US Contribution to the CAW

1. Introduction.

This document is a contribution to the JTC1 Cultural Adaptability Workshop (CAW) that is being organized for January 1998. The scope of the JTC1 CAW is to develop a program of work, including development of policies and definition of a process, to ensure availability of products/solutions that take into account cultural adaptability considerations. Several SCs and working groups, within JTC1, work on a variety of standards that provide pieces of what is partially needed but the efforts are fragmented and not very well integrated. In addition, other standardization efforts in IETF, CEN, W3C and other SDOs need to be taken into account.

The scope and program of work for the JTC1 entity recommended by the CAW is to achieve and implement global interoperability. This paper defines what global interoperability is, and identifies various aspects such as attributes, impacted tools and services, key players, what do we have today, what is needed now and a conclusion. It is not a paper with a lot of details but provides a basic look at where we are and what needs to be done to make such a vision come true.

2.0 Objective of recommendation to CAW.

Provide a localized or localizable (enabled/internationalized) product/solution to the end customer that can operate locally and globally to allow file and/or data transparency.

3.0 Definition of Global Interoperability.

Write-once run everywhere. There is only one source-code and one binary (object-code) with user installable locales. Create and publish once, read and process everywhere.

4.0 Attributes of Global Interoperability.

1. Character sets
2. Character properties
3. Global networking & Internet protocols
4. Locales for cultural conventions and a repository to use them
5. Manuals or subsets thereof in local language
6. Keyboards
7. APIs

5. Impacted Tools & Services for Global Interoperability.

1. Operating System Platforms including input methods and Display/Printer drivers
2. Programming Languages
3. Application software - platform independent if possible
4. Browsers
5. Tools that aid translation of user manuals
6. User interface: messages, icons, user feedback/responses

6. Key Players.

1. Industry Implementers
2. National standards bodies
3. Relevant JTC1 SCs
4. Liaison organizations to JTC1 SCs

5. IETF
6. W3C
7. Others

7. What do we have today?

1. Fragmented set of standards and protocols that leave a lot of room for individual interpretation that can lead to incompatible and non-interoperable implementations.
2. Operating system platforms that have many source-code and an equivalent, or possibly more, object-code (binary) versions that support a variety of languages/scripts and cultural conventions but are out of sync with each other and cannot be easily maintained.
3. There is little or no file or data transparency. A file, containing ASCII text, that is created using one national version, such as Japanese, cannot be used by a different national version, such as the U.S. one.
4. What exists today is a small group of various experts from a few JTC1 subcommittees working loosely together on an infrequent basis without solid industry participation.

8. What is needed now?

1. Reconfirmation and/or definition of the few fundamental building blocks that provide a structure for high-level solution for what the end user needs.
2. Identify and define the necessary protocols and standards to weave a comprehensive set that provides the implementers with the necessary tools to develop globally interoperable products/solutions.
3. Define data/file formats that allow transparency and ease of portability among the various platforms and/or application software packages in any localized or localizable solution.
4. Platforms and/or applications must be developed that are globally interoperable and portable (one source-code and one object-code/binary). This eliminates the synchronization problems of having various national versions that are released separately after release of the U.S. version.

9. Conclusion.

A platform implementation can be written to allow user installable locales by script or country/region of the world. A piece of software can be written to allow a user, for example in Japan, to create a bilingual presentation in Japanese and English, Japanese only, or English only. The user does not have to transform the presentation from a Japanese localized system to an English localized system. The system would work using a user installable locale including the appropriate fonts, desired formats and relevant cultural conventions.

Such a solution yields higher productivity at reduced cost in areas such as development, maintenance, usability and training.