Path: gmdzi!unido!mcsun!uunet!zaphod.mps.ohio-
state.edu!samsung!munnari.oz.au!manuel!cmf851

From: cmf...@anu.oz.au (Albert Langer)

Newsgroups: alt.sources.d,comp.archives.admin

Subject: Re: dl/describe (File descriptions) posted to alt.sources

Message-ID: <1991Aug7.225159.786@newshost.anu.edu.au>

Date: 7 Aug 91 22:51:59 GMT

References: <1991Aug7.124457.6814@csv.viccol.edu.au>
<1991Aug7.131048.6817@csv.viccol.edu.au>

Sender: ne...@newshost.anu.edu.au

Followup-To: comp.archives.admin

Organization: Computer Services Centre, Australian National University,
Canberra, Australia.

Lines: 291

Xref: gmdzi alt.sources.d:21891 comp.archives.admin:296

In article <1991Aug7.1...@csv.viccol.edu.au> ti...@csv.viccol.edu.au
(Tim Cook) writes:

>I have just posted a new version of "dl/describe" (previously known as
>"dls/describe") to alt.sources. This action was prompted by Christian
>Schlichtherle's <ch...@attron.ruhr.de> posting of "dls" and then "vls".

Competition works wonders!

>I have also added a few things. The "dl" command now has a recursive
>(-R) option, just like ls(1) [...]

I am very glad to see you have taken up this suggestion. But I still think
it is important for ALL options of ls (including the no options case)
to work IDENTICALLY to ls itself - especially when the program is
called with the name ($0) ls. That should be a VERY easy change to
make (just call ls :-) and should remove ANY reservations about
replacing ls with dl for ftp sites. (Descriptions should be added
whenever a special option, never used by either Sys V or BSD ls
is given, or perhaps also as a default when $0 is not ls, with an option
to suppress it.)

Another suggestion: It would be nice to allow multiple line
descriptions, even though that may be awkward with dbm. As well as
allowing for cases when one line just is not enough, this could
permit eventual transition to complete MARC records from the OCLC
project or other forms of more detailed cataloging. Also long
filenames should be handled transparently. A simple display convention

could be that anything starting in the first column is a new filename
(or a continuation of a previous ridiculously long filename that did
not end before the end of the previous line). Anything that starts
with whitespace is a continuation of a description from the previous
line or if at the top, is a description of the directory.
(A similar convention is used in FILES.BBS files that play a similar
role on some BBSes).

>#ifndef MODEST
>
>I would implore all those looking for a system of setting and listing
>descriptive file comments to investigate dl/describe. I think it is a
>very good solution (I expecially think the use of DBM files to store
>descriptions to be a superior method), and it has already been installed
>in several Anonymous FTP sites around the world. I plan on making it as
>portable as possible, and getting it to mesh with the archie system, and
>as many ftpd's as possible.
>
>#endif

I would like to endorse that strongly, with no risk of immodesty :-)

I hope you quickly add utilities for mv, cp, rm and ln as Chris did,
and also for processing MANIFEST files. These should be quite
trivial. (I also hope Chris upgrades vls for full compatability
with ls and to use dbm and that competition continues :-)

Again, these utilities should function IDENTICALLY to the normal
Sys V and BSD versions with ALL options, so that users can simply
include them ahead of the normal versions in their paths without
ANY scripts breaking. Incidentally, you might want to think about
a possible variation that redefines the appropriate system
calls themselves in an installable file system. As well as providing
transparent use of descriptions along with files for C programs
as well as shell scripts, this might be a way to implement long
file names and even symbolic links on a Sys V 3.2 installable
file system - thus earning eternal gratitude or perhaps hard
cash from many people unable to upgrade to Sys V R4.

SOLVING THE FTP AND CATALOGING PROBLEM

It looks like all the pieces are falling into place for a thorough
solution to ftp problems (X.500, archie, prospero, WAIS, the OCLC project,
Mark Moraes batched ftp and now dl or vls).

dl is much smaller than some of the others but I think it is also
VERY important so as to capture descriptions along with filenames
themselves. This provides the raw material for a decent indexing
facility for X.500, archie, prospero and WAIS without extra work adding
descriptions centrally. That is essential both as an interim measure and
to provide raw material for future upgrading to proper catalog
records when these have been defined by the OCLC project.

It should just become expected that anybody making a file
available for ftp will ALSO add a one line description - as
is usually the case on the most primitive BBSes. The "burden"

is unlikely to be resented as it makes the files that much more useful to the ftp site as well, and it is a FAR smaller burden for each ftp site than it would be for central database maintainers.

The quicker dl (or vls) gets widely used at ftp sites, the quicker it will be possible to find potentially interesting packages by keyword queries on archie etc instead of just by file name or portion of filename (or with restriction to only incomplete keyword indexes maintained centrally).

In the meantime it is a great utility for enhancing the normal use of any ftp site and simplifying administration of directory description files etc (and likewise for keeping track of files for any user on any unix system).

As soon as it gets widely used, tools will be needed, as Tim mentioned, for getting it to mesh with the archie system and with ftp. In particular:

1. Collection of descriptions by archie and delivery (optionally) along with filenames in response to queries.

2. Ditto for prospero.

3. Searching of raw descriptions by keyword queries for X.500, archie and WAIS.
It would be nice to also provide this within dl itself - or just as a simple utility to periodically do a recursive listing of descriptions and either index that file or use grep on it.

4. Facilities for reviewing multiple raw descriptions of the same filename and selecting one, or editing a new description that can optionally be used as a "revised" description instead of the raw description in response to a query (and for keyword searching).

5. Ways to pickup descriptions and merge them automatically in an ftp session. (One way involves running the dl command and processing the output locally, but an alternative may be to get the .pag file from each directory and process that - if it can be made machine independent).

6. A way to feed back "revised" descriptions from 4 to the individual ftp sites for optional replacement of or addition to their raw descriptions - perhaps using methods also developed for 5.

UNIQUE IDENTIFIERS

Finally, another issue that will arise is that of uniquely identifying files which may have different names and/or be in different directories on different systems (and also of being sure that files with the same name are identical - we don't even have the date preserved across ftp transfers and can only rely on the file size).

Specifying a file in a news article in the form host.domain:path/filename is fine as far as it goes. But any automatic extraction of that to

place a file request will go to the particular ftp site even though closer sites may have the same item. Cache and mirror sites can partially solve the problem, especially if extended more widely through an enhancement of Mark Moraes scripts, but there will still be a strong temptation to just take the easy way out, and not bother installing software that adds a delay consulting archie or any other method to check where to obtain a file locally.

If comp.archives and WAIS etc provide a unique identifier for each file which is independent of location, and there are convenient ways to automatically insert that identifier into a news article when referring to a file, then users would HAVE to lookup a directory before ftping the file, and could then be automatically informed of the nearest location. (This need be no burden on the user - they should be able to request by the unique identifier and have the request acted upon by the appropriate ftp archive in one operation while reading news or mail - at dial-up sites just as well as on the internet). Incidentally, funding this aspect of comp.archives and archie etc could be justified by NSFnet and regional networks as being concerned solely with bandwidth conservation rather than information value adding.

A simple method of defining a unique identifier that does NOT include a particular site identifier would be to use a hash function on the entire contents of the file. This can be generated locally without requiring a registration system and if long enough the chances of collision are negligible. I would suggest using a cryptographic hash function such as MD5 which generates a 16 byte result. The extra work to use cryptographic hashing is only done once when assigning the unique identifier and is therefore unimportant. But for any users that DO wish to check validity, it provides a VERY secure means of ensuring they have got an uncorrupted version of the specific file they were told about, regardless of where they can get it from. (There is currently no publicly known way to generate a file that would produce the same 16 byte MD5 code as any given file).

Instead of providing accession lists with bibliographic information in order to establish union catalogs, it should be quite simple for ftp sites to notify the MD5 codes and local directory path/filename of new files to central database servers. Use of MD5 could prevent possible sabotage of a system based on easily duplicated CRCs as well as providing a valuable service combating dissemination of viruses and serving various other authentication functions.

Utilities for inserting the unique code into a news article or mail message (along with a marker for automatic extraction) can simply calculate the MD5 function from a local copy of the file. But to speed things up it might be better to include the result in the local dl or vls system where it can be accessed quickly (though not normally displayed unless asked for). This could be combined with an enhancement to allow find like tree searching of ALL file descriptions rather than just those in a particular directory, and/or use a separate index by MD5 code. If a separate index is provided for a (hidden) MD5 subfield of the descriptions, similar indexing could be made available on other fields or on all words of the description at the same time.

A simple ftp implementation would just hardlink every file available for ftp

to a filename encoding of it's MD5 token. Users would then ftp the directory path and filename of the MD5 token and obtain the file. An archie or similar lookup could first determine which nearby systems have the file (though come to think of it, that database lookup may as well also provide the local directory and filename for it). For dial-up sites a mail-server request could be chained until it reached a site with directory access, and the files requested added to temporary caches on the way back.

For compatability with ALL Posix systems, only 14 character file and directory names are available. Using a simple 6 bit encoding with only 64 characters allows a filename to represent 88 bits. Directory names could be added to provide the remaining 40 bits or the MD5 code could be truncated if increased collisions and less security were acceptable.

Any collisions could be placed in a public central list, and the files affected assigned new unique identifiers (e.g. append the MD5 result to the file and try again). It probably would not be worth it, but individual sites could maintain copies of the public list so as to rename any files for which collisions later occur (or provide both identifiers where there is no local collision).

PACKAGES CONTAINING A DIRECTORY OR DIRECTORY TREE

A related problem is that essentially the same collection of information may be available as different .tar.Z or zoo or ZIP or shar files etc. This happens especially with files distributed through sources newsgroups and archived with different methods (or even with the same methods, but including the local headers, which are different). It will also happen where a local modification has been added to a package.

Ultimately these do have to be regarded as DIFFERENT files and any connections between them listed separately. Nevertheless a user may be wondering whether to ftp a package that has a new MD5 code to see if it contains new revisions and it would be nice to be able to tell the user without the need for collecting the entire package.

A simple convention should require that the code is always calculated on the raw file rather than on the .Z version (or equivalent for any other compression scheme). Also text files should be encoded from the unix form (ASCII code with LF as line end and TABs not expanded).

Likewise the code for a tar or cpio or ZIP archive etc or a collection of shar files (with or without uuencoding etc) could be the code obtained by applying MD5 again to the concatenation of the codes of the extracted files, in numeric order. (This deliberately loses any date and mode or ownership information and also loses the filename and directory structure information although there are arguments for retaining the latter and it could be done easily enough by preceding each MD5 code with the filepath relative to directory . as the top of the package).

That convention would help a lot, but does not solve the problem concerning packages that ARE slightly different.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.