NAME

grconv – universal Greek character code converter

SYNOPSIS

grconv [**–S enc**] [**–s cs**] [**–t cs**] [**–T enc**] [**–h**] [**–d c**] [*file*...]
grconv [**–S enc**] [**–s cs**] **–x xl** [**–d c**] [*file*...]
grconv –r [**–t cs**] [**–T enc**] [**–h**] [**–d c**] [*file*...]
grconv –v|**–L**|**-R**

DESCRIPTION

Grconv converts between a large number of character sets, transcription, and transliteration methods that are used to represent Greek text. In addition, it supports a number of encodings used to represent those character sets in different environments. *Grconv* reads the file(s) specified in its command line printing the converted results on its standard output, or runs as a filter, reading text from its standard input printing the converted result on its standard output; the redirection operator > can be used to write to files. *Grconv* can be used in three different ways:

1. To convert between one character set and its representation method into another.
2. To transcribe or transliterate Greek text represented using a specified method into plain Latin characters.
3. To convert Latin text produced by the above transliteration operation back into a specified representation of Greek text.

The following character sets can be specified for the source and the target text:
MS737, cp737, 737, 437GR, IBM423, cp423, ebcdic-cp-gr, Latin-greek-1, iso-ir-27, greek7-old, iso-ir-18, greek-ccitt, iso-ir-150, latin-greek, iso-ir-19, greek7, iso-ir-88, IBM851, cp851, 851, MAC_GR, ISO_5428:1980, iso-ir-55, MS1253, cp1253, Windows, Windows-1253, ISO10646, ISO_10646, ISO-10646, Unicode, IBM869, cp869, 869, cp-gr, ISO_8859-7:1987, 928, ELOT928, iso-ir-126, ISO_8859-7, ISO-8859-7, ELOT_928, ECMA-118, greek, and greek8.

Note that many of the above names are aliases for the same character set. Descriptions of these character sets can be found in RFC-1947 and RFC-1345.

MS737
(cp737, 737, 437GR) Microsoft DOS code page 737 also known as 437G. Note that some implementations of this code page do not include capital letters with acute or diairesis.

IBM423
(cp423, ebcdic-cp-gr) IBM EBCDIC code page 423.

Latin-greek-1
(iso-ir-27) Latin Greek 1. This page only contains the Greek capital letters whose glyphs do not exist in the Latin alphabet. The other capital letters are rendered using the equivalent Latin letter (e.g. "Greek capital letter alpha" is rendered as "Latin capital letter A"). When mapping "Latin Greek 1" text to ISO 8859-7 the Latin capital letters should only be transcribed to the equivalent Greek ones if a suitable heuristic determines that the specific Latin letters are used to represent Greek glyphs. *Grconv* does not perform this function.

greek7-old
(iso-ir-18) old 7 bit Greek.

greek-ccitt
(iso-ir-150) Greek CCITT.

latin-greek
(iso-ir-19) Latin Greek.

greek7
(iso-ir-88) 7 bit Greek.

IBM851
(cp851, 851) IBM code page 851 used on OS/2 and PS/2 machines.

MAC_GR
: The Greek code page implemented on the Apple Macintosh computers. Contributed by Michalis Tsaoutos.

ISO_5428:1980
: (iso-ir-55) character set ISO 5428:1980.

MS1253
: (cp1253, Windows, Windows-1253) a character set similar to 8859-7 as implemented in Microsoft Windows 3.1/3.11, and Microsoft Windows 95/98.

ISO10646
: (ISO_10646, ISO-10646, Unicode) the Unicode standard 1.1.

IBM869
: (cp869, 869, cp-gr) IBM code page 869.

ISO_8859-7:1987
: (928, ELOT928, iso-ir-126, ISO_8859-7, ISO-8859-7, ELOT_928, ECMA-118, greek, greek8) the current officially sanctified 8-bit character set ISO 8859-7:1987.

Unicode data can be read or written using the following encodings:
UCS-2, UCS-16, UCS-16BE, UCS-16LE, UTF-8, UTF-7, Java, and HTML.

UCS-2, UCS-16
: The UCS-2 and UCS-16 encodings represent all Unicode characters using two bytes. On input the byte-swapped version of the Unicode character 65534 (ZERO WIDTH NON-BREAKING SPACE 0xfffe) is transparently used for byte-order detection purposes.

UCS-16BE
: The UCS-16BE encoding represents all Unicode characters using two bytes in big-endian (network order) ordering.

UCS-16LE
: The UCS-16LE encoding represents all Unicode characters using two bytes in little-endian (e.g. Intel) ordering. This format should not be used for generating portable output.

UTF-8
: The UTF-8 encoding represents ASCII characters using their ASCII encoding and other characters using a variable length 8-bit encoding scheme.

UTF-7
: The UTF-7 encoding represents ASCII characters using their ASCII encoding and other characters using a variable length 7-bit encoding scheme. It is therefore portable across communication channels supporting only 7 bit characters.

Java
: The Java encoding (also used by C++) represents Unicode characters using the sequence ‘‘\u*hexadecimal code*’’.

HTML
: The HTML encoding represents 8-bit or Unicode characters using the sequence ‘‘&#*decimal code*;’’. It is specified in RFC-2070.

All 8-bit character sets can be read or written using the following encodings:
8bit, Base64, Quoted, RTF, HTML, HTML-Symbol, HTML-Lat, and Beta.

8bit
: The 8bit encoding represents all characters using a single byte.

Base64
: The base-64 encoding represents groups of three 8 bit characters using four characters that are portable across many environments representing 6 bit quantities. This method is commonly used to transfer mail. It is specified in RFC-2045.

Quoted
: The quoted-printable encoding represents characters that would not be portable across disparate environments (e.g. those whose value is above 127) using the sequence ‘‘=*hexadecimal code*’’. This method is also commonly used to transfer mail and specified in RFC-2045.

URL
: The URL encoding represents Greek characters by encoding their Unicode representation in UTF-8 and then representing characters whose value is above 127 with the sequence

"*%hexadecimal code*". This method is specified in RFC-3986. *Grconv* will not encode URL reserved characters, as it does not know which have a special meaning in a URI schema.

RTF
: The RTF encoding is an attempt to support Microsoft's rich text format. Output produced in the RTF encoding using the CP1253 character set and a header (via the −*h* option) can be read by Microsoft Word. Since the program does not contain an RTF parser, RTF input converted into another character set needs to be hand-edited to remove RTF commands.

HTML
: The HTML encoding represents characters using the sequence "&#*decimal code*;".

HTML-Symbol
: The HTML-Symbol encoding represents Greek characters using a textual representation of the corresponding mathematical symbol character (e.g. α) as defined in the HTML 4 specification. Since this encoding should normaly be used to encode mathematical symbols and not Greek text its use to create new content is discouriaged; code HTML pages containing Greek text using ISO-8859-7 and 8 bit characters or Unicode and UTF-8.

HTML-Lat
: The HTML-Lat encoding represents characters whose value is above 127 using a textual representation of the corresponding ISO-8859-1 (Latin 1) character (e.g. È). This encoding is sometimes used by defficient HTML-editors and generators to represent Greek characters that occupy the same code positions in the ISO-8859-7 character set. Since this use is non-standard its use to create new content is discouriaged; code HTML pages containing Greek text using ISO-8859-7 and 8 bit characters or Unicode and UTF-8.

Beta
: The Beta encoding is mainly used to encode Greek classical texts in projects such as the TLG (Thesaurus Linguae Graecae). Since *gronv* targets modern Greek encodings, the acute, grave, and circumflex accents are converted to the modern Greek acute accent. In addition, the smooth and rough breathings and the iota subscript are lost. *Grconv* handles internaly Beta code switching from a Greek font to a Roman font, however most other Beta escape sequences are not processed.

OPTIONS

–S *enc*
: Specify the source encoding. The default source encoding is UCS-2 for Unicode and 8bit for all other character sets.

–s *cs*
: Specify the source character set. The default source character set is ISO-8859-7:1987 for 8-bit input encodings and Unicode for 16 bit input encodings (UCS-2, UCS-16, UCS-16BE, UCS-16LE, UTF-8, UTF-7, and Java).

–T *enc*
: Specify the target encoding. The default target encoding is UCS-2 for Unicode and 8bit for all other character sets.

–t *cs*
: Specify the target character set. The default target character set is ISO-8859-7:1987 for 8-bit output encodings and Unicode for 16 bit output encodings (UCS-2, UCS-16, UCS-16BE, UCS-16LE, UTF-8, UTF-7, and Java).

–x *xl*
: Specify *transcribe* to perform transcription of Greek text into Latin characters or *transliterate* to transliterate Greek text into Latin characters. Both transcription and transliteration are performed according to ISO 843:1997. The transliteration is a reversible operation that should be used when the results need to be converted back into exactly the same Greek text. The transcription is non-reversible, but attempts to model the way the Greek words are pronounced. It should be used to represent names of people, streets, etc. when an alternative way to obtain the original Greek spelling is available.

–r
: Perform reverse transliteration from Latin into Greek text.

–h
: Create header (and footer) for the output encoding. An encoding-specific header will be produced, typically describing the output encoding and character set, making it readable by appropriate software. As an example the HTML will be bracketted by appropriate tags.

−d *char*
: Specify the character to be used for non-existent mappings between character sets. The default character is space.

−v
: Display program version and copyright message.

−L
: List the supported encodings and character sets.

−R
: Print a ‘‘Rosetta stone’’ of a test phrase and the Greek character set transliterated, transcribed, in all supported character sets and their respective encodings. Under normal circumstances only one character set and encoding of the test phrase will be readable on an output device that supports rendering of Greek fonts. This option can therefore be used to decide the character set and encoding that is best supported in a particular environment.

EXAMPLES

Convert a file from MS-DOS to Windows Greek:
grconv -s 737 -t Windows <greek.txt >wingreek.txt

Transcribe a Unicode UTF-8 file to a readable ASCII representation (also known as Grenglish):
grconv -S UTF-8 -s Unicode -x transcribe <file.utf8

Transliterate an ISO-8859-7 file into ASCII:
grconv -x transliterate <file1.txt >file2.txt

Perform a reverse transliteration function; file3.txt will be identical to file1.txt of the previous example:
grconv -r <file2.txt >file3.txt

Convert an ISO-8859-7, base64 coded mail document into RTF to be read by Microsoft Word:
grconv -S Base64 -h -t Windows -T RTF <mail.txt >mail.rtf

Transcribe the Winv -s 928 -x transcribe | winclip -c

Your Windows clipboard contains Greek text generated under MS-DOS, or Greek text generated in an old Windows application that does not support Unicode or different code pages. Microsoft Office 2000 applications refuse to paste it as Greek and show weird symbols instead. The following pipeline will convert the clipboard data into Unicode Greek. Also note that the console code page must be set to the Greek OEM character set (code page 737) using the *chcp* command.
winclip -p | grconv -s 737 -T UCS-16LE | winclip -cu

For the last two examples, note that *winclip* is part of the *outwit* tool suite, probably available at the same site that hosts *grconv*.

SEE ALSO

D. Spinellis. Greek character encoding for electronic mail messages. Network Information Center, Request for Comments 1947, May 1996. RFC-1947.

K. Simonsen. Character Mnemonics and Character Sets. Network Information Center, Request for Comments 1345, June 1992. RFC-1345.

Unicode Consortium. *The Unicode Standard, Version 1.1*.

F. Yergeau, G. Nicol, G. Adams, and M. Duerst. Internationalization of the Hypertext Markup Language. Network Information Center, Request for Comments 2070, January 1997. RFC-2070.

N. Freed and N. Borenstein. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. Network Information Center, Request for Comments 2045, November 1996. RFC-2045.

F. Yergeau. UTF-8, a transformation format of ISO 10646. Network Information Center, Request for Comments 2279, January 1998. RFC-2279.

D. Goldsmith and M. Davis. UTF-7: A Mail-Safe Transformation Format of Unicode. Network Information Center, Request for Comments 2152, May 1997. RFC-2152.

The TLG Beta Code Manual. http://www.tlg.uci.edu/BetaCode.html

AUTHOR

BUGS

The ISO 843 transliteration specifies that a letter i macron and a letter o macron should be used to represent eta and omega. Since such glyphs are not part of any widely used character set we represent them by a letter i or o followed by an underscore. According to ISO 843 this implementation is allowed, but is performed at a risk of interoperability. *Grconv* can correctly perform the reverse translitaration using the underscore convention; the behaviour of other programs may vary.

Grconv is targeted towards the handling of Greek text. It will not deal correctly with ISO 10646 characters with a scalar value above 0x1000.

RFC 2045 specifies exactly how line breaks are to be converted into carriage return, line feed pairs when handling base-64 and quoted printable encodings. Since *grconv* is not directly tied to mail transfer mechanisms we handle line breaks using the underlying implementation of the system's C++ compiler. On Unix systems this means that carriage return, line feed pairs will almost certainly *not be produced* by *grconv*.

Grconv implements almost 100 combinations of standard input / output encodings and character set conversions resulting in around 10,000 possible transformations. The code has not been reviewed and extensively tested; it should be treated as beta quality.