Jaquith cribsheet V 1.1

Introduction

Jaquith is an automated tape librarian for networked Unix systems. It's goal is to make access to tape storage moderately convenient by hiding the details about data storage location and format.

Jaquith manages a pool of tapes called the physical archive. This physical space is partitioned into named areas called logical archives. Users ainstruct Jaquith to move files between disk storage and a specified logical archive using one of the client programs described below. Jaquith always moves whole files, never parts of a file.

Each file is stored in the archive with the same name that it had on disk. Consequently, the archive represents snapshots of the disk name space at various points in time. Any of the versions of a file can be retrieved and reinstalled on disk.

Jaquith provides three client utilities:

- jput - Put files out onto the archive.
- jget - Retrieve files from archive.
- jls - List files in archive.

All of the utilities have default parameters which can be overridden on the command line. In addition they recognize the environment variables `JSERVER`, `JPORT`, and `JARCHIVE`. These represent the host name where the server is running, the port it's listening on, and the default logical archive respectively.

Below is a synopsis of the utilities. See the man pages for for more details.

jput

Jput writes disk files out to the archive. It's two important defaults are (1) it writes directories out recursively and (2) it only writes files that have been modified since the last time they were archived.

Examples:

`jput foo.c`	Write file *cwd*/foo.c conditionally
`jput -force foo.c`	Write file *cwd*/foo.c unconditionally
`jput -abs "version 1.0" /foodir`	Write out directory /foodir and its subdirectories tagging each file with "version 1.0"
`jput -prune "*.md" /foodir`	Write out directory /foodir and its subdirectories unless the name matches globbing expression `*.md`. Note that the metacharacter (`*`) is quoted.

jget

Jget retrieves files from the archive and restores them to disk. The important aspects of `jget` are (1) it restores directories recursively, (2) it supports wildcard searches and (3) it retrieves the latest version by default. n Examples:

`jget foo.c`	Restore *cwd*/foo.c with original name and permissions
`jget -abs "ver.* 1" /foodir`	Restore version of directory `/foodir` and its subdirectories that matches regular expression "ver.* 1"
`jget -target /tmp /foodir`	Restore `/foodir` relative to `/tmp`, not to its original location.
`jget -dir /foodir`	Restore top level `/foodir` only.
`jget -asof 1-mar-1990 /foodir`	Restore `/foodir` as it was on 1-mar-1990.

jls

Jls lists file names in the archive. It's useful for searching the archive before doing a lengthy `jget`. `Jls` duplicates the options of jget but returns metadata only. Unlike `jget`, it is not recursive by default. It has several formatting options modelled after `ls` but it does not approach the complexity of `ls`.

Examples:

`jls foo.c`	List *cwd*/foo.c
`jls -l foo.c`	List *cwd*/foo.c in long form
`jls -R /foodir`	List contents of directory `/foodir` recursively
`jls -a /foodir`	List contents of directory `/foodir` showing abstract
`jls -all -l -d /foodir`	List all versions of `/foodir`
`jls -l /foodir`	List all files in last version of `/foodir`
`jls -l "/foodir/*"`	List last version of all files that exist in any version of `/foodir`
`jls -all -l /foodir`	List all versions of `/foodir` showing contents
`jls -range (jan-1990,mar-1990) /foodir`	List versions of `/foodir` with last mod dates between January 1990 and March 1990
`jls -first 1 -last 2 /foodir`	List first two (oldest) versions of `/foodir`