Slay Some Earthly Demons XIX: - Remove undefined behavior for non-basic source characters in source files; modified (UB #6 in Annex J.2).

Document: n3866

Author: Ryan Karl

Date: 2025-04-20

Changes: Remove undefined behavior if a character not in the basic source character set is encountered in a source file, excluding valid exceptions.

Undefined Behavior: A character not in the basic source character set is encountered in a source file, except in an identifier, a character constant, a string literal, a header name, a comment, or a preprocessing token that is never converted to a token (J.2 (6), N3301). The editor should remove this from the Annex J.2 table.

Analysis:
The existing 3rd paragraph in section 5.3.1 ends with: “If any other characters are encountered in a source file (except in an identifier, a character constant, a string literal, a header name, a comment, or a preprocessing token that is never converted to a token), the behavior is undefined.” This statement essentially leverages undefined behavior to serve as a catch-all for off nominal scenarios and offers no guidance to implementors. This could lead to unhelpful or inconsistent diagnostics that vary wildly between implementations (e.g. pre-ANSI or legacy toolchains vs. modern implementations).

Recommendation:
The standard should be reworded to denote this as a constraint violation or as implementation defined. This would promote consistency in diagnosing stray characters and align the standard with widespread compiler behavior. See the appendix for examples.

Suggested Rewording I (relative to N3685):

Revise Section 5.3.1, paragraph 3 to say:
…
3. Both the basic source and basic execution character sets shall have the following members: the 26 uppercase letters of the Latin alphabet

A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z

the 26 lowercase letters of the Latin alphabet

a b c d e f g h i j k l m
n o p q r s t u v w x y z

the 10 decimal digit

0 1 2 3 4 5 6 7 8 9

the following 32 graphic characters

! " # % & ' ( ) * + , - . / :
; < = > ? [\ ] ^ _ { | } ~
@ $`

the space character, and control characters representing horizontal tab, vertical tab, and form feed. The representation of each member of the source and execution basic character sets shall fit in a byte. In both the source and execution basic character sets, the value of each character after 0 in the preceding list of decimal digits shall be one greater than the value of the previous. In source files, there shall be some way of indicating the end of each line of text; this document treats such an end-of-line indicator as if it were a single new-line character. In the basic execution character set, there shall be control characters representing alert, backspace, carriage return, and new line. ~~If any other characters are encountered in a source file (except in an identifier, a character constant, a string literal, a header name, a comment, or a preprocessing token that is never converted to a token), the behavior is undefined.~~

Constraints

A source file shall not contain any character other than members of the basic source character set, except in an identifier, a character constant, a string literal, a header name, a comment, or a preprocessing token that is never converted to a token.

…

Suggested Rewording II (relative to N3685):

Revise Section 5.3.1, paragraph 3 to say:

…

3. Both the basic source and basic execution character sets shall have the following members: the 26 uppercase letters of the Latin alphabet

A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z

the 26 lowercase letters of the Latin alphabet

a b c d e f g h i j k l m
n o p q r s t u v w x y z

the 10 decimal digit

0 1 2 3 4 5 6 7 8 9

the following 32 graphic characters

! " # % & ' ( ) * + , - . / :
; < = > ? [\ ] ^ _ { | } ~
@ $`

the space character, and control characters representing horizontal tab, vertical tab, and form feed. The representation of each member of the source and execution basic character sets shall fit in a byte. In both the source and execution basic character sets, the value of each character after 0 in the preceding list of decimal digits shall be one greater than the value of the previous. In source files, there shall be some way of indicating the end of each line of text; this document treats such an end-of-line indicator as if it were a single new-line character. In the basic execution character set, there shall be control characters

representing alert, backspace, carriage return, and new line. If any other characters are encountered in a source file (except in an identifier, a character constant, a string literal, a header name, a comment, or a preprocessing token that is never converted to a token), the behavior is implementation defined.

Recommended practice: implementations should issue a diagnostic that identifies the offending character and its location (and ideally prints the Unicode scalar value when applicable).

Acknowledgments: Thanks to the UB study group, David Svoboda, Dave Banham, and Joseph S. Meyers.

Appendix:

Consider the program below:

```
/* stray.c */

#include <stdio.h>

int main(void) {
    int ©rate = 100;   /* '©' is not in the basic source character set */
    printf("Rate: %d\n", ©rate);
    return 0;
}
```

Compiling this code with older releases of popular compilers demonstrates longstanding practices. For example, compiling on clang 3.5 we observe the following output:

```
<source>:5:9: error: non-ASCII characters are not allowed outside of literals
and identifiers
    int ©rate = 100;   /* '©' is not in the basic source character set */
        ^
<source>:6:26: error: non-ASCII characters are not allowed outside of
literals and identifiers
    printf("Rate: %d\n", ©rate);
                         ^
2 errors generated.
Compiler returned: 1
```

Compiling on TI C6x gcc 12.4.0 we observe the following output:

```
<source>: In function 'main':
<source>:5:9: error: stray '\302' in program
    5 |     int <U+00A9>rate = 100;   /* <U+2018><U+00A9><U+2019> is not in
the basic source character set */
      |         ^~~~~~~
<source>:6:26: error: stray '\302' in program
    6 |     printf("Rate: %d\n", <U+00A9>rate);
      |                          ^~~~~~~
Compiler returned: 1
```

Compiling on icc 16.0.3 we observe the following output:

```
<source>(5): error: unrecognized token
      int ©rate = 100;   /* ‘©’ is not in the basic source character set */
          ^

<source>(5): error: expected an identifier
      int ©rate = 100;   /* ‘©’ is not in the basic source character set */
          ^

<source>(5): error: unrecognized token
      int ©rate = 100;   /* ‘©’ is not in the basic source character set */
           ^

<source>(6): error: unrecognized token
      printf("Rate: %d\n", ©rate);
                           ^

<source>(6): error: expected an expression
      printf("Rate: %d\n", ©rate);
                           ^

<source>(6): error: unrecognized token
      printf("Rate: %d\n", ©rate);
                            ^

compilation aborted for <source> (code 2)
Compiler returned: 2
```

Compiling on msvc 16.1 we observe the following output:

```
example.c
<source>(5): error C3873: '0xa9': this character is not allowed as a first
character of an identifier
<source>(6): error C3873: '0xa9': this character is not allowed as a first
character of an identifier
Compiler returned: 2
```

Compiling on gcc 6.1 we observe the following output:

```
<source>: In function 'main':
<source>:5:9: error: stray '\302' in program
     int ??rate = 100;   /* ‘©’ is not in the basic source character set */
         ^
<source>:5:10: error: stray '\251' in program
     int ??rate = 100;   /* ‘©’ is not in the basic source character set */
          ^
<source>:6:26: error: stray '\302' in program
     printf("Rate: %d\n", ??rate);
                          ^
```

```
<source>:6:27: error: stray '\251' in program
     printf("Rate: %d\n", ▯▯rate);
                          ^
Compiler returned: 1
```