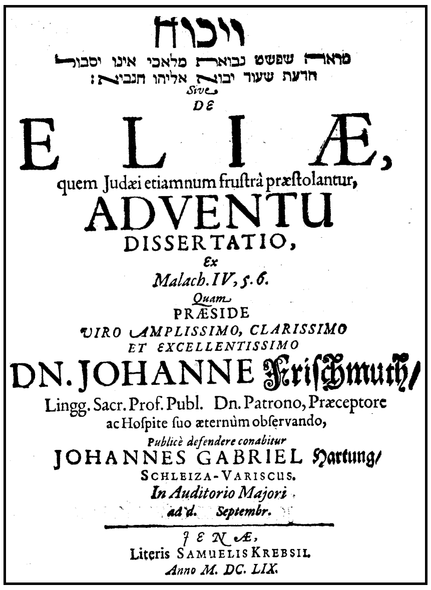
Language-specific encoding in multilingual corpora: Requirements and solutions

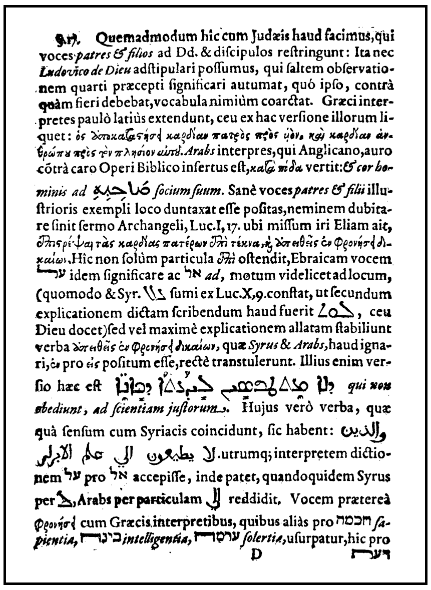
Jost GIPPERT

Only a century after Johannes Gutenberg had invented the art of employing movable types in printing, this new method of publishing had developed in such a way that it became possible to use it for all kinds of multilingual documents, the most striking examples being the so-called "polyglot" Bible editions where modern versions of the Holy Scriptures were arranged synoptically with their Latin, Greek, Hebrew, or Syriac ancestors, all printed in their original scripts (cf., e.g., HUTTER 1599-1643). And by 1650, it had become quite common to adopt this standard of printing to what we might call routine products such as a young theologist's fourty-page doctoral thesis; cp. the specimen taken from HARTUNG (1659) which shows the beautifully mixed arrangement of Latin, Greek, Hebrew, Syriac, and Arabic possible at that time.



ויכוח
מראה שפשט נבואת מלאכי אינו יסבול
הדעת שעוד יבוא אליהו הנביא:
Sive
DE
E L I Æ,
quem Judæi etiamnum fruſtrà præſtolantur,
ADVENTU
DISSERTATIO,
Ex
Malach. IV, ſ. 6.
Quam
PRÆSIDE
VIRO AMPLISSIMO, CLARISSIMO
ET EXCELLENTISSIMO
DN. JOHANNE Friſchmuth/
Lingg. Sacr. Prof. Publ. Dn. Patrono, Præceptore
ac Hoſpite ſuo æternùm obſervando,
Publicè defendere conabitur
JOHANNES GABRIEL Hartung/
SCHLEIZA-VARISCUS.
In Auditorio Majori
ad d. Septembr.
JENÆ,
Literis SAMUELIS KREBSII.
Anno M. DC. LIX.

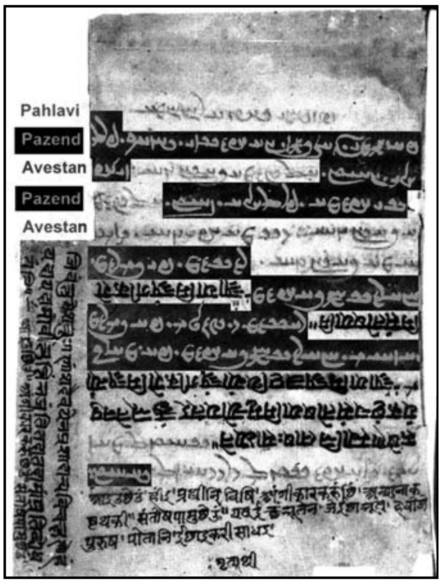
Figure 1: HARTUNG (1659), title page

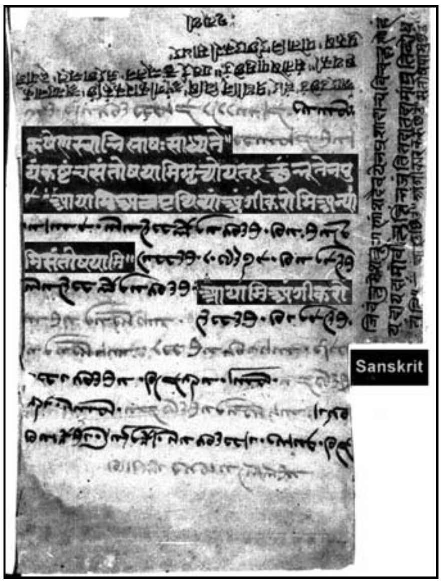


§. 17. Quemadmodum hic cum Judæis haud facimus, qui voces *patres & filios* ad Dd. & diſcipulos reſtringunt: Ita nec *Ludovico de Dieu* adſtipulari poſſumus, qui ſaltem obſervationem quarti præcepti ſignificari autumat, quó ipſo, contrà quàm fieri debebat, vocabula nimiùm coarctat. Græci interpretes paulò latiùs extendunt, ceu ex hac verſione illorum liquet: *ὃς ἀποκαταστήσει καρδίαν πατρὸς πρὸς υἱὸν, καὶ καρδίαν ἀνθρώπου πρὸς τὸν πλησίον αὐτοῦ.* *Arabs* interpres, qui Anglicano, auro cõtrà caro Operi Biblico inſertus eſt, *κατὰ πόδα* vertit: *& cor hominis ad* صاحبه *ſocium ſuum.* Sanè voces *patres & filii* illuſtrioris exempli loco duntaxat eſſe poſitas, neminem dubitare ſinit ſermo Archangeli, Luc. I, 17. ubi miſſum iri Eliam ait, *ἐπιστρέψαι τὰς καρδίας πατέρων ἐπὶ τέκνα, καὶ ἀπειθεῖς ἐν φρονήσει δικαίων.* Hîc non ſolùm particula *ἐπὶ* oſtendit, Ebraicam vocem על idem ſignificare ac אל *ad,* motum videlicet ad locum, (quomodo & Syr. ܠܘܬ ſumi ex Luc. X, 9. conſtat, ut ſecundum explicationem dictam ſcribendum haud fuerit ܠܘܬ, ceu Dieu docet) ſed vel maximè explicationem allatam ſtabiliunt verba *ἀπειθεῖς ἐν φρονήσει δικαίων,* quæ *Syrus & Arabs,* haud ignari, *ἐν* pro *εἰς* poſitum eſſe, rectè tranſtulerunt. Illius enim verſio hæc eſt ܠܐ ܡܬܛܦܝܣܢܐ ܠܝܕܥܬܐ ܕܟܐܢܐ *qui non obediunt, ad ſcientiam juſtorum.* Hujus verò verba, quæ quà ſenſum cum Syriacis coincidunt, ſic habent: والذين لا يطيعون الي علم الابرار. utrumq; interpretem dictionem על pro אל accepiſſe, inde patet, quandoquidem Syrus per ܠ, Arabs per particulam الي reddidit. Vocem præterea *φρονήσει* cum Græcis interpretibus, quibus aliàs pro חכמה *ſapientia,* בינה *intelligentia,* ערמה *ſolertia,* uſurpatur, hic pro
D דעת

Figure 2: HARTUNG (1659), p. [23]

Today, more than 500 years after Gutenberg's invention, we are still striving to attain a comparable standard when working with digital equipments, the most tackling problem consisting in the mixed processing of different script directions. This is by no means a far-fetched task to be mastered by a fringe group of ivory-towered outsiders only, as some people might think, because the production of multilingual texts plays a steadily increasing rôle in the transfer of data via the World Wide Web, and different script directions are involved wherever European and Near Eastern languages have to be arranged side by side. And the problem is in fact not even a modern one: It had to be coped with, e.g., by the writers of Zoroastrian manuscripts such as the one reproduced in figures 3 and 4[1]. Apart from its basic part, the Avestan (Old Iranian) text called *Aogəmadaēcā*, this manuscript contains the Middle Persian version of the same text written in two different scripts, its translation into Sanskrit and a translation into Old Gujarātī, added secondarily *in margine*. Of the scripts involved, only the





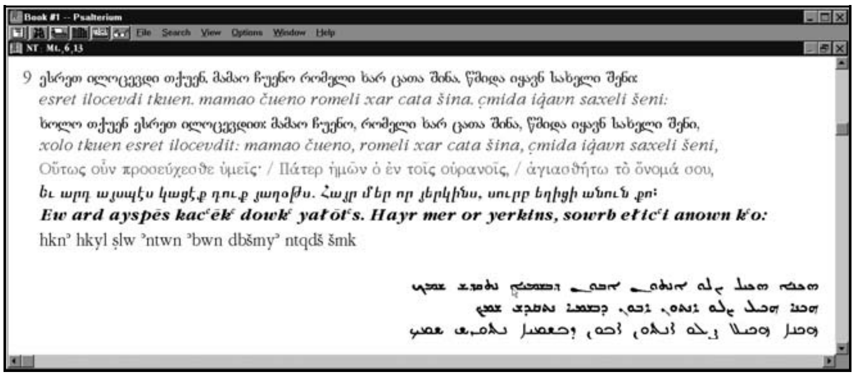
Figures 3-4: Avestan, Pahlavī, Pāzend and Sanskrit versions of *Aogəmadaēcā*

[1] Ms. K 42 of the Copenhagen Royal Library. Its first page reproduced here is taken from the facsimile edition printed in JAMASPASA (1982).

one used for Sanskrit and Gujarātī (the so-called “Devanāgarī”) is directed left-to-right while both the Avestan script (used for Avestan and Middle Persian in its so-called “Pāzend” style) and the Middle Persian script proper (so-called “Pahlavī”) are written from right to left. As the figures show, the solution applied by the writer is a simple one: Arriving at a position where he had to switch from Avestan to Sanskrit, he turned the paper by 180°, then continuing with Devanāgarī in its usual direction, but with lines arranged upwards; at the end of the Sanskrit passage, he turned the paper again in order to proceed with one of the other scripts, etc. etc.

It is not the question of processing and printing mixed scripts, however, that I intend to discuss in the following pages. Several software solutions have been developed for this problem in recent years (cp., e.g., figure 5 showing the “WordCruncher” screen output of Mt. 6,9, the beginning of *Our Father*, in a mixed arrangement of Georgian, Greek, Armenian, and Syriac Bible versions, both transliterated and in original scripts), and it is to be hoped (though by no means certain, cf. below) that after the adaptation of operating systems to real 16-bit encoding on the basis of Unicode, it will become more and more neglectable.

In the present paper, I shall deal with a different problem instead which is related to but not necessarily identical with the one discussed above, viz. the problem of preparing multilingual corpora for a language-specific retrieval. The synoptical arrangement of several Bible versions in one printed edition or in one electronic text file is indeed a multilingual corpus in its own right, just as the Zoroastrian manuscript we have seen



9 ესრეთ ილოცევდი თქუენ. მამაო ჩუენო რომელი ხარ ცათა შინა. წმიდა იყავნ სახელი შენი:
esret ilocevdi tkuen. mamao čueno romeli xar cata šina. çmida iqavn saxeli šeni:
ხოლო თქუენ ესრეთ ილოცევდით: მამაო ჩუენო, რომელი ხარ ცათა შინა, წმიდა იყავნ სახელი შენი,
xolo tkuen esret ilocevdit: mamao čueno, romeli xar cata šina, çmida iqavn saxeli šeni,
Οὕτως οὖν προσεύχεσθε ὑμεῖς· / Πάτερ ἡμῶν ὁ ἐν τοῖς οὐρανοῖς, / ἁγιασθήτω τὸ ὄνομά σου,
Եւ արդ այսպէս կացէք դուք յաղօթս. Հայր մեր որ յերկինս, սուրբ եղիցի անուն քո:
Ew ard ayspēs kacʿēkʿ dowkʿ yałōtʿs. Hayr mer or yerkins, sowrb ełicʿi anown kʿo:
hknʾ hkyl ṣlw ʾntwn ʾbwn dbšmyʾ ntqdš šmk

ܗܟܢܐ ܗܟܝܠ ܨܠܘ ܐܢܬܘܢ ܐܒܘܢ ܕܒܫܡܝܐ ܢܬܩܕܫ ܫܡܟ
ܗܟܢܐ ܗܟܝܠ ܨܠܘ ܐܢܬܘܢ ܐܒܘܢ ܕܒܫܡܝܐ ܢܬܩܕܫ ܫܡܟ
ܗܟܢܐ ܗܟܝܠ ܨܠܘ ܐܢܬܘܢ ܐܒܘܢ ܕܒܫܡܝܐ ܢܬܩܕܫ ܫܡܟ

Figure 5: Mt. 6,9 in several versions (“WordCruncher” screen output)

before: The data they contain do not represent one text in the linguistic sense of this word, but several texts that are related to each other by their contents and their structure but differentiated from each other by their language. A language-specific retrieval to be undertaken under these conditions presupposes the languages involved to be clearly identifiable, i.e., separable from one another so that analyses concerning the individual languages can be performed as well as analyses concerning the mutual relationship of, e.g., linguistic units of a translated text and its model.

A software solution that matches these requirements to a certain extent by combining a reasonable amount of functions designed for linguistic analysis with a programmable language-specific interface, is the system provided by “WordCruncher”. As I have discussed the facilities of this program recently in another context[2], I shall confine myself to a brief resumé here before trying to evaluate its efficiency.

In its present release (5.3) which is unfortunately available as a MS-Windows version only, the “WordCruncher” retrieval system[3] allows to index and analyse multilingual text files of up to 2 GB size[4]. The number of languages represented in a given document should not exceed ten if they are to be analysed separately and if the data are to be accessed from a remote server; otherwise up to 31 languages can be kept distinguished within one text file. The retrieval presupposes a preindexation which consists in the adaptation of several file formats: The initial input requires a plain (8-bit ASCII/ANSI or DOS) text file containing the textual data with additional tags storing information about the inherent structure of the text (e.g., page and line numbers, chapter or paragraph headings etc.) and its items. This input file (a so-called ETA file) must first be converted into a proprietary format (ETB file) which is ready for the preindexing process. The resulting index file (ETX) serves as the basis for the retrieval which is extremely fast as it involves no further (sequential) searching.

---

[2] GIPPERT (1997), 75-93.

[3] Developed by Brigham Young University (since 1985). The text viewer necessary for the retrieval of preindexed data (both locally and from a remote server) can be downloaded free of charge from the TITUS web pages (cf. `http://titus.uni-frankfurt.de/-texte/tituswc2.htm`). For the full version cf. `http://www.wordcruncher.com` or contact `johnston@wordcruncher.com` .

[4] The maximum file size depends on the operating system, not on the program.

Information concerning languages is stored in a twofold way in this system. Within the document to be preindexed, it is concealed in so-called TEXT_STYLE tags which represent, at a first glance, attributes of the output of written data rather than information about their linguistic properties. This can be seen in the example given in table 1 which shows the ETA file structure that corresponds to the NT passage illustrated in figure 5: All tags beginning with `T` such as `<Tmge16>` or `<Tgr16>` mark a certain “text style” that is defined by the use of a special font (e.g., Mxedruli-Georgian), together with its size (e.g., 16pt) and other attributes such as regular or italicised style. Via a FONT declaration, the same tags further refer to a LANGUAGE definition which is the basis of the language-specific retrieval. The latter information is not stored in the ETA file itself, though, but in a special Style Include File (SIF) which is required by the preindexing process. Lastly, the LANGUAGE definition may comprise information about a special keyboard assignment as well as the sorting order to be adapted for the data in question; these data are stored in extra files again (LST files). This complex system of interrelated information items is completed by the definition of text levels which is stored in the ETX file. Cf. tables 2 to 5 and figures 6 to 7 where some elements are illustrated (note that direction is part of the FONT declaration).

```
⟨Pnormal⟩
|p9
⟨Tdge16⟩esret ilocevdi tkuen. mamao ─ueno romeli xar cata Ωina.
     ┬mida iτavn saxeli Ωeni:⟨Tn16⟩
⟨Tcgei16⟩esret ilocevdi tkuen. mamao ─ueno romeli xar cata Ωina.
     ┬mida iτavn saxeli Ωeni:⟨Tn16⟩
⟨Tmge16⟩xolo tkuen esret ilocevdit: mamao ─ueno, romeli xar cata
     Ωina, ┬mida iτavn saxeli Ωeni,⟨Tn16⟩
⟨Tcgei16⟩xolo tkuen esret ilocevdit: mamao ─ueno, romeli xar
     cata Ωina, ┬mida iτavn saxeli Ωeni,⟨Tn16⟩
⟨Tgr16⟩O¡Ω∩@ o╓Σ τΦoΘ█úφ█Θ ██ ⁿπ█ç@_ / ╛Ω█Φ {π║Σ √ ╨Σ Ωoç@ o╙Φ╫-
     Σoç@, / °┘▀╫Θ▐⟨Ω∩ Ωò ÉΣoπ Θoδ¯,⟨Tn16⟩
⟨Thy16⟩Ew ard aysp╓s ka├╓╫ dow╫ ya┘π±s. Hayr mer or yerkins,
     sowrb e┘i├i anown ╫o:⟨Tn16⟩
⟨Thyti16⟩Ew ard aysp╓s ka├╓╫ dow╫ ya┘π±s. Hayr mer or yerkins,
     sowrb e┘i├i anown ╫o:⟨Tn16⟩
⟨Tcesk16⟩hkn& hkyl ϕlw &ntwn &bwn dbΩmy& ntqdΩ Ωmk⟨Tn16⟩
⟨Pright⟩
        ⟨Tsye16⟩hk?‟ hk0L ~j8 ˆntwW ˆb8W Db}.0″ ntqd\ |.6⟨Tn16⟩
        ⟨Tsyn16⟩hk?$ hk0L ~j8 ˆntwW ˆb8W Db}.0$ ntqd\ |.6⟨Tn16⟩
        ⟨Tsys16⟩hk?‟ hk0L ~j8 ˆntwW ˆb8W Db}.0″ ntqd\ |.6⟨Tn16⟩
```

Table 1: WordCruncher input (ETA) file containing Mt. 6,9

```
...
TEXT_STYLE mge16
     FONT georgian-mcxeta
     INDEX_FLAG on
     FONT_POINTSIZE 16
     TEXT_COLOR [0 128 0]
                [255 255 255]
     END
TEXT_STYLE mge22
     FONT georgian-mcxeta
     INDEX_FLAG on
...
```

Table 2: TEXT_STYLE definition

```
...
FONT georgian-mcxeta
  FONT_NAME Titus Mxedruli
  FONT_FAMILY roman
  CHAR_SET ansi
  PITCH proportional
  DIRECTION left-to-right
  FONT_TYPE TrueType
  LANGUAGE Oldgeorgian
  END
FONT greek
...
```

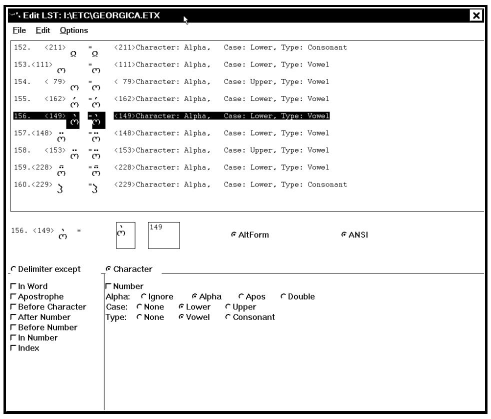
Table 3: FONT declaration

```
...
LANGUAGE Oldgeorgian
  LANGUAGE_ID GEORGIAN
  CHARACTER_MAP Georgian
  TEXT_STYLE mge16
  LST_FILENAME GEORGICA.ETX
  END
LANGUAGE Greek
  LANGUAGE_ID GREEK
  CHARACTER_MAP Greek
  TEXT_STYLE grx16
...
```

Table 4: LANGUAGE definition

```
...
CHARACTER_MAP Georgian
     MAPS   201=144   230=145
198=146    244=147    246=148
242=149    251=150    249=151
255=152    214=153    220=154
162=155 163=156 165=157
199=128    252=129    233=130
226=131    228=132    224=133
229=134    197=143    131=159
225=160    237=161    243=162
250=163 241=164 ...
```

Table 5: Keyboard definition


Edit LST: I:\ETC\GEORGICA.ETX

```
152.   <211> ჵ  = ჵ   <211>Character: Alpha,   Case: Lower, Type: Consonant
153.<111> ო     = ო   <111>Character: Alpha,   Case: Lower, Type: Vowel
154.   < 79> ო  = ო   < 79>Character: Alpha,   Case: Upper, Type: Vowel
155.   <162> ო́  = ო́   <162>Character: Alpha,   Case: Lower, Type: Vowel
156.   <149> ო̀  = ო̀   <149>Character: Alpha,   Case: Lower, Type: Vowel
157.<148> ო̈     = ო̈   <148>Character: Alpha,   Case: Lower, Type: Vowel
158.   <153> ო̈  = ო̈   <153>Character: Alpha,   Case: Upper, Type: Vowel
159.<228> ო̄     = ო̄   <228>Character: Alpha,   Case: Lower, Type: Vowel
160.<229> ჳ     = ჳ   <229>Character: Alpha,   Case: Lower, Type: Consonant

156. <149> ო̀  =       ო̀   149
```

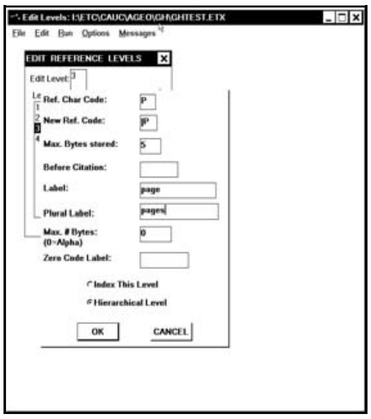
[✓ AltForm] [✓ ANSI]

Delimiter except [✓ Character]

- In Word
- Apostrophe
- Before Character
- After Number
- Before Number
- In Number
- Index

Number
Alpha: Ignore [✓ Alpha] Apos Double
Case: None [✓ Lower] Upper
Type: None [✓ Vowel] Consonant

Figure 6: Definition of sorting order


Edit Levels: I:\ETC\CAUC\AGEO\GH\GHTEST.ETX

EDIT REFERENCE LEVELS

Edit Level: 3

Ref. Char Code:	P
New Ref. Code:	P
Max. Bytes stored:	5
Before Citation:	
Label:	page
Plural Label:	pages
Max. # Bytes: (0=Alpha)	0
Zero Code Label:	

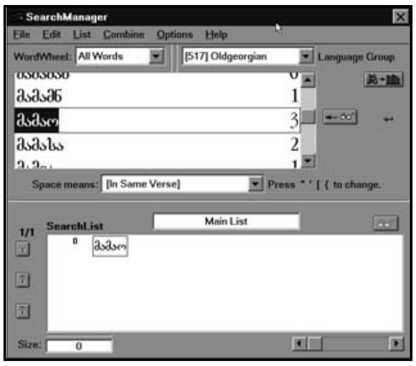
Index This Level
[✓ Hierarchical Level]

OK CANCEL

Figure 7: Definition of text levels

Once this information has been established, it can be used by the preindexing program for processing. The resulting information, consisting of language-specific indexes of word forms, is then available for immediate retrieval via a so-called “word wheel”, i.e. a look-up menue which displays the contents of the indexes in the respective script and alphabetic order of the language in question as defined in the LST file (cf. figures 8 to 12 showing the word-wheels for Georgian, Armenian, Syriac, and Greek, with a focus on the words meaning “Father”). This search engine meets further requirements of linguistic retrieval by enabling its user to extend the search to other preindexed texts collected in a “library”, to group (inflected) word forms pertaining to one lemma together, to investigate into grammatical features (provided these were prepared for indexing by entering appropriate tags) or to search for word forms that appear side by side within a certain distance and/or within certain levels of the text.

As to multilingual texts (or corpora, in the sense outlined above), the facilities provided by the WordCruncher system are not very elaborate yet. Even in the present release, it is possible, however, to formulate a cross-linguistic search such as the one indicated in figure 13. In this example, the search object is a context in which the Greek vocative πάτερ, the Armenian *hayr*, and the Georgian *mamao*, all meaning “father”, occur within the same verse. The result we expect to find is Mt. 6,9, but other contexts such as Mt. 26,42 will be found as well. Of course, this is not sufficient for an automatical generation of complete lists of the correspondences that exist between individual items of “parallel” texts, but it may give a first idea of where further investigation might be necessary.



WordWheel: All Words
[517] Oldgeorgian

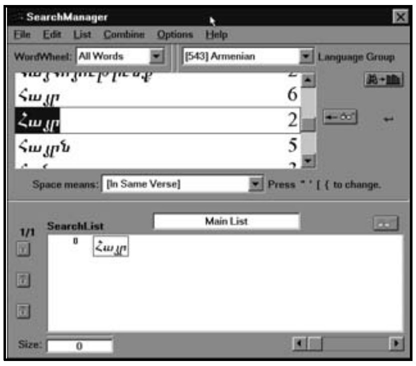
მამან	1
მამაო	3
მამასა	2

Space means: [In Same Verse]

0 მამაო

Size: 0

Figure 8: Georgian “word wheel”



WordWheel: All Words
[543] Armenian

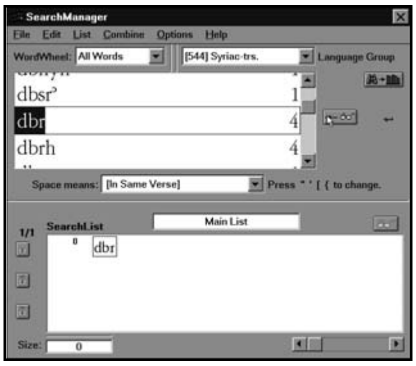
հայր	6
Հայր	2
հայրն	5

Space means: [In Same Verse]

0 Հայր

Size: 0

Figure 9: Armenian “word wheel”

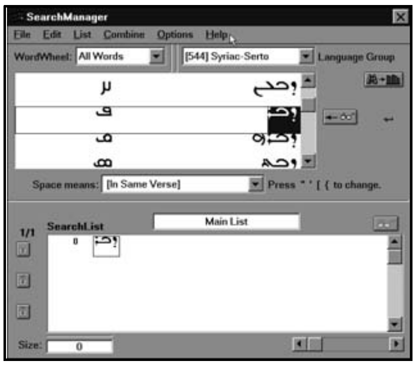

WordWheel: All Words [544] Syriac-trs. Language Group

dbsrʾ	1
dbr	4
dbrh	4

Space means: [In Same Verse] Press " ' [{ to change.

SearchList: dbr

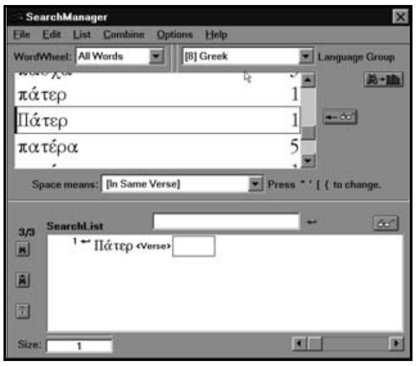
Size: 0


WordWheel: All Words [544] Syriac-Serto Language Group

Space means: [In Same Verse] Press " ' [{ to change.

Size: 0

Figures 10-11: Syriac "word wheels": Transcription vs. original script


WordWheel: All Words [8] Greek Language Group

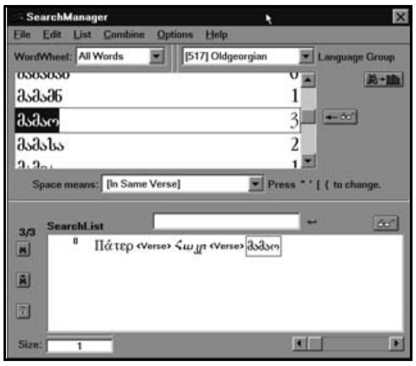
πάτερ	1
Πάτερ	1
πατέρα	5

Space means: [In Same Verse] Press " ' [{ to change.

SearchList: Πάτερ <Verse>

Size: 1

Figure 12: Greek "word wheel"


WordWheel: All Words [517] Oldgeorgian Language Group

მამან	1
მამაო	3
მამასა	2

Space means: [In Same Verse] Press " ' [{ to change.

SearchList: Πάτερ <Verse> Հայր <Verse> მამაო

Size: 1

Figure 13: Mixed language retrieval

Returning to the question of encoding, it has to be stated that the system used by WordCruncher is rather disadvantageous with respect to multilingual corpora. It is at least two features that must be mentioned here. First, the system is clearly surface-oriented, with language tagging being unseparable from (and even dependent on) script tagging. Although this feature did not prove to be impedient when the WordCruncher system was adopted to the requirements of the TITUS project, a data base consisting of more than 1 GB of mono- and multilingual corpora in Indo-European and adjacent languages[5], it will pose serious problems when the data

[5] Cf. GIPPERT (1996) for a detailed description and `http://titus.uni-frankfurt.de/texte/texte2.htm` for up-to-date information.

are to be converted into another format such as an SGML or XML tagged structure. The same holds true for the second disadvantage which consists in the fact that the present 8-bit basis does not allow for a unique and consistent encoding of several scripts, let alone languages. A multilingual corpus such as the one illustrated above is, in its WordCruncher representation, characterised throughout by so-called “font mapping” which means that a given byte value (e.g. #97) is used for different purposes (e.g., Latin *a*, Greek α, Armenian ա, or Georgian ა) depending on the actual font (cp. table 1). As a result of this, it is hardly possible to recover the intended structure whenever portions of the text are to be exported. In my personal view, the principle of font mapping, typical for WYSIWYG applications, has proved to be the most disastrous circumstance of 8-bit encoding with respect to data exchange, and time is ripe indeed to get rid of it.

With the 16-bit standard of UNICODE approaching its completion, a radical solution for this problem seems to be close, all the more since the unique encoding of 65,536 characters it implies can already be used to a certain extent in an 8-bit based environment[6]. It remains doubtful, however, whether the requirements I have exemplified above will indeed be met by Unicode implementations in near future.

Let us look at the *Aogəmadaēcā* manuscript again to see what problems we have to face when trying to use Unicode for the encoding of multilingual text documents written in several ancient scripts. The first problem consists in the fact that two of the scripts involved, viz. Avestan and Middle Persian-Pahlavī, are not and will not be part of the 16-bit Unicode standard proper (while Devanāgarī is). They are planned for storage in the so-called “surrogate area” instead which requires a 4-byte encoding and which is not accessible at present in any environment.

[6] The two leading WWW browsers, Netscape Navigator / Communicator and Microsoft Internet Explorer, have implemented Unicode support in their latest versions (4.0 and higher). What they process is not plain Unicode encoding, however, but a 7-bit or 8-bit transformation of it (UTF-7 and UTF-8). The capabilities of handling the data further depend on the operating system: By now, only MS-Windows 95, 98 and NT (4.0 and higher) are fully Unicode-compatible. For details cf. `http://titus-uni-frankfurt.de/unicode/unitest2.htm` which contains links to several sample pages; many texts of the TITUS data base are also available in UTF-8 format, cf. `http://titus-uni-frankfurt.de/texte/texte2.htm` .

Furthermore, there is no way yet to cope with the problem of script direction involved here: Although Arabic or Hebrew scripts have been included in the standard, the automatic handling of script directions which is declared to be the task of “rendering engines” that interpret and process the encoding[7], has not yet been implemented. The same holds true for the switching between different variants of characters (isolated, initial, central, and final, depending on the context) which is a typical feature of the Arabic script but also of Pahlavī: Unicode, by its structure, provides an encoding for basic “character” values only, not for variants that are regarded as “glyphs”, leaving the task of selection to “intelligent fonts” or “rendering engines” again[8].

A similar problem is met with when Devanāgarī is to be processed. Here, too, the encoding provided by Unicode is confined to basic characters, which in the case of Indic scripts means syllabic units like क, प, or भ, i.e. *ka*, *pa*, and *b^{h}a* (cf. figure 14 showing the Unicode block in ques-

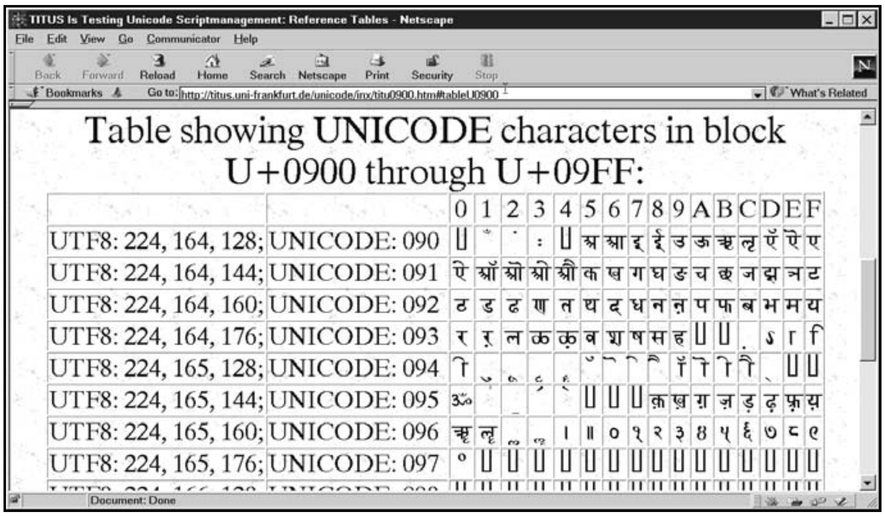

TITUS Is Testing Unicode Scriptmanagement: Reference Tables - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Go to: http://titus.uni-frankfurt.de/unicode/inx/titu0900.htm#tableU0900
What's Related
Document: Done

Table showing UNICODE characters in block U+0900 through U+09FF:

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
UTF8: 224, 164, 128;	UNICODE: 090	□	ँ	ं	ः	□	अ	आ	इ	ई	उ	ऊ	ऋ	ऌ	ऍ	ऎ	ए
UTF8: 224, 164, 144;	UNICODE: 091	ऐ	ऑ	ऒ	ओ	औ	क	ख	ग	घ	ङ	च	छ	ज	झ	ञ	ट
UTF8: 224, 164, 160;	UNICODE: 092	ठ	ड	ढ	ण	त	थ	द	ध	न	ऩ	प	फ	ब	भ	म	य
UTF8: 224, 164, 176;	UNICODE: 093	र	ऱ	ल	ळ	ऴ	व	श	ष	स	ह	□	□	़	ऽ	ा	ि
UTF8: 224, 165, 128;	UNICODE: 094	ी	ु	ू	ृ	ॄ	ॅ	ॆ	े	ै	ॉ	ॊ	ो	ौ	्	□	□
UTF8: 224, 165, 144;	UNICODE: 095	ॐ	॑	॒	॓	॔	□	□	□	क़	ख़	ग़	ज़	ड़	ढ़	फ़	य़
UTF8: 224, 165, 160;	UNICODE: 096	ॠ	ॡ	ॢ	ॣ	।	॥	०	१	२	३	४	५	६	७	८	९
UTF8: 224, 165, 176;	UNICODE: 097	॰	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

Figure 14: Devanāgarī encoding in Unicode

[7] According to the Unicode standard, the encoding must be in left-to-right order in all cases. This means that right-to-left direction is a matter of the output only.

[8] It is true that for Arabic, a large number of “glyphs” have been included as encodable units among the so-called “presentation forms” (Unicode blocks FB00 to FEFF). Their usage is not recommended, though, because it violates the basic encoding principle.

क्क क्क्य क्ख क्ख्य क्त क्त्य क्त्र क्त्र्य क्त्व क्थ क्न क्न्य क्प क्प्य क्फ क्फ्य क्म क्म्य क्य क्र क्र्य क्ल क्ल्य क्व क्व्य
ख्न ख्ण ख्म ख्म्य ख्य ख्व क्ष क्ष्य ग्द ग्द्य ग्ध ग्ध्य ग्ध्व ग्न ग्न्य ग्ब ग्ब्र ग्भ ग्भ्य ग्म ग्म्य ग्य ग्र ग्र्य ग्ल ग्व
घ्न घ्म घ्य घ्र ङ्क ङ्क्त ङ्क्ष ङ्क्ष्व ङ्ख ङ्ख्य ङ्ग ङ्ग्य ङ्ग्र ङ्ग्र्य ङ्घ ङ्घ्र ङ्ङ ङ्न ङ्म
च्च च्छ च्छ्य च्छ्र च्छ्व च्म च्य छ्य छ्र छ्र्य छ्व ज्ज ज्ज्ञ ज्ज्व ज्झ ज्ञ ज्ञ्य ज्म ज्य ज्र ज्व
ञ्च ञ्च्य ञ्छ ञ्छ्र ञ्ज ञ्ज्म ञ्ज्य ञ्झ ञ्श
ट्क ट्ट ट्य ट्व ट्स ठ्य ड्ग ड्ग्र ड्ड ड्य ढ्य ढ्र ढ्व ण्ट ण्ठ ण्ड ण्ढ ण्ण ण्म ण्य ण्व
त्क त्त त्त्य त्त्र त्त्व त्थ त्न त्न्य त्प त्प्र त्प्र्य त्फ त्म त्म्य त्य त्र त्र्य त्ल त्स त्स्न त्स्न्य त्स्य त्स्व
थ्य द्ग द्ग्र द्घ द्घ्य द्घ्र द्द द्द्य द्ध द्ध्य द्ध्व द्ध्व्य द्न द्न्य द्ब द्ब्र द्भ द्भ्य द्म द्य द्र द्र्य द्व द्व्य द्व्र ध्न ध्म ध्य ध्र ध्व
न्त न्त्य न्त्र न्त्स न्थ न्द न्द्य न्द्र न्द्व न्ध न्ध्य न्ध्र न्ध्व न्न न्न्य न्प न्प्र न्फ न्म न्य न्व न्स
प्त प्त्य प्न प्न्य प्प प्फ प्म प्य प्र प्ल प्स फ्य ब्घ ब्ज ब्द ब्ध ब्ध्व ब्ब ब्भ ब्य ब्र भ्न भ्य भ्र भ्व
म्न म्प म्प्र म्ब म्ब्य म्भ म्भ्र म्म म्य म्र म्ल
य्य य्व ल्क ल्ग ल्प ल्म ल्य ल्ल ल्व ल्ह व्य व्र व्व श्च श्च्य श्न श्न्य श्म श्य श्र श्र्य श्ल श्व श्व्य श्श
ष्क ष्क्र ष्ट ष्ट्य ष्ट्र ष्ट्र्य ष्ट्व ष्ठ ष्ठ्य ष्ठ्र ष्ण ष्ण्य ष्प ष्प्र ष्म ष्य ष्व
स्क स्क्र स्ख स्त स्त्य स्त्र स्त्व स्थ स्थ्य स्न स्प स्प्र स्फ स्म स्म्य स्य स्र स्व स्स ह्ण ह्न ह्य ह्र ह्ल ह्व

Table 6: Devanāgarī ligatures

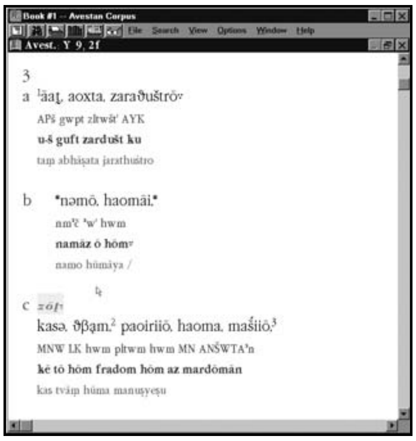
tion). When used for languages such as Sanskrit, however, Devanāgarī comprises a huge set of additional signs, mostly ligatures representing consonant clusters and the like; cf. table 6 where the most frequent ones are listed. As long as no rendering engine for Devanāgarī exists, there is no way to reproduce a Unicode-encoded Sanskrit text in a typographically acceptable form in its original script without violating Unicode prescriptions[9].

This means that for languages like Sanskrit, Avestan, or Middle Persian, the encoding of original scripts will remain difficult, if not impossible, for a certain period of time, and projects dealing with these languages, such as the TITUS data base, are better advised to stick to transcriptional solutions instead. But even then they will meet with considerable problems when trying to adopt Unicode encoding.

First of all, the transcription of Indic, Iranian and many other languages requires a lot of diacritic combinations that are not used in standardized Roman alphabets and are therefore not encodable as such, i.e., as

[9] In the Sanskrit sample pages accessible from `http://titus.uni-frankfurt.de/unicode/unitest.htm#samples` , the problem was solved by assigning one block of the so-called “user definable area” (Unicode E900-E9FF) to Devanāgarī glyphs of the indicated type.

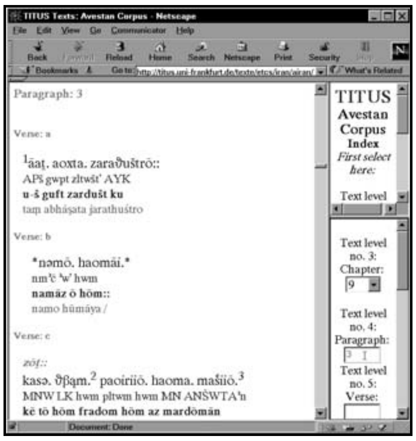
“precomposed characters”, in Unicode. In these cases, we have to employ the method of encoding basic characters and diacritics separately (i.e., as sequences such as *s* + ˇ + ´ for *š́*), which may pose serious problems for “rendering engines” or “intelligent fonts” again when the relative positioning of the items is concerned (cf. figures 15 and 16 showing a text passage from the Avestan *Hōm-Yašt*, *Yasna* 9,3, containing transcriptional Avestan, Middle Persian, Pazend and Sanskrit, in both 8-bit and 16-bit encoding).


Book #1 -- Avestan Corpus
File Search View Options Window Help
Avest. Y 9, 2f

3
a ¹āaṯ. aoxta. zaraϑuštrō·
APš gwpt zltwšt' AYK
u-š guft zardušt ku
taṃ abhāṣata jarathuštro

b *nəmō. haomāi.*
nm'č 'w' hwm
namāz ō hōm·
namo hūmāya /

c *zōt:*
kasə. ϑβąm.² paoiriiō. haoma. mašiiō.³
MNW LK hwm pltwm hwm MN ANŠWTA'n
kē tō hōm fradom hōm az mardōmān
kas tvāṃ hūma manuṣyeṣu


TITUS Texts: Avestan Corpus - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Go to: http://titus.uni-frankfurt.de/texte/etcs/iran/airan/ What's Related
Document: Done

Paragraph: 3

Verse: a

¹āaṯ. aoxta. zaraϑuštrō::
APš gwpt zltwšt' AYK
u-š guft zardušt ku
taṃ abhāṣata jarathuštro

Verse: b

nəmō. haomāi.
nm'č 'w' hwm
namāz ō hōm::
namo hūmāya /

Verse: c

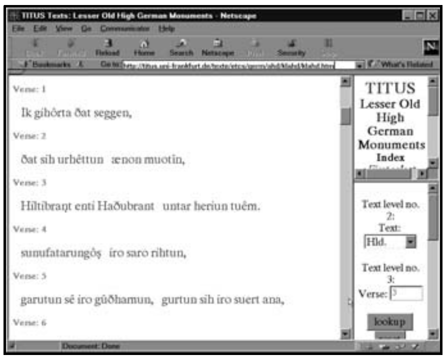
zōt::
kasə. ϑβąm.² paoiriiō. haoma. mašiiō.³
MNW LK hwm pltwm hwm MN ANŠWTA'n
kē tō hōm fradom hōm az mardōmān

TITUS
Avestan Corpus Index
First select here:
Text level

Text level no. 3: Chapter: 9
Text level no. 4: Paragraph: 3 1
Text level no. 5: Verse:

Figure 15-16: Y. 9,3 in 8-bit (WC) and 16-bit encoding (UTF-8 HTML)

The treatment of diacritic combinations is indeed a weak point in the concept of Unicode. By offering more than one possibility of encoding for characters like *ä* or *š* which can be treated both as “precomposed characters” and as combinations consisting of *a* + ¨ or *s* + ˇ, Unicode paves the way for arbitrary decisions that are diametrically opposed to the principle of “unique encoding”. Furtheron, Unicode does not, at least in its present state, distinguish diacritics in any way according to their function, the outer shape being taken as their basic property only. Thus it is not possible to differentiate the subscript dot that appears in *ṣ* as rendering the Sanskrit retroflex [ʂ] sound (cf. Devanāgarī ष = *ṣa*), from the dot below used in manuscript editing for denoting uncertain readings (cf. *sunufatarungoṣ* in the TITUS edition of the Old High German *Hildebrandslied*, figure 17). Moreover, the diacritics that have been included as such in Unicode (cf. figure 18 showing the block in question) were obviously intended not to be script-dependent; this means that one and the same “diaeresis” (¨) would have to be used for Latin-based *ä* or *f̈*, Greek ε̈, Cyrillic ӭ, and maybe also for ̈ which would represent Pahlavī *y*. It goes without saying that this must lead to typographically unacceptable results, at least as long as special rendering engines are not available. Lastly, the stock of diacritics that have been incorporated is by no means complete for Latin-based transcription systems, let alone other uses. This may well be due to the fact that, at least for parts, Unicode encoding schemes were created not on the basis of thorough investigation into the necessities of the languages and scripts concerned but by mixing existing standards with accidental collections. To overcome this state, the TITUS project has started to build up a data base where diacritics and diacritic combinations appearing in scientific publications are stored so that they can be used for a documentation which will hopefully lead to an extension of the Unicode standard. At present, this data base is being prepared for interactive usage via the WWW (the URL will be `http://titus.uni-frankfurt.de/unicode/unicsel/unicsel.htm`; cp. figure 19 showing the provisional template).


TITUS Texts: Lesser Old High German Monuments - Netscape
File Edit View Go Communicator Help
Reload Home Search Netscape Security
Bookmarks Go to: What's Related
Document: Done

Verse: 1

Ik gihôrta ðat seggen,

Verse: 2

ðat sih urhêttun ænon muotin,

Verse: 3

Hiltibrant enti Haðubrant untar heriun tuêm.

Verse: 4

sunufatarungôs iro saro rihtun,

Verse: 5

garutun sê iro gûðhamun, gurtun sih iro suert ana,

Verse: 6

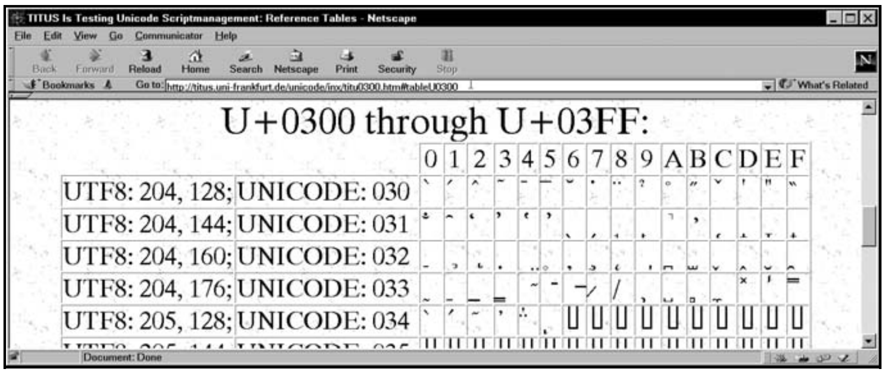
TITUS Lesser Old High German Monuments Index

Text level no. 2: Text: Hld.

Text level no. 3: Verse: 5

lookup

Figure 17: Passage from Hildebrandslied

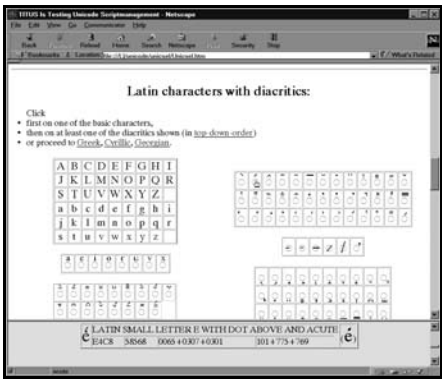

TITUS Is Testing Unicode Scriptmanagement: Reference Tables - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Go to: http://titus.uni-frankfurt.de/unicode/inx/titu0300.htm#tableU0300
What's Related
Document: Done

U+0300 through U+03FF:

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
UTF8: 204, 128;	UNICODE: 030	̀	́	̂	̃	̄	̅	̆	̇	̈	̉	̊	̋	̌	̍	̎	̏
UTF8: 204, 144;	UNICODE: 031	̐	̑	̒	̓	̔	̕	̖	̗	̘	̙	̚	̛	̜	̝	̞	̟
UTF8: 204, 160;	UNICODE: 032	̠	̡	̢	̣	̤	̥	̦	̧	̨	̩	̪	̫	̬	̭	̮	̯
UTF8: 204, 176;	UNICODE: 033	̰	̱	̲	̳	̴	̵	̶	̷	̸	̹	̺	̻	̼	̽	̾	̿
UTF8: 205, 128;	UNICODE: 034	̀	́	͂	̓	̈́	ͅ	□	□	□	□	□	□	□	□	□	□

Figure 18: Unicode block containing diacritics

Another problem which the use of Latin based transcriptions instead of original scripts brings about, consists in the fact that no inherent indication of the languages involved is available in this case if Unicode is applied. It is of course true that language encoding has never been an objective of Unicode, the target of which are scripts and characters, and it is merely accidental that some scripts are clearly, or even uniquely, related with certain languages. For a true language-specific encoding, it will therefore remain necessary to provide explicit tagging even in a Unicode environment.


TITUS Is Testing Unicode Scriptmanagement - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location
What's Related

Latin characters with diacritics:

Click
- first on one of the basic characters,
- then on at least one of the diacritics shown (in top-down order)
- or proceed to Greek, Cyrillic, Georgian.

A	B	C	D	E	F	G	H	I
J	K	L	M	N	O	P	Q	R
S	T	U	V	W	X	Y	Z	
a	b	c	d	e	f	g	h	i
j	k	l	m	n	o	p	q	r
s	t	u	v	w	x	y	z	

LATIN SMALL LETTER E WITH DOT ABOVE AND ACUTE
E4C8 58568 0065+0307+0301 101+775+769

Figure 19: TITUS interactive data base of diacritic combinations

References

GIPPERT, Jost (1996): TITUS – Alte und neue Perspektiven eines indogermanistischen Thesaurus. In: *Studia Iranica, Mesopotamica & Anatolica* 2, 1996 [1997], 49-76.

— (1997): Multilingual text retrieval: Requirements and solutions. In: *Studia Iranica, Mesopotamica & Anatolica* 3, 1997 [1998], 75-93. Preliminary version in: R. COOPER / T. GAMKRELIDZE (eds.), *Proceedings of the Second Tbilisi Symposium on Language, Logic and Computation*, Sept. 16-21, 1997, Tbilisi 1998, 106-118.

HARTUNG, Johannes Gabriel (1659): ויכוח מראה שפשט נבואת מלאכי אינו יסבול הדעת שעור יבוא אליהו הנביא׃ sive de Eliæ, quem Judæi etiamnum frustrà præstolantur, adventu dissertatio, ex Malach. IV, 5.6. Jenæ: Literis Samuelis Krebsii.

HUTTER, Elias (1599-1643): Novum testamentum Domini nostri Jesu Christi, Syriace, Ebraice, Graece, Latine, Germanice, Bohemice, Italice, Hispanice, Gallice, Anglice, Danice, Polonice. Nürnberg.

JAMASPASA, Kaikhusroo M. (1982): Aogəmadaēcā. A Zoroastrian Liturgy. Wien: Verlag der Österreichischen Akademie der Wissenschaften (Sitzungsbericht, phil.-hist.Kl., 397. / Veröffentlichungen der Iranischen Kommission, 11.)