tractor-spool

Contents

The spooler delivers job scripts into the Tractor job queue for processing and distribution on the farm. Job scripts are files, usually in the Tractor job scripting format (also known as Alfred scripts). Usually these scripts are written by other applications, such as the "RenderMan for Maya" plug-in or special job submission scripts at each studio. For simple tasks, tractor-spool itself will create a job for you based on a simple command that you want to send to the farm. If you already have a collection of RenderMan RIB files that you want to submit then tractor-spool can also automatically create a job for you from those filenames (see the "–ribs" example).

Usage:

```
tractor-spool [options] [jobfile ...]

tractor-spool [options] -c [/path/]appname appArg1 appArg2 ...

tractor-spool [options] --ribs frm1.rib frm2.rib ...

tractor-spool [options] --rib frm1_prologue.rib frm1.rib...

tractor-spool [options] --jdelete=JOB_ID --user=JOB_OWNER
```

Where "jobfiles" are job scripts describing the work to be done. Several job file names may be given; they will be submitted to the job queue sequentially. The "-c" and "–ribs" variants will create job scripts for you from the command or RIB arguments and submit those.

Note on terminology: Job submission is sometimes referred to as spooling by analogy to print spooling or traditional batch job systems.

Option	Description
--version	Show program's version information, then exit.
-h, --help	Print usage summary, then exit
-v	Add verbose status messages
-q	Quiet mode, print no status
--engine=HOST[:port]	Hostname[:port] of the central Tractor Engine service, default is tractor-engine:80. The hostname "tractor-engine" is usually a DNS alias added by local network administrators, pointing to whichever real hostname is running the service currently. The alias simplifies all connections to the tractor-engine process because everything defaults to trying that name. The default port is 80, and must match the port number on which tractor-engine is actually listening (chosen when starting tractor-engine). If this option is not given, then the spooler will first look for an environment variable named TRACTOR_ENGINE for a host:port specification before using the defaults.

-c
--command
 	
-C
	Send the single command, with the given arguments to the farm (rather than a prewritten job script), specified by collecting all of the remaining command-line parameters, and then creating a single-task Tractor job to send the given command to remote tractor-blade. **NOTE this must be the LAST option given**, since each of the "words" (shell tokens) following the -c will become individual arguments to the given application. For example:
	`tractor-spool --service=required_host_type -c echo hello world`
	
	The search paths and other environment settings used to launch the given command are under the control of the tractor-blade server on each host. There are default paths and settings as well as configurable site-defined "environment packages" -- such as all locked down settings for a given show in production -- that can be selected with other options described here, such as --envkey below.
	
	If the command requires a specific type of remote server, add a --service=name specification before -c. Service name abstractions are defined in blade.config, but you an also just use a hostname if you are targeting a specific host, or a profile name if any host from that class will be acceptable.
	
	There are several built-in "portability aliases" for common commands that are often in used in test or diagnostic situations. These aliases start with an equal-sign to distinguish them from specific actual executables. The most useful of which is probably:
	`tractor-spool --service=somehost -c =printenv`
	
	Use -c for the usual RemoteCmd format, and -C to force a local Cmd.

-r, --ribs	Treats the remaining filename arguments as individual RIB files to be rendered as INDEPENDENT prman processes on different remote tractor-blades; creates a multi-task Tractor job to to handle the renderings
-R, --rib	Treats the remaining filename arguments as RIB files to be rendered using a SINGLE prman process on a remote tractor-blade, that is prman will concatenate all of the rib files for rendering; creates a single-task Tractor job to handle the rendering
--jobcwd=DIRNAME	blades will attempt to chdir to the specified directory when launching commands from this job; default is simply the current directory at time when tractor-spool is run

--projects=PROJECTS	List of project affiliations, like 'TheFilm lighting', can be used with *Limit Allocations* to manage resource pools
--priority=FLOAT	An arbitrary, positive, floating-point priority for the job(s) being spooled; jobs with higher-valued priorities are processed first. Studios usually set conventions for how to use priorities.
--title="words"	Used with -c and -r to change the default job title of the auto-generated job
--svckey=SVCKEY	specifies an additional job-wide service key restriction for Cmds in the spooled job, the key(s) are ANDed with any keys found on the Cmds themselves. When used with -c or --rib option, it overrides "PixarRender" as the sole service key used to select matching blades for those Cmds.
--cmdsvckey=CMDSVCKEY --cmdservice=CMDSVCKEY	used with -c or -r option to specify the sole service key expression for each command; subject to substitution by RANGE and ITER
--remotecleankey=KEY	Converts "local" clean-up Cmds into RemoteCmds. The user may not have control over the the details of job generation in some contexts, so they can't otherwise force clean-ups to be remote. A "local" Cmd is required to run on the same host from which the job was spooled, such as an artist's workstation, meaning that the user would have to keep a tractor-blade process running on their desktop to handle these Cmd clean-ups. In contrast, a RemoteCmd clean-up will run on the first available farm machine. When this option is given, it will also convert Job -whenerror and -whendone blocks into RemoteCmds rather than local Cmds by default. This spooling option also requires a blade service key name to be given, it specifies the appropriate type of blade to run the clean-up. This conversion option is a workaround for cases where the job script generator itself cannot be updated to use RemoteCmd directly in cleanup blocks, and to convert Job -whendone/error blocks into the more general -postscript blocks.
--envkey=ENVKEY	used with -c and -r to change the environment key used to configure the environment variables applied to these auto-generated job scripts; default: None

--range=RANGE	Creates a job with a separate task for each integer in the given range. Ranges are specified like '1-10' or '1-4,7,9,16-20'. Works only with command templates specified with the --command (-c) option. Each new task contains a copy of the command template with the word 'RANGE' replaced by the next integer in the range sequence. Python string format() syntax is also supported, and preferred, using the field name {RANGE}. To insert leading-zero padding like 00041, 00042, use {RANGE:0>NN} where NN is the number of digits. (See also **--ribs** to easily submit a sequence of existing rib files.)
	
	For example:
	
	`tractor-spool --no-spool --range 11-13 -c cp /src/foo.{RANGE:0>5}.rib /dst/bar.RANGE.rib`
	
	displays the following job file:
	
	`Job -title {cp ...} -subtasks {`
	`  Task -title {cp} -service pixarRender -cmds {`
	`  RemoteCmd {cp /src/foo.00011.rib /dst/bar.11.rib}`
	`  }`
	`  Task -title {cp} -cmds {`
	`  RemoteCmd {cp /src/foo.00012.rib /dst/bar.12.rib}`
	`  }`
	`  Task -title {cp} -cmds {`
	`  RemoteCmd {cp /src/foo.00013.rib /dst/bar.13.rib}`
	`  }`
	`}`

--range-formatting=both|sfmt|text	The default setting 'both' attempts substitutions of range integers into -c command arguments using first Python string format() style {RANGE} expressions, followed by simple replacement of any remaining occurrences of the word RANGE. Use the value 'sfmt' here to process only string format {RANGE} expressions, or the value 'text' for simple replacement only.
--itervalues=LIST	create one task for each item in the list, specified by a comma-separated list of items, replacing ITER in the command string or serivce key expression with the current value; e.g. red,green,blue or 'red house,green lawn,blue sky'
--iterfile=FILENAME	create one task for each item in the file, which specifies a line-separated list of items
--task-title=TTITLE	Task title pattern, for use with auto-generated jobs using RANGE or ITER construction

-T [MINSECS,] MAXSEC --runtime-bounds=[MINSECS,] MAXSECS	For use with -c, adds minimum and maximum bounds on the elapsed time of the command on a blade. Give a range of seconds, as 'min, max' or 'min-max'. The launched command is marked Error if its elapsed run time is shorter or longer than the given bounds. Commands that exceed the maximum time are killed. If only one value is given, it specifies the max run time. A max time of 0 (zero) means unbounded.
--limit-tags=TAGS	Specifies the limit counting tags to be added to each command; subject to substitution by RANGE and ITER
--tier=TIER	Dispatching tier assignment, for special-case jobs
--paused	spool the job in a paused state, meaning that no tasks will be launched from it until its priority is later changed from a negative to a positive number.
--aftertime='MM DD HH:MM'	delay job start until the given date, as 'MM DD HH:MM'
--afterjid=JID	delay job start until the given job(s) complete, specified as jid(s)
--nrm	causes auto-generated --ribs to use netrender on the local blade rather than direct rendering with prman on a blade; used when the named RIBfile is not accessible from the remote blades directly
--spool-wait	block until job is fully spooled (to tractor-engine 2.0+)
--status-json	prints the submission confirmation message (or denial) as a JSON-format dict on stdout, rather than plain text
--status-plain	prints spool confirmation message as human-readable plain text; this is the default
-A, --in-alfred	indicates that the job file being submtted is in Alfred (tcl) format, the default
-J, --in-json	indicates that the job file(s) being submitted is formatted as Tractor compliant JSON
--alfescape	process job files assuming that they expect backward-compatible alfred-style two-level unquoting / substitution
--alf-argv-subst	apply second substitution pass on Cmd executable parameters only
--maxactive=MAXACTIVE	limit the maximum number of concurrently active commands of job
--user=LOGIN	The user (login) to be associated with this job; default is the name of the user executing the spooling script. May be disallowed by site configuration policies.
--jdelete=JDEL_ID --jretire=JDEL_ID	delete the requested job from the active queue
--haddr=HADDR	Host address selection for the submitting host, not usually needed. May be necessary only for "local Cmd" variant jobs when the submitting host has several network interfaces or tractor-engine might see a differing addresses for the submission script and for the locally running tractor-blade node.

-p PASSWD --passwd=PASSWD	Tractor login password for the user submitting the job (if engine passwords are enabled)
--configfile=CONFIGFILE	file containing cached login and password data for the user running tractor-spool
--parse-debug	parse the inbound job text and report errors, the job is not submitted to the engine for processing
-o --review --print-alfscript	print the job alfscript rather than spooling it; usually to view or save the job text that tractor-spool itself has generated based on other arguments, rather than when it is reading an existing job from a file.