What's the Deal with Newline?

```
print CLIENT_SOCK "This is a line.\n";

\015      # on Macintosh operating systems, ASCII 13
\012      # on UNIX/Linux operating systems, ASCII 10
\015\012  # on Microsoft operating systems, ASCII 13+10
\015\012  # with most Internet protocols, ASCII 13+10

print CLIENT_SOCK "This is a line.\012";
```

The Perl Socket API Newline Constants

```
use Socket qw( :DEFAULT :crlf );

    ...

print CLIENT_SOCK "This is a line.$CR";

print CLIENT_SOCK "This is a line.$LR";

print CLIENT_SOCK "This is a line.$CRLF";

    ...

my $line = <TCP_SOCK>;

$line =~ s/$CR?$LF/\n/;

print $line;
```

HTTP Requests and Responses (Commands)

GET
OPTIONS
HEAD
POST
PUT
DELETE
TRACE
CONNECT

The World's Worst Web Browser

```
#! /usr/bin/perl -w

use strict;
use Socket qw( :DEFAULT :crlf );
use IO::Socket;

my $http_method = shift || 'GET';
my $http_server = shift || 'localhost';
my $html_page   = shift || '/index.html';

my $http_obj = IO::Socket::INET->new( PeerAddr => $http_server,
                                      PeerPort => 80,
                                      Proto    => 'tcp' )
    or die "wwwb: could not create socket object: $!\n";

print $http_obj "$http_method $html_page HTTP/1.0$CRLF$CRLF";

while ( my $line = <$http_obj> )
{
    $line =~ s/$CR?$LF/\n/;
    print $line;
}
$http_obj->close;
```

Running the World's Worst Web Browser

```
chmod +x wwwb
./wwwb  | less
./wwwb GET localhost /index.html  | less
./wwwb GET pbmac.itcarlow.ie /index.html  | less
./wwwb POST pbmac.itcarlow.ie /index.html  | less
./wwwb OPTIONS pbmac.itcarlow.ie /index.html  | less
```

Downloading Embedded Graphics

```
./wwwb GET pbmac.itcarlow.ie /networking.gif

    ...


HTTP/1.1 200 OK
Date: Tue, 15 Jan 2002 10:29:58 GMT
Server: Apache/1.3.14 (Unix)  (Red-Hat/Linux) PHP/3.0.17 mod_perl/1.23
Last-Modified: Mon, 14 Jan 2002 14:36:30 GMT
ETag: "51ec6-227d-3c42ecee"
Accept-Ranges: bytes
Content-Length: 8829
Connection: close
Content-Type: image/gif

GIF89^^a$^%#&#*$^*DHSJ&$^#&@(#$#$#% ...

    ...
```

A Persistent `wwwb` - First Effort

```
GET /index.html HTTP/1.0

    ...

GET /index.html HTTP/1.1

    ...

print $http_obj "$http_method $html_page HTTP/1.1$CRLF$CRLF";
```

A Persistent `wwwb` - Second Effort

```
my $http_request;

$http_request =  "$http_method $html_page HTTP/1.1$CRLF" .
                 "Host: $http_server$CRLF" .
                 "$CRLF";

print $http_obj $http_request;
```

The Persistent wwwb - get_resource

```
sub get_resource {
    my $http_obj = shift;
    my %params = @_;

    my $http_method  = $params{METHOD}   || 'GET';
    my $html_page    = $params{RESOURCE} || '/index.html';
    my $connect_type = $params{CTYPE}    || 'Close';
    my $http_server  = $params{SERVER}   || 'localhost';

    my $http_request = "$http_method $html_page HTTP/1.1$CRLF" .
                       "Host: $http_server$CRLF" .
                       "Connection: $connect_type$CRLF" .
                       "$CRLF";

    print $http_obj $http_request;

    while ( $line = <$http_obj> )
    {
        $line =~ s/$CR?$LF/\n/;
        print $line;
    }
}
```

The Persistent wwwb - Invoking get_resource

```
get_resource( $http_obj, CTYPE => 'Keep-Alive' );

    ...

get_resource( $http_obj, RESOURCE => '/apache_pb.gif' );
```

A Better get_resource, 1 of 2

```
sub get_resource {
    my $http_obj = shift;
    my %params = @_;

    my $http_method  = $params{METHOD}   || 'GET';
    my $html_page    = $params{RESOURCE} || '/index.html';
    my $connect_type = $params{CTYPE}    || 'Close';
    my $http_server  = $params{SERVER}   || 'localhost';

    my $http_request = "$http_method $html_page HTTP/1.1$CRLF" .
                       "Host: $http_server$CRLF" .
                       "Connection: $connect_type$CRLF" .
                       "$CRLF";

    print $http_obj $http_request;

    my $body_length = 0;
    my $body_part = "";
    my $line;
    my $in_header = 1;
```

A Better `get_resource`, 2 of 2

```
    while ( $line = <$http_obj> )
    {
        if ( $line =~ /Content-Length: (\d+)/ )
        {
            $body_length = $1;
        }
        if ( $line =~ /^$CRLF$/ )
        {
            $in_header = 0;
        }
        $body_part = $body_part . $line if !$in_header;

        $line =~ s/$CR?$LF/\n/;
        print $line;

        if (!$in_header)
        {
            last if length( $body_part ) >= $body_length;
        }
    }
}
```

HTTP Status Codes

100-199 - Information messages
200-299 - Success messages
300-399 - Redirection messages
400-499 - Client Error messages
500-599 - Server Error messages

HTTP Status Codes - Example Invocations

```
./wwwb GET www.linuxjournal.com   | less

./wwwb DELETE www.ucd.ie  | less

./wwwb GET pbmac.itcarlow.ie /cgi-bin/printenv  | less

./wwwb GET pblinux.itcarlow.ie /cgi-bin/broken  | less
```

The Library for WWW Access in Perl

`libwww-perl` - Large Perl add-on class library.

Provides support for HTTP, HTTPS, GOPHER, FTP, NNTP, FILE and MAILTO protocols.

Key modules/classes:

`HTTP::Request` - for working with requests TO the HTTP server.

`HTTP::Response` - for working with responses FROM the HTTP server.

`LWP::UserAgent` - a mechanism for an application to access `libwwww-perl` functionality.

The LWPwwwb Program

```
#! /usr/bin/perl -w

use strict;
use LWP::UserAgent;

my $http_method = shift || 'GET';
my $http_server = shift || 'localhost';
my $html_page   = shift || '/index.html';
my $http_port   = shift || 80;

my $wwwb_useragent = new LWP::UserAgent;

my $wwwb_url = 'http://' . $http_server . ':'
                         . $http_port . $html_page;

my $wwwb_request = new HTTP::Request $http_method => $wwwb_url;

my $wwwb_response = $wwwb_useragent->request( $wwwb_request );

print $wwwb_response->as_string;
```

Parsing HTML, 1 of 3

```
#! /usr/bin/perl -w

use strict;
use LWP::UserAgent;
use HTML::Parser;

sub print_dtext {
    my ( $parser, $text ) = @_;

    print "text -> ", $text, "\n\n";
}

sub end {
    my ( $parser ) = @_;

    $parser->handler( text => undef );
    $parser->handler( end  => undef );
}
```

Parsing HTML, 2 of 3

```
sub print_link {
    my ( $parser, $tag, $attr ) = @_;

    if ( $tag eq 'a' )
    {
        print "link -> ", $attr->{href}, "\n";

        $parser->handler( text => \&print_dtext, 'self,dtext' );
        $parser->handler( end  => \&end, 'self' );
    }
}
```

Parsing HTML, 3 of 3

```
my $http_method = shift || 'GET';
my $http_server = shift || 'localhost';
my $html_page   = shift || '/index.html';
my $http_port   = shift || 80;

my $wwwb_useragent = new LWP::UserAgent;
my $wwwb_url = 'http://' . $http_server . ':'
                  . $http_port . $html_page;
my $wwwb_request = new HTTP::Request $http_method => $wwwb_url;
my $wwwb_response = $wwwb_useragent->request( $wwwb_request );

my $parser = HTML::Parser->new( api_version => 3 );

print "Parsing $http_server$html_page on port: $http_port:\n\n";

$parser->handler( start => \&print_link, 'self,tagname,attr' );

$parser->parse( $wwwb_response->content );
$parser->eof;

print "\nDone.\n";
```

Some `parsewwwb` Examples

```
./parsewwwb GET www.linuxjournal.com

./parsewwwb GET www.itcarlow.ie

./parsewwwb GET pbmac.itcarlow.ie

./parsewwwb GET pbmac.itcarlow.ie /manual/index.html
```

The Custom Web Server Source Code, 1 of 5

```
#! /usr/bin/perl -w

use strict;

use POSIX ":sys_wait_h";

use HTTP::Daemon;
use HTTP::Status;

use constant HTML_DEFAULT_PAGE => "index.html";

sub zombie_reaper {
    while ( waitpid( -1, WNOHANG ) > 0 )
    { }
    $SIG{CHLD} = \&zombie_reaper;
}
$SIG{CHLD} = \&zombie_reaper;
```

The Custom Web Server Source Code, 2 of 5

```
sub continue_as_child {
    my $http_client = shift;

    while ( my $service = $http_client->get_request )
    {
        my $request = $service->uri->path;

        print $service->method, ": ", $request, " -> ";

        if ( $service->method eq 'GET' )
        {
            my $resource;

            if ( $request eq "/" )
            {
                $resource = HTML_DEFAULT_PAGE;
            }
            else
            {
                $request =~ m{^[./]*(.*)};
                $resource = $1;
            }
```

The Custom Web Server Source Code, 3 of 5

```
            print $resource, " -> ";
            if ( -e $resource )
            {
                $http_client->send_file_response( $resource );
                print "OK.";
            }
            else
            {
                $http_client->send_error( RC_NOT_FOUND );
                print "NOT FOUND.";
            }
        }
        else
        {
            $http_client->send_error( RC_METHOD_NOT_ALLOWED );
            print "NOT OK.";
        }
        print " Remote addr: ", $http_client->peerhost, "\n";
    }
}
```

The Custom Web Server Source Code, 4 of 5

```
my $tcp_port = shift || 8080;

my $httpd = HTTP::Daemon->new( LocalPort => $tcp_port,
                               Reuse     => 1 )
    || die "simplehttpd: could not create HTTP daemon.\n";

print "\nListening for clients at: ", $httpd->url, "\n\n";
```

The Custom Web Server Source Code, 5 of 5

```
while ( my $http_client = $httpd->accept )
{
    my $child_pid = fork;

    if ( $child_pid )
    {
        next;
    }
    elsif ( defined( $child_pid ) )
    {
        continue_as_child( $http_client );
        exit;
    }
    else
    {
        print "simplehttpd: fork failed: $!\n";
    }
}
continue
{
    $http_client->close;
    undef( $http_client );
}
```

The Custom Web Server In Action

```
./simplehttpd

Listening for clients at: http://pbmac.itcarlow.ie:8080/

GET: /index.html -> index.html -> OK. Remote addr: 149.153.100.104
GET: /simplehttpd -> simplehttpd -> OK. Remote addr: 149.153.100.65
GET: /etc/passwd -> etc/passwd -> NOT FOUND. Remote addr: 149.153.1.5
POST: /test.html -> NOT OK. Remote addr: 149.153.100.23
GET: /test.html -> test.html -> OK. Remote addr: 149.153.100.23

    ...

./LWPwwwb GET pblinux.itcarlow.ie /index.html 8080
```

The News Reading Source Code, 1 of 5

```
#! /usr/bin/perl -w

use strict;
use Net::NNTP;

$| = 1;

my $the_server = shift;
my $userid = shift;
my $passwd = shift;

my $server = Net::NNTP->new( $the_server )
    or die "nws: Can't connect to server: $@\n";

$server->authinfo( $userid, $passwd ) if defined( $userid );

my ( $group_to_check, $n, $f, $l, $g );
```

The News Reading Source Code, 2 of 5

```
while ( do {
                print "\nGroup: ";
                $group_to_check = <STDIN>;
                chomp( $group_to_check );
                eval
                {
                    ( $n, $f, $l, $g ) =
                        $server->group( $group_to_check )
                            or die "No group: $group_to_check\n";
                };
                if ( $@ ) { 1; };
            } )
{
    print "\nGroup not found: $group_to_check: try another.";
}

print "\nNumber of articles:\t$n\n";
print "Value of first:\t\t$f\n";
print "Value of last:\t\t$l\n";
```

The News Reading Source Code, 3 of 5

```
print "\nEnter start of range: ";
my $start = <STDIN>;
print "\nEnter end of range: ";
my $end = <STDIN>;

chomp( $start );
chomp( $end );

if ( $start eq '' ) { $start = $f; }
if ( $end eq '' ) { $end = $l; }

print "\nProcessing this range: $end to $start ... \n";

for ( my $i = $end; $i >= $start; --$i )
{
    my $filename = 'art' . $i;

    my ( $header, $bodytext );
```

The News Reading Source Code, 4 of 5

```
eval
{
    $header = $server->head( $i )
        or die "Can't get header from article $i in $g\n";
};
if ( $@ ) { next; }

print "\n";
foreach my $line ( @{$header} )
{
    print $line if $line =~ /^From/;
    print $line if $line =~ /^Date/;
    print $line if $line =~ /^Lines/;
    print $line if $line =~ /^Subject/;
}
print "Get article $i? (y/n/q): ";
my $yorn = <STDIN>;
chomp( $yorn );

if ( $yorn eq 'q' )
{
    last;
}
```

The News Reading Source Code, 5 of 5

```
    if ( $yorn eq 'y' )
    {
        eval
        {
            $bodytext = $server->body( $i )
                or die "Can't get body from $i in $g\n";
        };
        if ( $@ ) { next; }

        open OUTARTFILE, ">$filename" or
            die "Could not open $filename: $@\n";
        print OUTARTFILE @{$bodytext};
        close OUTARTFILE;

        my $cmd = 'less ' . $filename;

        system( $cmd );
    }
}

$server->quit;

print "\nDone.\n";
```

NetDebugging SMTP, 1 of 9

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
149.153.100.8 -> 149.153.100.65 (id: 57624, ttl: 128)

TCP Source: 25 -> TCP Destination: 1128
TCP Header Length: 5, TCP Checksum: 55369
TCP Data:

220 PAT Mercury 1.48 ESMTP server ready.^M
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
149.153.100.65 -> 149.153.100.8 (id: 1871, ttl: 64)

TCP Source: 1128 -> TCP Destination: 25
TCP Header Length: 5, TCP Checksum: 15266
TCP Data:

EHLO itcarlow.ie^M
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

NetDebugging SMTP, 2 of 9

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
149.153.100.8 -> 149.153.100.65 (id: 57880, ttl: 128)

TCP Source: 25 -> TCP Destination: 1128
TCP Header Length: 5, TCP Checksum: 62996
TCP Data:

250-PAT Hello itcarlow.ie; ESMTPs are:^M
250-TIME^M
250 HELP^M
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
149.153.100.65 -> 149.153.100.8 (id: 1873, ttl: 64)

TCP Source: 1128 -> TCP Destination: 25
TCP Header Length: 5, TCP Checksum: 7989
TCP Data:

MAIL FROM:<paul.barry@itcarlow.ie>^M
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

NetDebugging SMTP, 3 of 9

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
149.153.100.8 -> 149.153.100.65 (id: 58392, ttl: 128)

TCP Source: 25 -> TCP Destination: 1128
TCP Header Length: 5, TCP Checksum: 36521
TCP Data:

250 Sender OK - send RCPTs.^M
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
149.153.100.65 -> 149.153.100.8 (id: 1874, ttl: 64)

TCP Source: 1128 -> TCP Destination: 25
TCP Header Length: 5, TCP Checksum: 25389
TCP Data:

RCPT TO:<test.user@itcarlow.ie>^M
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

NetDebugging SMTP, 4 of 9

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
149.153.100.8 -> 149.153.100.65 (id: 58648, ttl: 128)

TCP Source: 25 -> TCP Destination: 1128
TCP Header Length: 5, TCP Checksum: 49474
TCP Data:

250 Recipient OK - send RCPT or DATA.^M
+
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
149.153.100.65 -> 149.153.100.8 (id: 1875, ttl: 64)

TCP Source: 1128 -> TCP Destination: 25
TCP Header Length: 5, TCP Checksum: 30509
TCP Data:

DATA^M
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

NetDebugging SMTP, 5 of 9

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
149.153.100.8 -> 149.153.100.65 (id: 58904, ttl: 128)

TCP Source: 25 -> TCP Destination: 1128
TCP Header Length: 5, TCP Checksum: 63950
TCP Data:

354 OK, send data, end with CRLF.CRLF^M
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

NetDebugging SMTP, 6 of 9

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
149.153.100.65 -> 149.153.100.8 (id: 1877, ttl: 64)

TCP Source: 1128 -> TCP Destination: 25
TCP Header Length: 5, TCP Checksum: 53249
TCP Data:

Message-ID: <3BE7B5A0.AC5DE93B@itcarlow.ie>^M
Date: Tue, 06 Nov 2001 10:04:16 +0000^M
From: Paul Barry <paul.barry@itcarlow.ie>^M
Reply-To: paul.barry@itcarlow.ie^M
Organization: IT Carlow^M
X-Mailer: Mozilla 4.73 [en] (X11; I; Linux 2.2.18-4hpmac ppc)^M
X-Accept-Language: en^M
MIME-Version: 1.0^M
To: test.user@itcarlow.ie^M
Subject: Capturing SMTP Traffic with NetDebug^M
Content-Type: text/plain; charset=us-ascii^M
Content-Transfer-Encoding: 7bit^M
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

NetDebugging SMTP, 7 of 9

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
^M
Hi!^M
^M
Here it is.  A simple test message for NetDebug to capture.^M
This will be used when explaining SMTP in Chapter 4 of^M
Programming the Network with Perl.^M
^M
Paul.^M
--^M
Paul Barry, Lecturer, IT Carlow, Ireland.^M
email: paul.barry@itcarlow.ie^M
.^M
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

NetDebugging SMTP, 8 of 9

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
149.153.100.8 -> 149.153.100.65 (id: 59672, ttl: 128)

TCP Source: 25 -> TCP Destination: 1128
TCP Header Length: 5, TCP Checksum: 4410
TCP Data:

250 Data received OK.^M
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
149.153.100.65 -> 149.153.100.8 (id: 1878, ttl: 64)

TCP Source: 1128 -> TCP Destination: 25
TCP Header Length: 5, TCP Checksum: 29169
TCP Data:

QUIT^M
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

NetDebugging SMTP, 9 of 9

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
149.153.100.8 -> 149.153.100.65 (id: 60184, ttl: 128)

TCP Source: 25 -> TCP Destination: 1128
TCP Header Length: 5, TCP Checksum: 36182
TCP Data:

221 PAT Service closing channel.^M
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

E-mail Enabling `simplehttpd`

```
    else
    {
        $http_client->send_error( RC_NOT_FOUND );
        print "NOT FOUND.";
        eval {
           email_log( $service->method,
                          $resource,
                              $http_client->peerhost );
        };
    }
}
else
{
    $http_client->send_error( RC_METHOD_NOT_ALLOWED );
    print "NOT OK.";
    eval {
        email_log( $service->method,
                       $request,
                           $http_client->peerhost );
    };
}
print " Remote addr: ", $http_client->peerhost, "\n";
```

Creating `simplehttp2d`, 1 of 2

```
use Net::SMTP;

use constant SMTP_SERVER       => 'pat.itcarlow.ie';

use constant SMTP_TO           => 'web.admin@itcarlow.ie';
use constant SMTP_FROM         => 'simplehttp2d@itcarlow.ie';

sub email_log {
    my ( $method, $resource, $peerhost ) = @_;

    my $email = Net::SMTP->new( SMTP_SERVER )
                    or die "Could not create SMTP object.\n";

    $email->mail( SMTP_FROM );
    $email->to( SMTP_TO );
    $email->data;
```

Creating `simplehttp2d`, 2 of 2

```
    $email->datasend( "To: ", SMTP_TO, "\n" );
    $email->datasend( "From: ", SMTP_FROM, "\n" );
    $email->datasend( "Subject: Message from simplehttp2d\n" );
    $email->datasend( "\n" );
    $email->datasend( "The following request could not be " );
    $email->datasend( "satisfied:\n\n" );
    $email->datasend( "    Remote user: $peerhost\n" );
    $email->datasend( "    Method:      $method\n" );
    $email->datasend( "    Resource:    $resource\n" );
    $email->dataend;
    $email->quit;
}
```

Installing `Net::Telnet`

```
gunzip Net-Telnet-3.02.tar.gz
tar xvf Net-Telnet-3.02.tar
cd Net-Telnet-3.02
perl Makefile.PL
make
make test
su
make install
<ctrl-D>
man Net::Telnet
perl -e 'use Net::Telnet'
```

A Net::Telnet Example, 1 of 2

```
#! /usr/bin/perl -w

use strict;
use Net::Telnet ();

sub do_who {
    my ( $telnet_host, $userid, $passwd ) = @_;

    my $telnet_obj = new Net::Telnet ( );

    $telnet_obj->open( $telnet_host );
    $telnet_obj->login( $userid, $passwd );

    my @who_list = $telnet_obj->cmd( "/usr/bin/who" );

    $telnet_obj->close;

    $#who_list = $#who_list - 1;

    return @who_list;
}
```

A Net::Telnet Example, 2 of 2

```
my @list;

@list = do_who( 'linux303', 'barryp', 'passwordhere' );
@list = (@list, do_who( '149.153.103.15',
                     'barryp', 'passwordhere' ));
@list = (@list, do_who( 'glasnost', 'barryp', 'passwordhere' ));

print "@list\n";
```

A Net::Telnet Example - Output Generated

```
root      tty1     Apr 19 16:31
cno2031   pts/0    May 31 16:30 (pc310-10.itcarlow.ie)
cno2020   pts/1    May 31 16:14 (pc3-16.itcarlow.ie)
cno2020   pts/2    May 31 16:27 (pc3-16.itcarlow.ie)
cno2026   pts/3    May 31 16:18 (149.153.131.117)
cno2018   pts/4    May 31 16:30 (pc3-20.itcarlow.ie)
cno2019   pts/5    May 31 16:26 (pc3-21.itcarlow.ie)
COM2059   pts/8    May 31 15:20 (pc3-14.itcarlow.ie)
cno2006   pts/7    May 31 13:30 (pc3-13.itcarlow.ie)
barryp    pts/6    May 31 16:32 (PBMac.itcarlow.ie)
com3027   pts/0    May  9 09:52 (pc2-2.itcarlow.ie)
barryp    pts/1    May 31 16:23 (PBMac.itcarlow.ie)
meudecc   pts/0    May 31 16:30 (staff102.itcarlow.ie)
hickeypm  pts/1    May 31 16:14 (staff23.itcarlow.ie)
kinsella  pts/2    May 31 16:27 (akmac.itcarlow.ie)
whyte     pts/3    May 31 16:18 (149.153.100.117)
barryp    pts/4    May 31 16:31 (PBMac.itcarlow.ie)
```