

Re: Escape character treatment in string library functions

```
answer[j++] = in[i];
if(in[i] == '\\')
answer[j++] = '\\';
}
answer[j] = 0;

return answer;

}

int main(int argc, char **argv)
{
int i;

for(i=0;i<32;i++)
printf("%s\n", doublebackslashes("My\\Fred"));
return 0;

}
```

I've knocked up a little function for you.
I suspect that what you really need is a "make C escapes" however,
which is
a little more work.

—
Free games and programming
goodies: <http://www.personal.leeds.ac.uk/~bgy1mm>

Thanks everyone for the reply. But the issue I am facing is

I get a string literal in the format "domain\username" (with a single
\ and not \\ ,. say char *str= "dom\ret") .If I try to parse the
string char by char C treats '\r' as a single character.

For eg:

```
char *str="dom\rret";
while( *(str++) !='\0')
if(*str == '\\')
printf("found");
```

is not working since it sees '\r' as a single char .

If the string is being read by an external source, and has the characters
\ and 'r' next to each other, they will be interpreted as two different
characters, '\ and 'r'. If, however, "\r" is typed in the source code, it

Re: Escape character treatment in string library functions

will be treated as a single escaped character. The way to prevent this is in your source code you escape the \ with "\". I.E.

```
char* str = "dom\\ret";
```

will produce '\ ' and 'r' as 2 separate characters. Reading the text from a file with "dom\ret" in it will also produce 2 separate characters. The only real concern you should have is always escape your \ in your source code.

Now, there are some exceptions with external libraries. There are, for example, RE libraries that would treat the string "dom\ret" as the r being escaped. In which case you need to manually escape the \ in the string.

In your example given, this will work:

```
char* str="dom\\ret";
while( *(str++) !='\0')
if(*str == '\\')
printf("found");
```

should produce the output
found

—

Jim Langston
tazmaster@xxxxxxxxxxxxxxxx

.