

tractor-spool

The spooler delivers job scripts into the Tractor job queue for processing and distribution on the farm. Job scripts are accepted as files in alfred job format, and are converted upon receipt by the Tractor engine to an internal job database format (currently JSON files).

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 jobfile(s) 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.

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	print verbose status messages
-q	quiet mode, print no status
-p PASSWD, -- passwd= PASSWD	login password for job owner (if engine passwords enabled)
-o, -- review, -- print- alfscript	print the job alfscrip 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.
-- user=JO BOWNER	alternate job owner, default is user spooling the job
-- configfile =CONFI GFILE	file containing login and password data
-- engine= HOST[: port]	hostname[:port] of the master Tractor daemon, default is tractor-engine - which is usually a DNS alias. The default port is 80, and must match the port number on which tractor-engine is actually listening (chosen when starting the engine). If this option is not given, then the spooler will also look for an environment variable named TRACTOR_ENGINE for a host:port specification before using the defaults.
-c, -C, -- command	<p>execute the command with the given arguments, 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. Each of the "words" (shell tokens) following the -c will become individual arguments to the given application. For example:</p> <pre>tractor-spool -c echo hello world</pre> <p>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.</p> <p>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 can 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.</p> <p>There are several built-in "portability aliases" for common commands that are often 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:</p> <pre>tractor-spool --service=somehost -c =printenv</pre> <p>Use -c for the usual RemoteCmd format, and -C to force a local Cmd.</p>

-T [MINSEC S,] MAXSEC S, -- runtime- bounds= [MINSEC S,] 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.
-- haddr=H ADDR	address of remote spooling client
-- user=LO GIN	the user (login) to be associated with this job; default is the name of the user executing the spooling script
-- 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
-- priority=F LOAT	an arbitrary, positive, floating-point priority for the job(s) being spooled; jobs with higher-valued priorities are processed first
-- projects= PROJEC TS	list of project affiliations, like 'TheFilm lighting'
-- 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=J ID	delay job start until the given job(s) complete, specified as jid(s)
-- 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
-- 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.
--title=" words"	used with -c and -r to change the default title of the auto-generated job
--task- title=TTI TLE	task title
-- iterfile=FI LENAME	create one task for each item in the file, which specifies a line-separated list of items
-- itervalue s=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 service key expression with the current value; e.g. red,green,blue or 'red house,green lawn,blue sky'
--spool- wait	block until job is fully spooled (to tractor-engine 2.0+)
--status- json	causes the diagnostic message regarding the spooling outcome to be formatted as a JSON dictionary, rather than the default plain text message.
--status- plain	prints spool confirmation message as human-readable plain text; this is the default

-r, --ribs	treat the jobfile filename arguments as individual RIB files to be rendered as INDEPENDENT prman processes on different remote tractor-blades; a multi-task Tractor job is automatically created to handle the renderings
-J, --in-json	specified job file is formatted as Tractor compliant JSON
--status-json	prints spool confirmation message (or denial) as a JSON-format dict on stdout
-A, --in-alfred	specified job file is in Alfred (tcl) format, the default
--alfescape	use alfred-compatible two-level unquoting / substitution
--alf-argv-subst	apply second substitution pass on Cmd executable parameters only
-R, --rib	treat the jobfile filename arguments as RIB files to be rendered using a SINGLE prman process on a remote tractor-blade; a single-task Tractor job is automatically created to handle the rendering
--nrm	causes auto-generated --rib jobs 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
--jretire=INTEGER	remove the job indicated by the integer job-id from the active job queue, and terminate any running commands
--range=RANGE	<p>Creates auto-generated jobs in which a template containing the pattern "%d" is expanded to a series of parallel tasks, each referring to a sequential integer in the specified range. For example:</p> <pre>tractor-spool --no-spool --range 1-3 -c cp /src/foo.%d.rib /dst/bar.%d.rib</pre> <p>displays the following job file:</p> <pre>Job -title {cp ...} -subtasks { Task -title {cp} -cmds { RemoteCmd {{cp}} {/src/foo.1.rib} {/dst/bar.1.rib}} -service {pixarRender} } Task -title {cp} -cmds { RemoteCmd {{cp}} {/src/foo.2.rib} {/dst/bar.2.rib}} -service {pixarRender} } Task -title {cp} -cmds { RemoteCmd {{cp}} {/src/foo.3.rib} {/dst/bar.3.rib}} -service {pixarRender} } }</pre> <p>A range can be: (a) a single integer (e.g. 5) (b) two integers separated by a hyphen (e.g. 1-5) (c) a comma-separated list of (a) or (b) (e.g. 1-5,10,15,20-30)</p>
--limit-tags=TAGS	specifies the tags to be added to each command; subject to substitution by RANGE and ITER
--maxactive=MAXACTIVE	limit the maximum number of concurrently active commands of job
--jdelete=JDEL_ID, --jretire=JDEL_ID	delete the requested job from the active queue
--remotecleankey=KEY	Converts "local" clean-up Cmds into RemoteCmds. The user may not have control over 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 -whenever 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.
--parse-debug	parse the inbound job text and report errors, the job is not submitted to the engine for processing

<code>-- cmdsvck ey=CMD SVCKEY , -- cmdservi ce=CMD SVCKEY</code>	used with -c or --rib option to specify the sole service key expression for each command; subject to substitution by RANGE and ITER
<code>-- jobcwd=J OBCWD</code>	blades will attempt to chdir to the specified directory when launching commands from this job; default is simply the cwd at time when tractor-spool is run