ISO/IEC JTC1/SC2/WG2 N2873
2005-1-19

L2/05-013

Universal Multiple Octet Coded Character Set
International Organization for Standardization
Organisation internationale de normalisation
Международная организация по стандартизации

Doc Type:	**Working Group Document**
Title:	**Unicode Liaison Report for WG Meeting #46 in Xiamen**
Source:	**Asmus Freytag**
Status:	**Expert contribution**
Action:	**For consideration by JTC1/SC2/WG2**
Related:	**N2895 and others**

For review at WG2 #46 meeting in Xiamen, China

Character encoding proposals

The Unicode Consortium has received and reviewed many character encoding proposals since the last WG2 meeting in Markham, Ont. Character proposals approved by the UTC have been submitted under separate cover.

Ballot comments

The Unicode Consortium supports the FPDAM1 and PDAM2 Ballot comments submitted by the US National Member Body, whose experts meet in joint session with the Unicode Technical Committee.

For Information

Upcoming Versions of the Unicode Standard

The next minor version of the Unicode Standard will be Unicode 4.1. This version is currently in beta review. Unicode 4.1 will be synchronized with Amd1 to ISO/IEC 10646:2003. It is intended to be an online-only minor version of the Unicode Standard. Unicode 4.1 is to be released in Q2 2005.

Unicode 5.0 will again be a full, book plus online version of the Unicode Standard. Tentatively it is planned to be synchronized with Amd2 to ISO/IEC 10646:2003. The release date for Unicode 5.0 is still tentative.

Registration of Ideographic Variation Sequences

The UTC approved a project of developing of a Unicode Technical Standard that will establish a registry of variation sequences for Ideographic characters. Such variation sequences can be used to reference specific variants of ideographs. The UTC recognizes that the needs of various user communities for such variation sequences cannot be accommodated by a single, unified collection of sequences. The purpose of the registry is to ensure that multiple collections can coexist without compromising the interchangeability of texts using them. Input on the development of this standard is welcome.

Common Locale Data Repository, Version 1.2 released

New versions of the Common Locale Data Repository (CLDR 1.2) and the Locale Data Markup Language specification (LDML 1.2), were released November 4, 2004. They provide key building blocks for software to support the world's languages. The release contains data for 232 locales, covering 72 languages and 108 territories. There are also 63 draft locales in the process of being developed, covering an additional 27 languages and 28 territories.

To support users in different languages, programs must not only use translated text, but must also be adapted to local conventions. These conventions differ by language or region and include the formatting of numbers, dates, times, and currency values, as well as support for differences in measurement units or text sorting order. Most operating systems and many application programs currently maintain their own repositories of locale data to support these conventions. But such data are often incomplete, idiosyncratic, or gratuitously different from program to program. In the age of the internet, software components must work together seamlessly, without the problems caused by these discrepancies.

The CLDR project provides a general XML format, LDML, for the exchange of locale information used in application and system software development, combined with a public repository for a common set of locale data in that format. In this release, there are major additions to the CLDR data, to the LDML specification, and in implementation support.

The CLDR is continually being enhanced and extended, with CLDR 1.3 expected early in 2005. All new data or defect reports for CLDR 1.3 must be submitted no later than January 15, 2005.

For more information about the CLDR project, with details about the new features in this release and the languages and territories supported, see http://www.unicode.org/cldr/.

Identifier Syntax and Character Foldings

When parsing for identifiers in programming languages and similar environment two tasks need to be performed. The first identifies the characters that may occur in an identifier, the other treats certain characters as equivalent for the purpose of identifier matching. An example of the latter is case folding which is used in case insensitive identifier matching.

UAX#31 *Identifier and Pattern Syntax* will be a new Unicode Standard Annex and will first be released as part of Unicode 4.1. It describes how to parse for identifier in programming languages and other environments, based on character classifications in the Unicode Character Database. As a UAX it is automatically maintained synchronously with any additions to the standard. The characters covered are a superset of those for which ISO/IEC TR10176 provides classification. In this context please note document N2895 submitted by V.S. Umamaheswaran which contains a draft version of this document.

UTS #30 *Character Foldings* has been approved and is pending publication. This standard provides an overview of common folding operations including those used in some types of identifier parsing.