

# Creating a Dynamic Menu using Perl CGI

by  
Don Franke  
Originally Written in 1999  
Re-written 22 March 2005

## Introduction

Let's say you need to create a dynamic nested menu for a grocery store site. You have three categories of items:

Fruits  
Vegetables  
Breads

For each of these categories you have several items, e.g.:

Fruits  
  Apples  
  Oranges  
  Grapes

It's relatively easy to come up with a way to open/close a menu category. The trick is, how do you have the menu keep the other unclicked categories open, instead of resetting the menu (collapsing all the other open categories)? In other words, how do you have more than one category open at the same time?

A couple caveats:

- This article does not go into setting up Apache, Perl, or CGI. It is assumed you already have this setup and running
- This code uses a little binary arithmetic. A good primer on this is at <http://www.freesoft.org/CIE/Topics/19.htm>. Also, if you want to go more in depth with this subject, an excellent book on this has the same title as this heading: "Ones and Zeros" by John R. Gregg.

## Ones and Zeros

When you click a category link, you pass to itself a binary number. Each digit represents the open/closed state of a category.

$2^2$	$2^1$	$2^0$
Menu Item 2	Menu Item 1	Menu Item 0

This value is passed by the querystring parameter 'n.'

Example:

N=010 means:

Menu Item 2 is closed  
Menu Item 1 is open  
Menu Item 0 is closed

Note: Since this is binary, the menu items are read left to right.

This binary value is then used to determine how the menu is drawn on the page.

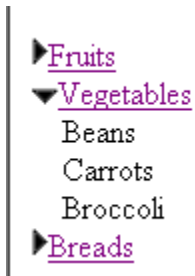


Figure 1: n=010

Now when you click on Breads, the Vegetables category remains open, and the Breads category opens as well.

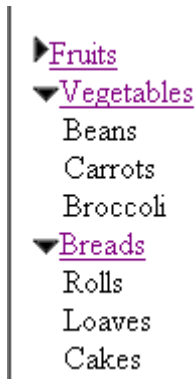


Figure 2: n=011

A lesser menu would have shown only the last-clicked category to be open.

## The Code

I've added the images `arrow_open.gif` and `arrow_closed.gif` to emphasize which categories are open/closed. There is plenty of other dressing up that can be done (e.g. Stylesheets, etc.)

This code uses the function `bin2dec`, which I found at [http://www.unix.org.ua/oreilly/perl/cookbook/ch02\\_05.htm](http://www.unix.org.ua/oreilly/perl/cookbook/ch02_05.htm)

Also, this Perl script was written to work on the following setup:

```
Windows XP
Perl 5.8.2
Apache 2.0.48
```

Note: Modify the first line to work on your particular OS/path as necessary.

```
#!/d:/stuff/perl/bin/perl -w
# filename: menu.pl

use CGI;

my $cgi = CGI->new();

print $cgi->header( -type => 'text/html' );
print $cgi->start_html( -title => 'Menu' );

# grab binary number from querystring
$num = $cgi->param( 'n' );

# convert number to decimal
$numdec = bin2dec($num);

# initialize menu headings
$menu0 = "Fruit";
$menu1 = "Vegetables";
$menu2 = "Breads";

@fruits = ("Apples", "Oranges", "Grapes");
@vegetables = ("Beans", "Carrots", "Broccoli");
@breads = ("Rolls", "Loaves", "Cakes");

# initialize menu boolean settings
$menu0true = "";
$menu1true = "";
$menu2true = "";

# determine menu boolean settings
$menu0true = ($numdec & bin2dec(1));
$menu1true = ($numdec & bin2dec(10));
$menu2true = ($numdec & bin2dec(100));

if($menu0true==bin2dec(1)) {
    $menu0true=1;}
if($menu1true==bin2dec(10)) {
    $menu1true=1;}
if($menu2true==bin2dec(100)) {
    $menu2true=1;}

# function to convert binary to decimal
sub bin2dec {
    return unpack("N", pack("B32", substr("0" x 32 . shift, -32)));
}
```

*Code continued on page 4*

```

# draw out menu
print "\n\n";
if($menu0true) {
    print "<A HREF='menu.pl?n=" . $menu2true . $menultrue . "0'>\n<IMG
SRC='../arrow_open.gif' BORDER='0'>Fruits</A><BR>\n";
    $sending_value = scalar(@fruits) ;
    for($counter=0 ; $counter < $sending_value ; $counter++) {
        print "\t&nbsp; &nbsp; $fruits[$counter] <BR>\n";
    }
} else {
    print "\n<A HREF='menu.pl?n=" . $menu2true . $menultrue . "1'>\n<IMG
SRC='../arrow_closed.gif' BORDER='0'>Fruits</A><BR>\n";
}

if($menultrue) {
    print "<A HREF='menu.pl?n=" . $menu2true . "0" . $menu0true . "'>\n<IMG
SRC='../arrow_open.gif' BORDER='0'>Vegetables</A><BR>\n";
    $sending_value = scalar(@vegetables) ;
    for($counter=0 ; $counter < $sending_value ; $counter++) {
        print "\t&nbsp; &nbsp; $vegetables[$counter] <BR>\n";
    }
} else {
    print "\n<A HREF='menu.pl?n=" . $menu2true . "1" . $menu0true . "'>\n<IMG
SRC='../arrow_closed.gif' BORDER='0'>Vegetables</A><BR>\n";
}

if($menu2true) {
    print "<A HREF='menu.pl?n=0" . $menultrue . $menu0true . "'>\n<IMG
SRC='../arrow_open.gif' BORDER='0'>Breads</A><BR>\n";
    $sending_value = scalar(@breads) ;
    for($counter=0 ; $counter < $sending_value ; $counter++) {
        print "\t&nbsp; &nbsp; $breads[$counter] <BR>\n";
    }
} else {
    print "\n<A HREF='menu.pl?n=1" . $menultrue . $menu0true . "'>\n<IMG
SRC='../arrow_closed.gif' BORDER='0'>Breads</A><BR>\n";
}
print $cgi->end_html;

```

Some things to observe about the code:

- At the top the arrays of items for each category is set up. This chunk of code could be replaced by a database call.
- Each category, when open iterates through that category's item array.

### ***Conclusion***

Hopefully this article has provided to you an interesting way to do dynamic menus in a Perl CGI script, as well as perhaps as some introductory fun with binary arithmetic.

### ***About the Author***

Don Franke is currently a contractor and writer residing in San Antonio, TX. He can be reached at [donfranke@satx.rr.com](mailto:donfranke@satx.rr.com)