

Re: Need Help in Loop Foreach

Source: <http://coding.derkeiler.com/Archive/Tcl/comp.lang.tcl/2006-09/msg00990.html>

- *From:* "Bezoar" <cwjolly@xxxxxxxxxx>
 - *Date:* 26 Sep 2006 08:02:27 -0700
-

Henr wrote:

Hi, I'm new to Expect/TCL can someone help me out with this simple script.

I'd like to insert a loop, in such a way that it reads all the "System ID" from a file called "SysIDList"

```
# cat SysIDList
8
9
10

#!/usr/local/bin/expect
# Query all the systems status by inputting the SystemIDs on Expect Script

set env(HOME);
log_file $env(HOME)/getsys-out3;
spawn sys_monitor;
expect
"*****";
expect "* SYSMonitor
*";
expect
"*****";
expect " SYSMonitor-> ";
send "QUERY\r";
expect "===== ";
expect " System ID :";
send "8\r"; # This is where the loop would be needed.
expect " SYSMonitor-> ";
send "exit\r";
log_file;
```

I think that by default expect used glob style matching by default so expect "*" would match everything and you would not advance past the initial expect until a timeout occurred. I assume that you want to read one

Re: Need Help in Loop Foreach

line at a time from a file that contains a single number on one line.

For

each one of these lines you want to run a query then return the

information

obtained from the query. I would read in the file first and make a list of

IDs then I would log into the sys_monitor and get to the prompt then loop the QUERY until finished.

returns list that can be arrayable using array set

```
proc getSysIDList { file { time_out 0 } } {
```

```
  global env global spawn_id
```

```
  array set retarray ""
```

```
  set cmd [ auto_execok sys_monitor ]
```

```
  if { [string length $cmd] == 0 } {
```

```
    error "Cannot find executable sys_monitor anywhere in $env(PATH)"
```

```
  }
```

```
  set idlist ""
```

```
  if { [ catch { open $file "r" } fd ] != 0 } {
```

```
    error "Unable to open file $file: $fd"
```

```
  }
```

```
  set fbuff [read $fd]
```

```
  catch { close $fd }
```

```
  set idlist [split $fbuff "\n"]
```

```
  if { [llength $idlist] } {
```

```
    return {};
```

```
  }
```

```
  # ok login to sys_monitor
```

```
  set timeout $time_out; # wait forever if default used
```

```
  log_user 0
```

```
  set pid [ spawn $cmd ]
```

```
  set myid $spawn_id
```

```
  set count 0
```

```
  expect -i $myid -re {.*SYSMonitor->}
```

```
  send_user "Prompt found\n"
```

```
  exp_send -i $myid -- "QUERY\r"
```

```
  set id ""
```

```
  expect {
```

```
    -i $myid
```

```
    -re {.*System ID[ \t]+:} {
```

```
      set id [ lindex $idlist count ];
```

```
      exp_send -i $myid -- "$id\r"
```

```
      exp_continue;
```

```
    }
```

```
    -re {(.*).SYSMonitor->} {
```

```
      # get data
```

```
      set retarray($id) $expect_out(1,string);
```

```
      incr count
```

```
      # go to next id in file/list or exit if at end
```

```
      if { $count >= [ llength $idlist ] } {
```

```
        exp_send -i $myid -- "exit\r"
```

```
      } else {
```

Re: Need Help in Loop Foreach

Re: Need Help in Loop Foreach

```
exp_send -i $myid -- "QUERY\r"
}
# loop around expect
exp_continue;
}
eof {
# use eof detection to tell us when we are done
# may want to break after sending exit in above section if this does
# not work reliably.
if { $count < [llength $idlist ] } {
send_user "Process ended prematurely"
} else {
send_user "Process finished"
}
}
timeout {
send_user "Process timed out with only $count/[llength $idlist ]
queries performed"
}
}
# must reap children or you get zombies
set exitstatus [ exp_wait -i $myid -nowait ]
if { [catch {eval format "pid:%s fd: %s expect exit: %s process
exit:%s" $exitstatus } err ] != 0 } {
send_user "Exit status was $exitstatus"
} else {
send_user "Exit status:$err"
}
catch { exp_close -i $myid }
# if you have any data send it back
if { [ array size retarray ] } {
return [ array gets retarray ]
}
return {};
}

set filename SysIDList
# using default timeout so will wait forever for queries to return
array set systemoutput [ getSysIDList $filename ] ;
parray systemoutput
```

.