

Re: Net::Telnet and SMTP

Source: <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2005-06/msg01376.html>

- *From:* "crr" <pattern@xxxxxxxxxxxxxxxx>
 - *Date:* 21 Jun 2005 13:56:11 -0700
-

Mike wrote:

> Ahhhh so its not being flushed...
>
> Sometimes when dealing with routers and other devices i have to send a
> carriage return after a command in order to get the result of the
> previous command. I believe there is a way to change this behavior.. but
> i have been content with just sending carriage returns..

Hrm, well it seems to be a bit more complex than that. I wrote up a quick test script to see if I could work out somethings without mucking about with my full script. The first test I wrote up worked great, with perfectly interspersed commands and responses, just like I need. I then proceeded to implement these changes into my main script, but I still see the same erroneous behaviour from before where I only see the smtp greeting and the response to the ehlo command.

So I created a second script that includes all the calls to the telnet object in a subroutine (much like my main script) and see the same problems again, but this time I noted that the email isn't going though. Long story short (too late, I know) I realized I was passing the arrays incorrectly.

For whatever reason, it definitely seems to like my reading the files separately into an array and then issuing the commands from that, rather than reading them directly from the file. Could be I introduce some sort of a race between the IO to and from the files and telnet object.

For those interested, I'll include the shorter script that does the trick, which I've backstitched into my main script.

Thanks again for the help.

crr

```
*****Script*****  
use strict;  
use warnings;  
use Net::Telnet;
```

Re: Net::Telnet and SMTP

```
my $server = '10.113.15.66';
my $port = '25';
my $t = localtime(time());
my $logfile = "./logs/smtplog${t}.txt";
my @smtp = ("ehlo test.com", "mail from: bob\@test.com", "rcpt to:
nadmin\@ace66.roke.com", "data", "From: bob", "To: bill", "Subject:
This is a test", "This is a test email!", ".", "quit");
our $stopreading = 0;
my @smtp2 = ("ehlo test.com", "mail from: bob\@test.com", "data",
"quit");

if (-e $logfile) {
die "Log file ${logfile} already exists!";
}
mkdir 'logs', 0744;

open LOGFILE, ">${logfile}" or die "Cannot open log file,
${logfile}";

our $telnet = new Net::Telnet (Timeout => 30,
Telnetmode => 0,
Host => $server,
Port => $port,
Prompt => '//',
Errmode => 'die',
Binmode => 0,
Input_log => *LOGFILE,
Output_log => *LOGFILE);

dotest(\@smtp, \@smtp2);

sub dotest {

my ($smtpcommands, $smtpcommands2) = @_ ;

$telnet->open;
delay(2);
$telnet->get;

foreach my $element (@{$smtpcommands}) {
$telnet->print($element);
delay(2);
#don't want to wait for responses after we read the response to the
#data command, since there ain't none, but we want to start reading
#again once we hit the end of the data section
if ($element eq ".") {$stopreading = 0;}
unless($stopreading) {$telnet->get;}
if ($element eq "data") {$stopreading = 1;}
}#end foreach
```

```
$telnet->open;
delay(2);
$telnet->get;
foreach my $element (@{$smtpcommands2}) {
$telnet->print($element);
delay(2);
if ($element eq ".") {$stopreading = 0;}
unless($stopreading) {$telnet->get;}
if ($element eq "data") {$stopreading = 1;}
}#end foreach
}#end dotest
```

```
#small sub to introduce a delay in the code
sub delay {
my $delaytime = shift;
my $returntime = (time() + $delaytime);
```

```
while (1) {
if (time() >= $returntime) {return;}
}#end while
}#end delay
```

.

• **References:**

- ◆ **[Net::Telnet and SMTP](#)**
 - ◇ From: crr
 - ◆ **[Re: Net::Telnet and SMTP](#)**
 - ◇ From: Mark Clements
 - ◆ **[Re: Net::Telnet and SMTP](#)**
 - ◇ From: Mike
 - ◆ **[Re: Net::Telnet and SMTP](#)**
 - ◇ From: crr
 - ◆ **[Re: Net::Telnet and SMTP](#)**
 - ◇ From: Mike
 - ◆ **[Re: Net::Telnet and SMTP](#)**
 - ◇ From: crr
 - ◆ **[Re: Net::Telnet and SMTP](#)**
 - ◇ From: Mike
-
- Prev by Date: **[Re: Algorithm: spreading out jobs/events in time](#)**
 - Next by Date: **[Re: Email Address Validation](#)**
 - Previous by thread: **[Re: Net::Telnet and SMTP](#)**
 - Next by thread: **[Re: Net::Telnet and SMTP](#)**
 - Index(es):
 - ◆ **[Date](#)**
 - ◆ **[Thread](#)**