to the archie system.

quit — Quit the archie session and return to the local host.

servers — Display a list of all publicly accessible archie servers worldwide. The names of the hosts, their IP addresses and geographical locations are listed.

set *variable-name* [*value*] — Set the specified variable. Variables are used to control various aspects of the way archie operates; the interpretation of *pattern* arguments, the format of output from various commands, etc. See section 4 for a description of each one and the entries for unset and show.

show [*variable-name* …] — Without any argument, display the status of all the user-settable variables, including such information as its type (boolean, numeric, string), whether or not it is set and its current value (if its type requires a value). Otherwise show the status of each of the specified arguments.
Example: > show maxhits

site *sitename* — This command is currently unimplemented.

stty [[*option character*] …] — This command allows the user to change the interpretation of specified characters, in order to match their particular terminal type. At the moment only *erase* is recognized as an *option*. Typically, *character* is a control character and may be specified as a pair of characters (e.g. control-h). Without any arguments the command dis-

plays the current values of the recognized options.

unset *variable* — Remove any value associated with the specified variable. This may cause counter-intuitive behaviour in some cases; for example, if maxhits is not defined by the user, the find command will print the internal default number of matches rather than an unlimited number of matches.

version — Print the current version of the archie system.

whatis *substring* — Search the Software Description Database for the given substring, ignoring case. This database consists of names and short descriptions of many software packages, documents, and data files stored on the Internet.

Example: > whatis hpgl
in part gives as a result:

hpgl2ps	HPGL to PostScript translator
hpglplot	HPGL to LaserJet translator
yahp2ps	HPGL to PostScript converter

4. Archie Variables:

The behaviour of archie can be modified by certain variables, the values of which may be changed using the *set* command, or removed entirely by the *unset* command. There are three variable types: *boolean* (set or unset), *numeric* (Integer within a defined range), *string* (String of characters which may or may not be restricted). If the value of a string variable should contain leading or trailing spaces then it should be quoted.

autologout — Set the length of idle time (in minutes) allowed before automatic logout (range: 1–300; default: 60).
Example: > set autologout 30

compress — The kind of data compression the user can specify when mailing back output. Currently allowed values are *none* and *compress* (standard Unix compress(1), with a default of *none*.

encode — The type of post-compression encoding the user can specify when mailing back output. Currently allowed values are *none* and *uuencode*, with a default of *none*. Note that this variable is ignored unless compression is enabled (via the compress) variable.

language — Allows the user to specify the language in which the help, etc. is presented. Currently the default value is English.

mailto — If the mail command is issued with no arguments, mail the output of the last command to the address specified by this string variable. Initially this variable is unset.
Example: > set mailto user@frobozz.com

maxhits — Allow the find command to generate at most the specified number of matches (permissible range: 0–1000; default: 1000).
Example: > set maxhits 100
halts prog after 100 matches have been found.

max split size — Approximate maximum size, in bytes, of a file to be mailed to the user. Any output larger than this will be split in pieces of about this size. This can be set by the user in the range 1024 to ~2 GByte with a default of 51,200 Byte.

output format — Affects the way the output of find and list is displayed. User settable, with valid values of machine (machine readable format), terse and verbose, with a default of verbose.

pager — Filter all output through the default pager (default: unset). When using the pager you may also want to set the term variable to your terminal type (see term variable).
Example: > set pager

search — The type of search done by the find (or prog) command. User settable with a range of *exact*, *regex*, *sub*, *subcase*, *exact regex*, *exact sub* and *exact subcase* with a default of *sub*. (The *exact x* types cause it to try *exact* first, then fall back to type *x* if no matches are found). The values have the following meanings:

exact – Exact match (the fastest method). A match occurs if the file (or directory) name in the database corresponds exactly to the user-given substring (including case). For example, this type of search could be used to locate all files called "xlock.tar.Z"

regex – Allow user-specified (search) strings to take the form of Unix ed(1) regular expressions. Note: unless specifically anchored to the beginning (with ^) or end (with $) of a line, ed(1) regular expressions (effectively) have ".*" prepended and appended to them. For example, it is not necessary to type
find .*xnlock.*
because find xnlock
suffices. In this instance, the regex match is equivalent a simple substring match. Those unfamiliar with regular expressions should refer to the section 5, entitled "Regular Expressions".

sub – Substring (case insensitive). This is the default. A match occurs if the file (or directory) name in the database contains the user-given substring, without regard to case.
Example: The pattern: is
matches any of the following:
islington
this
poison

subcase – Substring (case sensitive). As above, but taking case as significant.
Example: The pattern: TeX
will match: LaTeX
but neither of the following:
Latex
TExTroff

server – The Prospero server to which the client connects when find or list commands are invoked. User settable, with a default value of localhost.

sortby – Set the method of sorting to be applied to output from the find command. Typing the keyboard interrupt character (generally Ctrl-C on Unix hosts) aborts a search. Unlike previous versions of the archie system, version 3 does not allow partial results. The output phase may be aborted by typing the abort character a second time.

The five permitted methods (and their associated reverse orders) are:

none – Unsorted (default; no reverse order, though *rnone* is accepted).

filename – Sort files/directories by name, using lexical order (reverse order: *rfilename*).

hostname – Sort on the archive hostname, in lexical order (reverse order: *rhostname*).

size – Sort by size, largest files/directories first (reverse order: *rsize*).

time – Sort by modification time, with the most recent file/directory names first (reverse order: *rtime*).

status — Controls the display of a "spinner" during the database search, which indicates that we are awaiting results from the archie server. Set by default.

Note that information such as number of matches found and percentage of database searched is not available under the current version of the archie system (as it was under previous versions).

term — Specify the type of terminal in use (and optionally, its size in rows and columns). This information is used by the pager.
The usage is:
> set term *terminal-type* [*#rows* [*#columns*]]
The *terminal type* is mandatory, but the number of rows and columns is optional; specify either rows only, or both rows and columns (default: 24 rows, 80 columns). The default value for this variable is dumb. However it may be set automatically through the Telnet protocol negotiation.
Examples: > set term vt100
> set term xterm 60
> set term xterm 24 100

Das Internet: Benutzung der Archiv-Server.

© 1991–1993 *Karl-Heinz Dittberner*

5. Regular Expressions:

Regular expressions follow the conventions of the Unix ed(1) command, allowing sophisticated pattern matching. In the following discussion, the string containing a regular expression will be called the "pattern", and the string against which it is to be matched is called the "reference string". Regular expressions imbue certain characters with special meaning, providing a quoting mechanism to remove this special meaning when required. The rules governing regular expression are:

c — A character c matches itself unless it has been assigned a special meaning as listed below. A special character loses its special meaning when preceded by the character '\'. This does not apply to '{', which is non-special until it is so treated. Thus, although '*' normally has special meaning, the string '*' matches itself.

Example: The pattern acdef
matches any of the following:
s83acdeffff
acdefsecs
fooacdef
but neither of the following:
accdef
aacde1f

Example: Normally the characters '*' and '$' are special, but the pattern a*bse\$ acts as above.
Any reference string containing:
a*bse$
as a substring will be flagged as a match.

. — A period (known as a wildcard character) matches any character except the newline character.

Example: The pattern .... will match any 4 characters in the reference string except a newline character.

^ — A caret (^) appearing at the beginning of a pattern requires that the reference string must start with the specified pattern (an escaped caret, or a caret appearing elsewhere in the pattern, is treated as a non-special character).

Example: The pattern ^efghi will match only those reference strings starting with efghi; thus, it will match either of the following:
efghi
efghijlk
but not: abcefghi

**$** — A dollar sign ($) appearing at the end of a pattern requires that the pattern appear at the end of a reference string (an escaped dollar sign, or a dollar sign appearing elsewhere, is treated as a regular character).

Example: The pattern efghi$
will match either of the following:
efghi
abcdefghi
but not: efghijkl

[**string**] — Match any single character within the brackets. The caret (^) has a special meaning if it is the first character in the series: the pattern will match any character other than one in the list.

Example: The pattern [^abc]
will match any character except one of: a b c
To match a right bracket (]) in the list, put it first, as in: [ ]ab01]

A caret appearing anywhere but the in first position is treated as a regular character. The minus (–) character is special within square brackets. It is used to define a range of ASCII characters to be matched. For example, the pattern: [a–z] matches any lower case letter. The minus can be made non-special by placing it first or last within the square brackets. The characters '$', '*' and '.' are not special within square brackets.

Example: The pattern [ab01] matches a single occurrence of a character from the set:
a b 0 1

Example: The pattern [^ab01] will match any single character other than one from the set:
a b 0 1

Example: The pattern [a0–9b] matches one of the characters: a b or a digit between 0 and 9, inclusive.

Example: The pattern [^a0–9b.$] matches any single character which is not in the set: a b . $ or a digit between 0 and 9, inclusive.

***** — Match zero or more occurrences of an immediately preceding regular expression.

Example: The pattern a* matches zero or more occurrences of the character: a

Example: The pattern [A–Z]* matches zero or more occurrences of the upper case alphabet.

\{m\} — Match exactly m occurrences of a preceding regular expression, where m is a non-negative integer between 0 and 255 (inclusive).

Example: The pattern ab\{3\} matches any substring in the reference string consisting of the character 'a' followed by exactly three 'b' characters.

\{m,\} — Match at least m occurrences of the preceding regular expression.

Example: The pattern ab\{3,\} matches any substring in the reference string of the character 'a' followed by at least three 'b' characters.

\{m,n\} — Match between m and n occurrences of the preceding regular expression (where n is a non-negative integer between 0 and 255, and n > m).

Example: The pattern ab\{3,5\} matches any substring in the reference string consisting of the character 'a' followed by at least three but at most five 'b' characters.

Tips for Using Regular Expressions:

1) When matching a substring it is not necessary to use the wildcard character to match the part of the reference string preceding and following the substring.
   Example: The pattern abcd will match any reference string containing this pattern. It is *not* necessary to use .*abcd.* as the pattern.
2) In order to constrain a pattern to the entire reference pattern, use the construction: ^pattern$
3) The '[ ]' operator provides an easy mechanism to obtain case insensitivity. For example, to match the word: hello regardless of case, use the pattern: [Hh] [Ee] [Ll] [Ll] [Oo]

6. Beispiel einer Software-Recherche:

Zur Demonstration einer Software-Recherche ist im folgenden eine interaktive TELNET-Verbindung [1] mit dem Archiv-Server der Technischen Hochschule in Darmstadt dargestellt.

Bei dieser Recherche sollen für das Software-Paket „Kermit" alle Fundstellen festgestellt werden. Außerdem soll ermittelt werden, ob bereits kostenlose Programme zum Konvertieren von Plot-Dateien von HPGL nach PostScript zur Verfügung stehen.

Die Ergebnisse der Recherche sollen – wie das allgemein üblich ist – per Electronic Mail (E-Mail) an die eigene Mailbox [3] geschickt werden, um diese auswerten zu können.

Eine solche Software-Recherche kann von jedem Terminal des HAIRclusters (VAXcluster) durchgeführt werden. Alle Benutzereingaben sind im folgenden fett gedruckt.

```
VAXUS>  TELNET  archie.th-darmstadt.de  <Return>
                        ...
AIX telnet (rs1.hrz.th-darmstadt.de)

IBM AIX Version 3 for RISC System/6000
(C) Copyrights by IBM and by others 1982, 1991.
login: archie  <Return>

       Welcome to archie.th-darmstadt.de

# Bunyip Information Systems, 1993
# Terminal type set to 'vt100 24 80'.
# 'erase' character is '^H'.
# 'search' (type string) has the value 'sub'.

th-archie> set mailto dit@vaxser.grumed.fu-berlin.de  <Return>
th-archie> set sortby hostname  <Return>
th-archie> whatis kermit  <Return>

c-kermit.ann   C-Kermit & USENET
ckermit        The 'C' implementation of Kermit
kermit         Communications software package
kermit.hdb     Kermit patches to enable dial to use HDB database

th-archie> find kermit  <Return>
working...

Host ariel.unm.edu    (129.24.8.1)
Last updated 23:30 10 Jun 1993
    Location: /pub/docs
      DIRECTORY    drwxr-xr-x 512 bytes 01:00 28 Jul 1992  kermit

Host ftp.dkfz-heidelberg.de    (192.54.49.67)
Last updated 04:05  6 Jun 1993
    Location: /pub/msdos
      DIRECTORY    drwxr-xr-x 512 bytes 16:24  4 Mar 1992  kermit

Host ftp.slu.se    (130.238.98.11)
Last updated 00:03 11 Jun 1993
    Location: /pub/mac
      DIRECTORY    drwxrwxr-x 512 bytes 05:04 22 Jan 1993  kermit

Host ftp.uni-paderborn.de    (131.234.2.32)
Last updated 19:40  8 Jun 1993
    Location: /pcsoft/msdos
      DIRECTORY    drwxr-xr-x 1536 bytes 10:52  2 Jun 1992  kermit

Host gatekeeper.dec.com    (16.1.0.2)
Last updated 23:35 10 Jun 1993
    Location: /.0/comm
      DIRECTORY    dr-xr-xr-x 512 bytes 05:37 18 May 1993  kermit

Host gatekeeper.dec.com    (16.1.0.2)
Last updated 23:35 10 Jun 1993
    Location: /.2/micro/msdos/simtel20
      DIRECTORY    dr-xr-xr-x 1024 bytes 05:42 18 May 1993  kermit

Host lilac.berkeley.edu    (128.32.136.12)
Last updated 23:17 10 Jun 1993
    Location: /help/cat/micros/software/mac
      FILE    -rw-r—r— 1661 bytes 02:02 29 Apr 1993  kermit
                        ...

th-archie> mail  <Return>

th-archie> whatis hpgl  <Return>

hpgl2ps        HPGL to PostScript Translator
hpglplot       Hpgl to LaserJet translator
yahp2ps        HPGL to PostScript converter

th-archie> find HPGL  <Return>
working...

Host augean.eleceng.adelaide.edu.au    (129.127.4.2)
Last updated 22:55  9 Jun 1993
    Location: /pub
      FILE    -r—r—r— 42682 bytes 00:00  6 Apr 1990  hpgl2ps.tar.Z

Host bode.ee.ualberta.ca    (129.128.16.96)
Last updated 14:27  1 Jun 1993
    Location: /pub/dos/util
      FILE    -rw-rw-rw- 167259 bytes 02:00 29 Oct 1992  hpgl2ps.zip

Host bode.ee.ualberta.ca    (129.128.16.96)
Last updated 14:27  1 Jun 1993
    Location: /pub/unix/UTIL
      FILE    -rw-r—r— 29130 bytes 02:00 28 Oct 1992  hpgl2ps.tar.Z
                        ...

th-archie> mail  <Return>
th-archie> quit  <Return>
# Bye.
```

Die vollständigen Software-Listen dieser Recherche sind nach wenigen Minuten per E-Mail auf dem HAIRcluster eingetroffen.

7. Sonstige Hinweise:

■ Sämtliche von den Archiv-Servern nachgewiesenen Programme und Software-Pakete sind „frei“, d. h. sie können kostenlos per „*Anonymous FTP*“ [2] über das Internet von den jeweils angegebenen Hosts bezogen werden (kopiert werden).

■ Im Internet werden weltweit zur Zeit über 1000 Software-Archive (Hosts) betrieben, die per „*Anonymous FTP*“ zu erreichen sind. Es stehen hier insgesamt über 2,5 Millionen Dateien zur Verfügung. Diese „freien“ Programme und Informationen haben derzeit einen Umfang von etwa 200 Gigabyte (das sind 200.000.000.000 Byte). Allein das Verzeichnis aller weltweit archivierten Dateien – das ist die Datenbasis der Archiv-Server – belegt derzeit über 250 Megabyte auf einer Platte.

■ Die Internet Archiv-Server lassen sich auch per E-Mail [3] benutzen. Das ist dann interessant, wenn keine interaktive Verbindung möglich ist. Die E-Mail Adresse ist **archie@*archie-server***, wobei für *archie-server* die Adresse des gewünschten Archiv-Servers (siehe Liste auf der Seite 1) eingesetzt wird. Im (elektronischen) Brief werden nacheinander die für die Recherche erforderlichen Befehle aufgeführt, wobei jeder Befehl auf einer neuen Zeile stehen muß. Der letzte Befehl ist „quit“.

■ Leider können beim Archie-System noch keine logischen Verknüpfungen von Suchbegriffen angegeben werden. Es sind deshalb einzelne Recherchen notwendig.

■ Das Archie-System wurde von der Firma Bunyip Information Systems Inc., Montreal, Kanada (info@bunyip.com) entwickelt.

■ Die Beschreibung des Archie-Systems besorgten R. P. C. Rodgers, UCSF School of Pharmacy, San Francisco, California (rodgers@maxwell.mmwb.ucsf.edu), N. H. F. Beebe (beebe@math.utah.edu) und A. Emtage (bajan@bunyip.com). Diese Originalbeschreibung wurde für dieses Merkblatt von der Abt. Wiss. Datenverarbeitung den lokalen Bedürfnissen angepaßt und überarbeitet.

8. Literatur:

[1] Dittberner, K.-H.: Benutzung des Internets: TELNET. FU Berlin (IfP): wdv-notes Nr. 25, 1991–1993.

[2] Dittberner, K.-H.: Benutzung des Internets: FTP. FU Berlin (IfP): wdv-notes Nr. 26, 1991-1993.

[3] Dittberner, K.-H.: Elektronische Post: MX-Mailing vom HAIRcluster. FU Berlin (IfP): wdv-notes Nr. 131, 1992–1993.

[4] Kehoe, Bendan P.: Zen and the Art of the Internet. A Beginner's Guide to the Internet. Chester (USA): Widener University 1992 (1. Edition).

***Notizen*:**

© khd-research • 2004 • Eine Publikation von http://www.khd-research.net/ • File: Notes/wdv-notes_180.pdf