Title

stata.com

unicode convertfile — Low-level file conversion between encodings

Description Syntax Options Remarks and examples Also see

Description

`unicode convertfile` converts text files from one encoding to another encoding. It is a low-level utility that will feel familiar to those of you who have used the Unix command `iconv` or the similar International Components for Unicode (ICU)-based command `uconv`. If you need to convert Stata datasets (`.dta`) or text files commonly used with Stata such as do-files, ado-files, help files, and CSV (`*.csv`) files, you should use the `unicode translate` command; see [D] **unicode translate**. If you wish to convert individual strings or string variables in your dataset, use the `ustrfrom()` and `ustrto()` functions.

Syntax

`unicode convertfile` *srcfilename destfilename* [, *options*]

srcfilename is a text file that is to be converted from a given encoding and *destfilename* is the destination text file that will use a different encoding.

options	Description
`srcencoding(`[*string*]`)`	encoding of the source file; UTF-8 if not specified
`dstencoding(`[*string*]`)`	encoding of the destination file; UTF-8 if not specified
`srccallback(`*method*`)`	what to do if source file contains invalid byte sequence(s)
`dstcallback(`*method*`)`	what to do if destination encoding does not support characters in the source file
`replace`	replace the destination file if it exists

method	Description
`stop`	specify that `unicode convertfile` stop with an error if an invalid character is encountered; the default
`skip`	specify that `unicode convertfile` skip invalid characters
`substitute`	specify that `unicode convertfile` substitute invalid characters with the destination encoding's substitute character during conversion; the substitute character for Unicode encodings is `\ufffd`
`escape`	specify that `unicode convertfile` replace any Unicode characters not supported in the destination encoding with an escaped string of the hex value of the Unicode code point. The string is in 4-hex-digit form `\uhhhh` for a code point less than or equal to `\uffff`. The string is in 8-hex-digit form `\Uhhhhhhhh` for code points greater than `\uffff`. `escape` may only be specified when converting from a Unicode encoding such as UTF-8.

Options

`srcencoding(`[*string*]`)` specifies the source file encoding. See `help encodings` for a list of common encodings and advice on choosing an encoding.

`dstencoding(`[*string*]`)` specifies the destination file encoding. See `help encodings` for a list of common encodings and advice on choosing an encoding.

`srccallback(`*method*`)` specifies the method for handling characters in the source file that cannot be converted.

`dstcallback(`*method*`)` specifies the method for handling characters that are not supported in the destination encoding.

`replace` permits `unicode convertfile` to overwrite an existing destination file.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Conversion between encodings

`unicode convertfile` is a utility to convert strings from one encoding to another. Encoding is the method by which text is stored in a computer. It maps a character to a nonnegative integer, called a code point, and then maps that integer to a single byte or a sequence of bytes. Common encodings are ASCII, UTF-8, and UTF-16. Stata uses UTF-8 encoding for storing text. Unless otherwise noted, the terms "Unicode string" and "Unicode character" in Stata refer to a UTF-8 encoded Unicode string or character. For more information about encodings, see [U] **12.4.2.3 Encodings**. See `help encodings` for a list of common encodings, and see [D] **unicode encoding** for a utility to find all available encodings.

If you are using `unicode convertfile` to convert a file to UTF-8 format, the string encoding using by Stata, you only need to specify the encoding of the source file. By default, UTF-8 is selected as the encoding for the destination file. You can also use `unicode convertfile` to convert files from UTF-8 encoding to another encoding. Although conversion to or from UTF-8 is the most common usage, you can use `unicode convertfile` to convert files between any pair of encodings.

Be aware that some characters may not be shared across encodings. The next section explains options for dealing with unsupported characters.

Invalid and unsupported characters

Unsupported characters generally occur in two ways: the bytes used to encode a character in the source encoding are not valid in the destination encoding such as UTF-8 (called an invalid sequence); or the character from the source encoding does not exist in the destination encoding.

It is common to encounter inconvertible characters when converting from a Unicode encoding such as UTF-8 to some other encoding. UTF-8 supports more than 100,000 characters. Depending on the characters in your file and the destination encoding you select, it is possible that not all characters will be supported. For example, ASCII only supports 128 characters, so all Unicode characters with code points greater than 127 are unsupported in ASCII encoding.

Examples

Convert file from Latin1 encoding to UTF-8 encoding

```
. unicode convertfile data.csv data_utf8.csv, srcencoding(ISO-8859-1)
```

Convert file from UTF-32 encoding to UTF-16 encoding, skipping any invalid sequences in the source file

```
. unicode convertfile utf32file.txt utf16file.txt, srcencoding(UTF-32)
> dstencoding(UTF-16) srccallback(skip)
```

Also see

[D] **unicode** — Unicode utilities

[D] **unicode translate** — Translate files to Unicode

[U] 12.4.2 Handling Unicode strings

[U] 12.4.2.6 Advice for users of Stata 13 and earlier