

**NAME**

job – Invoke Tcl procedures at regular intervals.

**DESCRIPTION**

The **job** command provides a convenient mechanism to implement Tcl procedures that are invoked at regular intervals. Jobs are activated from the Tcl event loop. Every job can have arbitrary attributes to store job specific state information between two invocations. These attributes help to avoid global variables in order to keep the Tcl name-space clean.

The job scheduler itself relies heavily on the Tcl timer interface which itself depends on the system clock. Moving the system clock backwards can have the effect that jobs are not activated for the amount of time the system clock was moved backwards. Similarly, moving the system clock forward can have the effect that jobs are activated earlier for the amount of time the system clock was moved forward.

**JOB COMMAND**

The **job** command allows to create new jobs and to wait for jobs to complete. The **job** command can also be used to retrieve status information from the job scheduler.

**job create** [*option value ...*]

The **job create** command creates a new job object. The *options* can be used to configure the behaviour of the new job. See the configure command below for the details.

**job current**

The **job current** command returns the name of the currently running job. An empty string is returned if no job is running.

**job info**

The **job info** command returns a list of all job names. The list will be empty if there are no jobs known in this Tcl interpreter.

**job schedule**

The **job schedule** command invoke the scheduler. This is usually done automatically from the event loop. This command may be used during long computations since the scheduler is not pre-emptive. Note, this command can cause arbitrary side effects and should be used with care.

**job wait**

The **job wait** command blocks until all existing jobs have been finished. The command will wait infinitely if there is at least one job left. Events are processed while waiting for the jobs to finish which can have arbitrary side effects.

**JOB INSTANCE COMMAND**

Every job is represented by a job object command which allows to manipulate the job.

**job# attribute** [*name [value]*]

The **job# attribute** command allows to save job specific data in job attributes. If called without any arguments, all existing attribute names for this job will be returned. If called with a name argument, the current value of the attribute will be returned. Non-existing attributes do not produce an error. Instead, an empty string is returned. Calling the option with a name and a value causes the value to be saved in the job attribute.

**job# cget** *option*

The **job# cget** command returns the current value of a configuration option. See the description of supported options below.

**job# configure** [*option value ...*]

The **job# configure** command manipulates configuration options. See the description of supported options below. A list describing all current options is returned if the *options* argument is missing.

**job# destroy**

The **job# destroy** command destroys the job object. This command forces the object to change into the expired state. The scheduler will cleanup this object the next time it is active.

**job# wait**

The **job# wait** command blocks and processes events until the current job changes into the expired state.

## JOB OPTIONS

Every job has associated configuration options which control when a job is activated and which Tcl command is bound to the job.

**-command** *command*

The **-command** option defines the Tcl command that is evaluated whenever the job is activated.

**-interval** *time*

The **-interval** option defines the time interval between two job activations in milliseconds.

**-iterations** *number*

The **-iterations** option defines the total number of times that a job is activated. If this value reaches 0, the job will change its state to expired. Note, if the iterations option is never used, it will also have the value 0 but nothing special happens to the job object.

**-status** *state*

The **-status** option provides access to the current job state. A job is always in one of the states waiting, suspended, running and expired. A job in the expired state will be removed from the system once the scheduler gets activated again. Suspended jobs are not activated by the job scheduler. A suspended job can be resumed by changing its state to running.

**-time** *time*

The **-time** option returns the number of milliseconds until this job is activated the next time. This option is read-only.

## SEE ALSO

scotty(1), Tnm(n), Tcl(n)

## AUTHORS

Juergen Schoenwaelder <schoenw@cs.utwente.nl>