Subcommand	Description
`-c, --create`	create templates
`-d, --delete`	delete templates
`-s, --submit`	submit the jobs from templates
`-p, --purge`	purge the existing jobs from templates
`-k, --kill`	kill the existing jobs from templates
`-i, --info`	information about status of the submitted jobs
`-v, --version`	version number of the program
`-l, --license`	credits and license
`-h, --help`	print help

Option	Description
`-w, --worker`	add worker file
`-t, --template`	add template file
`--signal`	add signal for kill
`--debug`	show debugging information
`--config key=value`	assign a key=value pair for configuration settings

Syntax
`-c <PARAMETER_FILE> [-w WORKER_FILE] [-t TEMPLATE_FILE]`
`-d <all|[un]submitted|[un]finished|[un]successful|FROM-TO>`
`-s <all|[un]submitted|[un]finished|[un]successful|FROM-TO>`
`-p <all|[un]finished|[un]successful|FROM-TO>`
`-k <all|[un]finished|[un]successful|FROM-TO> [--signal SIG]`
`-i <history|now|evolution>`

Argument	Description
`PARAMETER_FILE`	file with parameter description
`WORKER_FILE`	executable file to be run remotely
`TEMPLATE_FILE`	append extra variables from template file
`all`	all the templates are subcommanded
`[un]submitted`	only those which were [not] submitted
`[un]finished`	only those whose jobs have [not] finished
`[un]successful`	only those whose jobs finished [un]succesfully
`FROM-TO`	deletes the range [FROM,TO]
`SIG`	kill with signal passed to gwkill
`history`	full historic information about each job
`now`	last status update for each job
`evolution`	timely snapshots of job statuses