

From: Frank Farance  
Organization: Farance Inc.  
Telephone: +1 212 486 4700  
Fax: +1 212 759 1605  
E-mail: frank@farance.com  
Date: 1995-12-22  
Document Number: WG14/N533 X3J11/95-134  
Subject: Extended String Literals

Since the 1995-10 WG14 meeting in Nashua, the extended character problem has been divided into two parts: (1) extended identifiers in C programs, (2) extended characters in string literals. The companion paper is WG14/N532. This paper identifies several outstanding issues. This paper is called extended string literals because this is the mechanism that is most visible to the programmer when supporting data with international characters. However, before we can define these extensions, we must resolve the following issues.

#### HOW DO I WRITE A TEXT EDITOR FOR C PROGRAMS

There are three significant applications that characterized the C programming language: (1) the 'hello world' program, (2) a text editor for writing C programs, (3) the UNIX operating system implemented in C. Even if we have identified a syntax for supporting extended identifiers (see WG14/N532, 'Extended Identifiers'), we have no mechanism for handling the characters of the C source code.

ISSUE: What character I/O should we support?

#### WHAT ENCODINGS ARE SUPPORTED

How do we interface to known encodings? For example, let's say we want to read a file that is an ISO 10646-1 UCS-4 stream and "wchar\_t" holds only 16 bits? While it might have been acceptable (but inconvenient) in ASCII-EBCDIC environments to have mapping tables, this becomes error-prone and impractical for most programmers, i.e., the compilation system should provide some support for this. Furthermore, since this is typically how files are specified (their encoding format), isn't this the interface we should be targeting towards?

ISSUE: How do we reconcile the limited capabilities of "wchar\_t" (and the libraries that support this type) with the desire to support character sets that are 32 bits wide?

ISSUE: If we can reconcile this (above), what is the minimum 'international' program we can write that will work on all systems?