

Re: How to add thousand separators

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2007-11/msg00102.html>

- *From:* Joseph M. Newcomer <newcomer@xxxxxxxxxxxxx>
 - *Date:* Sun, 04 Nov 2007 14:01:54 -0500
-

First, this code is obsolete as written, because char is a dead data type and should not be used for any purpose whatsoever. You should not be returning char *, and you should DEFINITELY not be using a static variable to hold the result! None of this is required, and it is needlessly complicated, uses obsolete and dangerous functions like sprintf, assumes 8-bit characters, presumes the separates are ,. (in other locales they differ a lot, and let's try to avoid discussing Indic notation which does not use groups of three). This code is not thread-safe.

See my earlier post for a more elegant way to deal with all of this using the NLS APIs, uses Unicode, does not use static variables anywhere, and is written for MFC, not PDP-11 C.

If you need to return values, return a CString or pass in a CString & argument. NEVER use a static buffer.

On Sat, 03 Nov 2007 12:51:10 -0700, przemyslaw.sliwa@xxxxxxxxxx wrote:

Here you go:

```
char* AmountComma(double field, int dec_places /*=0*/)

****
CString AmountComma(double field, int dec_places /*=0*/)
****

    {
        if(_isnan(field))
            return "NaN";

****
return _T("NaN");
****

        else if(!_finite(field))
            return "Infinity";

****
return _T("INF");
```

Re: How to add thousand separators

Note that both of these should be stored as string resources since they might need to be localized!

```
char buf[256];
static char out_buf[256];
```

```
CString buf;
CString out_buf;
****
```

```
if(fabs(field) > 1e16)
{
// probably a bad number. don't make an attempt to show all digits
sprintf(out_buf, "%18e", field);
return out_buf;
}
```

```
out_buf.Format(_T("%18e"));
return out_buf;
****
```

```
}
```

```
char fbuf[16];
sprintf(fbuf, "%18.0f", dec_places);
sprintf(buf, fbuf, field);
```

Assuming it made sense to use obsolete and dangerous library calls like this, you could have written

```
_tsprintf(buf, _T("%18.*f"), dec_places, field);
```

you don't need to construct a format string if the values of the format are themselves variables

```
CString buf;
buf.Format(_T("%.0f"), dec_places, field);
```

Note that no fixed-size buffers are required; it is Unicode-aware, and there is no need to format a formatting string. Also, I see no reason to construct a fixed-length string of 18 digits for any reason.

```
int len = strlen(buf);
assert(len > 0);
len--;
int pos = 255;
out_buf[pos--] = '\0'; // populate out_buf from right to left
```

Re: How to add thousand separators

It is a lot easier to simply construct it left-to-right, and call `_tcsrev` to reverse it when done. Why write such complicated code to accomplish a simple task?

First step: Reverse the string

```
buf.MakeReverse();
```

Second step: copy all digits right to left until the decimal point (whatever it is) is reached.

```
int n = buf.Find(decimalsep);
if(n >= 0)
out_buf = buf.Left(n);
```

Third step: copy all digits right to left, putting the thousand separator in every 3, until a sign is reached

```
buf = buf.Mid(n);
for(int i = 0; ; i++)
{ /* copy */
CString t = buf.Left(1);
buf = buf.Mid(1);
out_buf += t;
if(t == _T("+") || t == _T("-"))
break;
if(buf.IsEmpty())
break;
if(i > 0 && i % 3 == 0)
out_buf += thousandsep;
} /* copy */
```

Fourth step: reverse the string

```
out_buf.MakeReverse();
return out_buf;
```

Note the above code is smaller, Unicode-aware, thread-safe, written in MFC, localizable (except for Indic!), and much easier to understand.

Here are some guidelines to adopt in writing code:

Assume that if you ever write 'char' except in rare and exotic circumstances, you have made a serious coding error. (Note the code shown here is neither rare nor exotic)

Assume that if you ever use an 8-bit literal except in rare and exotic circumstances, you have made a coding error

Assume that if you EVER use a static buffer inside a function, and that declaration is not preceded by the word 'const', you have made a VERY SERIOUS DESIGN ERROR.

Assume that if you ever write a declaration of a fixed-size character buffer, even of TCHAR, except in certain rare and exotic circumstances (usually dealing with raw API calls), you have made a serious design error. And even then you should avoid writing them.

Re: How to add thousand separators

Assume that if you use strcpy, wcsncpy, _tcscopy, strcat, wscat, _tcscat, sprintf, _tsprintf, or wsprintf, you have made an EXTREMELY SERIOUS design error, so bad that you can expect to see headlines on some anti-virus site about your program. Use only the safe forms, such as the strsafe.h functions (VS6 through VS2005) or the _s functions in VS2005 and beyond.

Assume that if you ever write a literal string which depends on knowing how to read the language in which the programmer coded, you have made a serious design error.

```
for(int id = 0; id<dec_places;++id)
{
out_buf[pos--] = buf[len--];
}
// and the ".":
if(dec_places)
out_buf[pos--] = buf[len--];

assert(len > 0);

int have_digits = 0;
for(int i = len; i>=0;i--)
{
char c = buf[i];
if(isdigit(c))
{
//insert a comma every 3 digits
if(have_digits == 3)
{
have_digits = 0;
out_buf[pos--] = ',';
}
have_digits++;
}
out_buf[pos--] = c;

if(isspace(c))
{ // non-white space means we got to far left (don't forget digits
and +/- etc)
break;
}
}

// start of return string is leftmost character populated
return &out_buf[pos+1];
}
```

Re: How to add thousand separators

On Nov 3, 6:45 pm, "Giovanni Dicanio" <giovanni.dica...@xxxxxxxxxx>
wrote:

"Charles Tam" <Charles...@xxxxxxxxxxxxxxxxxxxxxxxxxx> ha scritto nel
messaggio news:0B999E85-655F-4C40-A179-7C75B3EC7008@xxxxxxxxxxxxxxxxxx

I've a CString containing a numeric value. How do I format it
to include
the
thousand separators?

I think you may find the following CodeProject article to be interesting:

"XFormatNumber – A function to format numbers with commas"

<http://www.codeproject.com/string/xformatnumber.asp>

Giovanni

Joseph M. Newcomer [MVP]
email: newcomer@xxxxxxxxxxxx
Web: <http://www.flounder.com>
MVP Tips: http://www.flounder.com/mvp_tips.htm