

Unicode Support in FME 2007

Like many developers around the world, Safe found the prospect of deciphering the mysteries of character encoding systems a little overwhelming. But with FME 2007, we faced up to the challenge: FME is now a lot more savvy about handling Unicode encoded text strings in several "Unicode aware" formats. Users who need to combine source data in multiple non-native languages will find FME 2007 a substantial improvement.

As a result of this work, we've come to appreciate Unicode's elegant and unambiguous schemes for encoding characters. Text files encoded in UTF-8, UTF-16, or UTF-32 have a Byte Order Marker (or BOM) at the beginning of the file - a two-byte instruction that indicates that the file is in Unicode and whether the bytes are sequenced in big endian or little endian order. To borrow a phrase from an old Peter Sellers movie, you could say, we've "learned to stop worrying [about international character sets] and love the BOM!"

FME 2007 not only recognizes the Unicode character encoding schemes included in source data for a number of common text file formats, but also preserves this encoding information for each attribute string throughout any transformations that may be applied to the data during processing.

Supported Unicode-aware Formats in FME 2007

- Text File
- CSV
- dBASE III
- ESRI Shape
- ESRI ArcSDE
- ESRI ArcSDE Raster
- ESRI Geodatabase (Personal, File, & Enterprise)
- GML
- KML
- XML
- GeoRSS
- Microsoft Access
- Microsoft Excel
- Microsoft SQL
- Oracle 10g/9i/8i

FME's XML writer has always performed well with respect to representing international character sets, but in FME 2007, users will notice considerably improved results when writing out data to KML. Spatial data visualized in Google Earth will now display annotations correctly in multiple languages simultaneously, including Chinese, Japanese and Korean.

FME 2007 supports the three main Unicode schemes for encoding character representations: UTF-8, UTF-16 and UTF-32. KML, XML, and GML files are encoded in either UTF-8 or UTF-16. For UTF-16, the specific byte order for reading in the file - as indicated by the Byte Order Marker (BOM) - is automatically detected by FME.

Shape files have a fixed list of encoding schemes available and the scheme used is always indicated in the file; FME will automatically detect this information. Text File and CSV files, however, may not always include information on the encoding scheme in the file. When present, it is specified by the BOM in the file header and is automatically detected. If a BOM is not included in the file, the user must specify the Unicode character encoding scheme as a source data parameter. Source data in non-Unicode character encodings can be written to a Unicode encoding in any of the supported formats.

