The Unicodify Suite

User's Manual

1. Introduction

a. About *Unicodify*

Unicodify is a suite of programs that map text files in a variety of 8-bit encodings to 16-bit little-endian Unicode.

The main program, *Unicodify*, handles conversion from a range of 8-bit fonts commonly used on the World Wide Web to encode the alphabets of South Asia. See section 2. Other programs handle the ISCII and PASCII encodings, and the text format exportable from the popular *Inpage* Urdu word-processing software. See sections 3 and 4.

Unicodify was created as part of the EMILLE Project (see www.emille.lancs.ac.uk). It was used to map the texts collected for the EMILLE Corpora to a standardised Unicode format. It has since been extended to handle the Latin-2 text encoding. **Unicodify** was written by Andrew Hardie of the Department of Linguistics and Modern English Language, University of Lancaster.

b. General notes

All the **Unicodify** programs supplied with this manual file have been compiled to function on a 32-bit Windows system.

However, all source code is ANSI-standard C, and is available on request to anyone who wishes to compile it for another system.

In general, this software is made available to anyone for any non-profit academic purposes.

All efforts have been made to ensure the accuracy of the programs described here, and their success rate is sufficiently high that all the text in the EMILLE Monolingual Written Corpora has been produced by one of these programs. *However, we are unable to guarantee perfect performance of the programs on any text file, or accept any liability for their failure to accurately map any text.*

All the programs are called from a command-prompt window with one or more arguments as set out in the more detailed descriptions below.

The output text is encoded as UTF-16, little-endian, with a byte-order mark (U+feff) at the start of each file. Line breaks are encoded as a U+000d, U+000a combination.

2. Unicodify

Unicodify, the main program of the suite, handles conversion from a range of 8-bit fonts commonly used on the World Wide Web to encode the alphabets of South Asia. It takes HTML input and produces CES-compliant[1] SGML output.

Unicodify takes as its input text in the HTML format produced by Microsoft Word. This is in contrast to *deisciify* and *disinpage*, which take plain text as their input. The historical reason for this is that the EMILLE Corpus texts were produced by analysts copying text from a web browser to Microsoft Word, and then saving the result as HTML. Since most text using one of these 8-bit fonts will probably be stored either as a web document or as a word-processed document, and not as plain text, there did not seem to be any reason to move away from this system.

Furthermore, using HTML as input has significant advantages: mark-up such as paragraphs can be preserved, <gap> elements can be inserted into the text for pictures, etc, and the program can avoid mapping any text that was not in the target font in the original file. The HTML format produced by Microsoft Word is used in preference to HTML files saved directly from the web because the process of copying the text through Word standardises the HTML code somewhat, making it easier for *Unicodify* to interpret. All HTML mark-up that has not been converted to CES SGML tags is stripped away from the final output file.

It is assumed that only one South Asian script font will be used in any one file. The corollary of this is that any text in the file which is not in the target font is treated as if it were standard, Latin-alphabet ASCII text.

As specified above, *Unicodify* maps a proportion of the HTML tags in the input into CES-compliant SGML tags in text. It also replaces the original HTML header with a CES header. This header is left empty, except for details of when the file was created by *Unicodify* and the file's name.

Unicodify is called from the command-prompt using the following argument format:

```
unicodify fontname input_filename output_filename
```

The arguments should be as follows:

`fontname`	The font in which the file is encoded, chosen from the list below
`input_filename`	The name of the text file to be mapped. This should be in the same directory as the unicodify.exe file. Furthermore, it should be an HTML file created using the "Save as Web Page" function in Microsoft Word.
`output_filename`	The name which the new, Unicode file is to be given. This will also appear in the same directory that the unicodify.exe file is in.

The following fonts[2] are handled by *Unicodify* and may be given as the first argument in a call to *Unicodify*:

[1] See http://www.cs.vassar.edu/CES/

[2] Since all these fonts are copyrighted, we cannot supply them with this suite of programs. However, most are available freely on the World Wide Web from one or more of the various sites that utilise them.

Font Name	Alphabet (Language(s))
Webdunia	Devanagari (Hindi, Marathi, Sanskrit, etc.)
Shree-Dev-0714	Devanagari (Hindi, Marathi, Sanskrit, etc.)
Express	Devanagari (Hindi, Marathi, Sanskrit, etc.)
Xdvng[3]	Devanagari (Hindi, Marathi, Sanskrit, etc.)
Gopika	Gujarati
AdarshaLipi / AdarshaLipiExp / AdarshaLipiNormal[4]	Bengali
Amrit-Lipi-Light	Gurmukhi (Punjabi)
Satluj	Gurmukhi (Punjabi)
GurbaniLipi	Gurmukhi (Punjabi)
AnmolLipi	Gurmukhi (Punjabi)
AnandpurSahib	Gurmukhi (Punjabi)
Punjabi	Gurmukhi (Punjabi)
Amudham	Tamil
DL-Manel-bold	Sinhala
kaputadotcom	Sinhala
MiANCL	Sinhala

Troubleshooting Unicodify

There is a great deal of variability in HTML and it is sometimes possible for Unicodify to be deceived by an input file. In these cases, strings of what look like gibberish will appear in the output. This occurs if Unicodify thought that a particular block of text was in the Latin alphabet, when actually it was in the target font.

If this occurs extensively and systematically, it should be reported as a bug. However, if it only occurs occasionally, the problem can be rectified by following these steps:

[3] For *Xdvng* files that have been produced manually, using entity numbers (such as &#242), rather than saved from MS Word, the font name **XdvngPre** should be used to tell the program that an additional preprocess is required.
[4] These fonts may appear together. A file which contains more than one of these three fonts can be mapped even if only one is specified on the command-line.

a. Copy the block of gibberish from the Unicode output file into a blank document in Microsoft Word

b. Set the font of the block to the target font of the original text. It should now appear as legible South Asian-script text. *If the text you have copied contains any SGML elements in <angled brackets>, these **must** be left as Times New Roman or another rLAtin-alphabet font.*

c. Save this document as a web page.

d. Run the new HTML file through Unicodify in the normal way.

e. Open the new output file. It will contain a Unicode version of text that previously came out as gibberish. You can copy this and paste it over the gibberish in your original Unicode file.

3. Deisciify

The deisciify program maps plain text encoded as ISCII (*Indian Standard Code for Information Interchange*) and PASCII (*Perso-Arabic Standard Code For Information Interchange*) to Unicode.

The ISCII standard consists of a generic encoding for Indian alphabets using the range of characters from decimal 128 to 255. Since the various writing systems of South Asia use the same structure of alphabet, with only the shapes of the glyphs and the way they combine together being different, a single value can represent the same consonant or vowel character in any of the alphabets[5]. The characters below 128 are used as in ASCII, to allow Latin alphabet letters and the basic range of control characters and operators to be used alongside the South Asian script.

The PASCII applies the same principle to the Perso-Arabic alphabet used to write many of the north-western Indo-Aryan languages (for instance, Urdu, Western Punjabi, Sindhi and Kashmiri).

It should be noted that the ISCII and PASCII encoding standards have both undergone changes over time. *Deisciify* handles the standards as they were at a particular moment in time. If a file you have mapped comes out as gibberish, it may be that the standard employed in encoding your file was not that expected by *deisciify*. In this case, you will need to map the file to the correct ISCII/PASCII format before using *deisciify*.

Deisciify has been extensively tested on ISCII texts and is believed to be fully reliable. It has also been found to be reliable for Urdu texts encoded as PASCII. However, its accuracy in handling other Perso-Arabic script languages has not been assessed and cannot be guaranteed.

Deisciify is called from the command-prompt using the following argument format:

```
deisciify alphabet input_filename output_filename
```

[5] Of course, the "same" vowel or consonant may not be pronounced identically in different languages, much as the letter E has many different pronunciations in the various languages that use the Latin alphabet.

The arguments should be as follows:

`alphabet`	The alphabet used in this text file, chosen from the list below[6]
`input_filename`	The name of the text file to be mapped. This should be in the same directory as the deisciify.exe file
`output_filename`	The name which the new, Unicode file is to be given. This will also appear in the same directory that the deisciify.exe file is in.

The alphabets which may be specified are:

Devanagari	Bengali	Gurmukhi	Gujarati
Oriya	Tamil	Telugu	Kannada
Malayalam	Urdu		

The language should be set to “Urdu” for any PASCII text.

4. Delatin2ify

The Latin-2 encoding uses the consists characters from decimal 128 to 255 to represent the accented character used in Eastern European languages such as Polish. The characters below 128 are used as in ASCII. In Unicode, however, the characters from 128 to 255 are encoded parallel to the Western European Latin-1 encoding, and the accented characters found in Latin-2 appear in a higher range.

Delatin2ify takes a plain text file in Latin-2 (for example, as produced by an Eastern European operating system) and maps it to Unicode.

Unlike the other programs in the Unicodify suite, it can *also* go the other way, and map Unicode text to Latin-2. Any characters that do not occur in Latin-2 will be mapped to a tilde (~).

Delatin2ify is called from the command-prompt. To map from Latin-2 to Unicode, use the following argument format:

```
delatin2ify input_filename output_filename
```

The arguments should be as follows:

`input_filename`	The name of the text file to be mapped. This should be in the same directory as the delatin2ify.exe file
`output_filename`	The name which the new, Unicode file is to be given. This will also appear in the same directory that the delatin2ify.exe file is in.

To map from Latin-2 to Unicode, use the following argument format:

```
delatin2ify input_filename output_filename R
```

[6] ISCII/PASCII has no file-internal means of specifying which writing system the upper 128 characters of the 8-bit range are being used to represent. Therefore, the user must specify which alphabet is being used.

The arguments should be as follows:

`input_filename`	The name of the text file to be mapped. This should be in the same directory as the delatin2ify.exe file
`output_filename`	The name which the new, Latin-2 file is to be given. This will also appear in the same directory that the delatin2ify.exe file is in.
`R`	**R**everse mode – this argument tells the program it should reverse the mapping process and map from Unicode to Latin-2.

5. Disinpage

The popular Urdu word-processor *Inpage* has a function which allows text to be exported to file from its proprietary file format. *Disinpage* maps this exported text to Unicode.

Please note that although *disinpage* can handle the normal run of Urdu or Western Punjabi text, its efficacy with regard to Classical (especially Qu'ranic) Arabic has not been tested and is, therefore, subject to considerable doubt.

Disinpage is called from the command-prompt using the following argument format:

```
disinpage input_filename output_filename
```

The arguments should be as follows:

`input_filename`	The name of the text file to be mapped (i.e. the file exported from *Inpage*). This should be in the same directory as the disinpage.exe file
`output_filename`	The name which the new, Unicode file is to be given. This will also appear in the same directory that the disinpage.exe file is in.