L2/13-015

On noncharacters: a response to L2/13-006

Authors: Tom Bishop, Richard Cook
Date: January 24, 2013

Issue

Statements in the Unicode Standard about noncharacters have received different and incompatible interpretations. The document L2/13-006 proposes to change the wording of TUS to agree with one interpretation.

Proposal

A different, more cautious interpretation should be considered and compared with the interpretation in L2/13-006. Changes to TUS wording about noncharacters should not be made before evaluating the pros and cons of at least these two interpretations. This evaluation should take into account how existing applications actually use or handle noncharacters, with the highest priorities being stability, predictability, and compatibility.

Background

Below are two excerpts from TUS 6.2, Section 16.7:

> "Noncharacters are code points that are permanently reserved in the Unicode Standard for internal use. They are forbidden for use in open interchange of Unicode text data."

> "Applications are free to use any of these noncharacter code points internally but should never attempt to exchange them. If a noncharacter is received in open interchange, an application is not required to interpret it in any way. It is good practice, however, to recognize it as a noncharacter and to take appropriate action, such as replacing it with U+FFFD replacement character, to indicate the problem in the text."

Bold Interpretation

L2/13-006 implies what may be considered a "bold" interpretation. "Internal use" is interpreted very broadly, to include publishing documents containing noncharacters, with the expectation that these documents can be opened and edited by any general-purpose Unicode text editor. The phrases "forbidden for use in open interchange", "never attempt to exchange them", and "problem in the text" are all interpreted to mean effectively nothing, and their deletion is proposed.

TUS 6.2, Section 16.7 says that replacement of noncharacters with U+FFFD is "good practice", but, according to the bold interpretation, text editors that currently follow this practice are making a mistake of "over-rejection of noncharacters".

When a particular interpretation of a rule renders it effectively pointless or worse than pointless, alternative interpretations should be considered.

Cautious Interpretation

Cautious application developers try to avoid even the potential for conflict with TUS or other applications. They need not be too concerned with where exactly the dividing line is between conformance and non-conformance to TUS, or between “internal use” and “open interchange”. When in doubt whether exchange is forbidden, they don’t exchange. When in doubt whether to follow the “good practice” of replacing noncharacters with U+FFFD, they follow it.

By a “cautious interpretation”, we mean an interpretation that is appropriate for cautious developers. One such interpretation is as follows. The phrase “internal use” means “usage within a single program”. (This may be almost what L2/13-006 means by “ephemeral occurrence during program execution”, except that the duration of usage may be unlimited, and may extend to any number of executions, as long as it remains internal.) For example, noncharacters may be used as arguments and return values of functions, and for storage in variables in RAM or in temporary/private/non-public-readable text files or proprietary binary files.

The phrases “forbidden for use in open interchange” and “never attempt to exchange them” are to be taken as meaningful, and very seriously. Noncharacters are never to be stored in a text file or database or clipboard that is meant to be accessible by other programs. Noncharacters are not to be exchanged with any library, service, or API, unless it is explicitly documented to support such usage. “Good practice” means, as TUS recommends, that a program should treat any noncharacter received in open interchange as a “problem in the text” and replace it with U+FFFD.

While noncharacters can be convenient for internal use, there does not appear to be any software feature that would be impossible, or even much more difficult, to implement without exchanging noncharacters. Therefore, it is feasible for any application to follow a cautious interpretation of TUS.

Compatibility, Stability, and Predictability

If all applications followed cautious interpretations, noncharacters would not be problematic. If all applications followed bold interpretations, it is hard to imagine that the frequent interchange of noncharacters would not lead to incompatibilities. Currently, some applications may follow cautious interpretations while others may follow bold ones. Before reaching any conclusion about revising TUS, there should be an attempt to determine roughly how many current applications fall into each of these two categories. That determination should be followed by a careful evaluation of the feasibility of resolving the incompatibilities in various ways.

For the sake of stability, the need to revise or have an impact on a large (or unknowable) number of applications should be avoided if possible. Unpredictable results of software interactions may have security repercussions. Therefore a high priority for deciding on a course of action should be to take into account whether it makes software interactions more or less predictable. As an alternative to changes in widely held properties and behaviors of currently encoded noncharacters, it may be preferable to assign new code points with the desired properties, if such properties can be identified.