Subject: Re: 16 bits wchar
Posted by cbpporter on Mon, 04 Aug 2008 22:12:19 GMT
View Forum Message <> Reply to Message

I was trying to finish my methods, but I came to the conclusion that it is far too complicated and I wouldn't be able to maintain it. But then I tried something else. Something a lot simpler.

Add this to CharSet.cpp (or any other package except MakeList, to escape the aggressive link optimizer if method is in same package):

```
WString FromUtf8Op(const char *_s, int len)
{
 if (len >= 8000)
  return FromUtf8(_s, len);

 const byte *s = (const byte *)_s;
 const byte *lim = s + len;
 //int tlen = utf8len(_s, len);
 //WStringBuffer result(tlen);
 wchar buf[33000];
 wchar *t = buf;
 if(len > 4)
  while(s < lim - 4) {
   unsigned code = (byte)*s++;
   if(code < 0x80)
    *t++ = code;
   else
   if(code < 0xC2)
    *t++ = 0xEE00 + code;
   else
   if(code < 0xE0) {
    word c = ((code - 0xC0) << 6) + s[0] - 0x80;
    if(s[0] >= 0x80 && s[0] < 0xc0 && c >= 0x80 && c < 0x800)
     *t++ = c;
    else {
     *t++ = 0xEE00 + code;
     *t++ = 0xEE00 + s[0];
    }
    s += 1;
   }
   else
   if(code < 0xF0) {
    word c = ((code - 0xE0) << 12) + ((s[0] - 0x80) << 6) + s[1] - 0x80;
    if(s[0] >= 0x80 && s[0] < 0xc0 && s[1] >= 0x80 && s[1] < 0xc0 && c >= 0x800
       && !(c >= 0xEE00 && c <= 0xEEFF))
     *t++ = c;
    else {
     *t++ = 0xEE00 + code;
```

```
    *t++ = 0xEE00 + s[0];
    *t++ = 0xEE00 + s[1];
   }
   s += 2;
  }
  else
   *t++ = 0xEE00 + code;
 }
 while(s < lim) {
  word code = (byte)*s++;
  if(code < 0x80)
   *t++ = code;
  else
  if(code < 0xC0)
   *t++ = 0xEE00 + code;
  else
  if(code < 0xE0) {
   if(s > lim - 1) {
    *t++ = 0xEE00 + code;
    break;
   }
   word c = ((code - 0xC0) << 6) + s[0] - 0x80;
   if(s[0] >= 0x80 && s[0] < 0xc0 && c >= 0x80 && c < 0x800)
    *t++ = c;
   else {
    *t++ = 0xEE00 + code;
    *t++ = 0xEE00 + s[0];
   }
   s += 1;
  }
  else
  if(code < 0xF0) {
   if(s > lim - 2) {
    *t++ = 0xEE00 + code;
    while(s < lim)
     *t++ = 0xEE00 + *s++;
    break;
   }
   word c = ((code - 0xE0) << 12) + ((s[0] - 0x80) << 6) + s[1] - 0x80;
   if(s[0] >= 0x80 && s[0] < 0xc0 && s[1] >= 0x80 && s[1] < 0xc0 && c >= 0x800
      && !(c >= 0xEE00 && c <= 0xEEFF))
    *t++ = c;
   else {
    *t++ = 0xEE00 + code;
    *t++ = 0xEE00 + s[0];
    *t++ = 0xEE00 + s[1];
   }
   s += 2;
```

```
  }
  else
   *t++ = 0xEE00 + code;
 }
 *t = 0;
 //ASSERT(t - ~result == tlen);
 return WString(buf, t - buf);
}
```

Then try out this test package to see if there is really a performance improvement. It contains a simple benchmark.

File Attachments

1) MakeList.rar, downloaded 439 times