Title: extented identifiers in C and C++

Source: Keld Simonsen

Date: 1992-03-16

References: WG14 N203 = WG21 N0113

1. Analyses:

In WG14 N203 a number of issues with having extended identifiers are listed. An analyses of these are given, and a proposal is then given.

1. Use of "dollar sign" etc. in addition to "letters" for identifiers. This can be solved by making a new class, "ident" that specifically lists the various characters allowed in identifiers.

2. only 8-bit not general: this can be solved by making it dependent on the locale, which already can have multibyte capability.

3. The lexical analyses is already locale-dependent, with character string and comments allowed to be multibyte.

4. I do not beleive that Asian characters can be called "alpha". "iswident" for ident class may be approbiate, as this names just indicates the use ofor this set of characters, namely for inclusion in identifiers.

5. there should be a source locale for each source. A proposal would be that every source file would start as being in the C locale, and that you could shift the source locale via a #source <loc> directive. Equivalently you could have a #object <loc> directive to specify the object locale.

6. A solution to vendor defined charsets may be to have a system-wide definition of mapping from different charsets to a unique representation, eg. 10646. For C there already must exist a mapping from external codesets to the widechar representation used.

7. the ISO 2022 system may be good for identifying character sets, but it cannot currently specify the codesets which are of major interest, such as vendor codesets.

8. the linker issue is an implementation issue, and as such not an issue for the standard. Methods as outlined under 6. can also be applied here.

9. The rules for identifiers are different from programming language to programming language. We cannot expect extensions to be uniform for all programming languages, although we can try.

10. The issue is on the table of WG21, and proposals from WGs are certainly appreciated.

Recommendation:

A "ident" class and associated functions, "iswident" or iswmember("ident") or the like is introduced.

Two directives:

```
    #source  <loc>
    #object  <loc>
```

assumed.

External identifiers should be stored in library files in a uniform manner, eg in the widechar format.

Special rules, such as allowing underlines in identifiers should be specail rules of the language, and not part of the "ident" specificaton, like the digits are often specially allowd, but not as the initial character. This will enhance the possibility of having "ident" classes uniform for more programming languages.