UTC 98/002
1998-November-30
1999-January-28 (Revised)

Subject: Additional Requirement for Unicode 3.0 Conformance

From: Edwin Hart

The conformance requirements in chapter 3 of *The Unicode Standard, Version 2.0* book give an excellent explanation of requirements for implementing a complete Unicode 2.0 implementation. Unfortunately, the reality is partial rather than full implementations of Unicode 2.0/2.1. Consequently, I am asking the UTC to refine the Unicode 3.0 conformance clause with more-details to reflect both full *and* partial implementations of Unicode 3.0.

Requirement

Purchasers who need products that support Unicode need more detailed information about how a prospective product actually supports Unicode. What Unicode features are included in the product, and what features are not? The UTC needs to strengthen the Unicode 3.0 and future conformance requirements to provide the necessary information

Problem

When customers purchase products that claim to support Unicode, without evaluating it, they have no idea what such support entails and whether one product will interoperate with another one.

Vendors are delivering products that claim to support Unicode but that do not support the full set of features specified in *The Unicode Standard, Version 2.0* book (the "Unicode book"). In chapter 3, the Unicode book lists requirements for conformance for a full implementation of Unicode 2.0. This information is extremely important to developers. However, implementing Unicode is a complex, time-consuming effort. Consequently, organizations are implementing Unicode in stages and delivering products without *all* of the features of Unicode 2.0. As of November, 1998, the author is unaware of any shipping commercial product that fully supports Unicode 2.0. Although customers welcome incomplete Unicode support over no support, the Unicode 2.0 compliance statement provides purchasers with no information about assessing what features of Unicode 2.0 are actually implemented in a product.

Justification

Until full conformance testing by an independent authority is available, customers need the vendors to provide a minimal level of conformance information to help make purchase decisions. Unicode support is a complex subject. Vendors and customers need a clear way to communicate what features a product supports and what features a product does not support. Customers realize that the quality of support for a product that claims conformance for feature "*X*" may vary from product to product. However, customers will know that a product that makes no claims about conformance for feature "*X*" will very likely have very poor or no support for this feature. More detailed conformance assessment information will help customers eliminate products from more detailed (and more expensive) evaluation. It would also help developers identify what is missing to help schedule future development. In addition, the UTC would establish better relations by working with customers to implement this requirement. Among others with an interest in this are: Jennifer DeCamp (MITRE), Everette Jordan (DCI Foreign Language Committee), and James Agenbroad (Library of Congress).

Suggested Solution

Include a new conformance statement with a checklist similar to the following. The intent of the list is to provide ideas of what is required but not necessarily the complete list.

> A product that claims conformance to the Unicode 3.0 standard shall state which features of Unicode 3.0 are supported in the product. By implications, a feature not included in the statement shall be considered to have no support in the product.
>
> Products claiming conformance to Unicode shall include the following information:
>
> - The version of Unicode supported.
> - Unicode 3.0 compliance with the section corresponding to 3.1 of the Unicode 2.0 book.
> - Repertoire supported (character ranges and individual characters).
> - Which of the character properties are supported for the supported character ranges and individual characters? Are combining characters supported (which combining characters with which scripts)? Are equivalences supported? Is the bi-directional (bi di) algorithm supported? Is case conversion supported? Is the character-mirroring property supported? A table would, perhaps, simplify the communication of this information.
> - Types of processing supported.
> - Conventions for sorting.
> - Types of input supported (keyboard, hexadecimal, text strings, files, voice (audio), etc.).
> - Types of output supported (text strings, files, rendering onto screens and paper).
> - Forms of Unicode supported for input and output (UCS-2, UTF-16, UTF-8, UCS-4).
> - Conversions of coded character sets supported for input and output. Which other coded character sets are supported? Can customers modify the delivered conversions? Are customer-defined conversions supported? How are undefined characters converted on input and on output?
> - List of dependencies to obtain support for the above.
> - Predefined Usage of private use space (e.g., predefined allocations for logos, additional CJKV characters, etc.)
>
> The following information is optional:
>
> - List of known exceptions.
> - List of other products with which the product in question has demonstrated interoperability.