NAME

catdoc − reads MS-Word file and puts its content as plain text on standard output

SYNOPSIS

catdoc [**-vlu8btawx**] [**-m** *number*] [**-s** *charset*] [**-d** *charset*] [**-f** *output-format*] *file*

DESCRIPTION

catdoc behaves much like **cat**(1) but it reads MS-Word file and produces human-readable text on standard output. Optionally it can use **latex**(1) escape sequenses for characters which have specail meaning for LaTeX. It also makes some effort to recognize MS-Word tables, although it never tries to write correct headers for LaTeX tabular environment. Additional output formats, such is HTML can be easily defined.

catdoc doesn't attempt to extract formatting information other than tables from MS-Word document, so different output modes means mainly that different charachers should be escaped and different ways used to represent characters, missing from output charset. See CHARACTER SUBSTITUTION below

catdoc uses internal **unicode**(4) representation of text, so it is able to convert texts when charset in source document doesn't match charset on target system. See CHARACTER SETS below.

If no file names supplied, **catdoc** processes its standard input unless it is terminal. It is unlikely that somebody could type Word document from keyboard, so if **catdoc** invoked without arguments and stdin is not redirected, it prints brief usage message and exits. Processing of standard input (even among other files) can be forced using dash '-' as file name.

By default, **catdoc** wraps lines which are more than 72 chars long and separates paragraphs by blank lines. This behavoir can be turned of by **-w** switch. In *wide* mode **catdoc prints each paragraph as one long line, suitable for import into** word processors which perform word wrapping theirselves.

OPTIONS

-a - shortcut for -f ascii. Produces ASCII text as output. Separates table columns with TAB

-b - process broken MS-Word file. Normally, **catdoc checks if first 8 bytes** of file is Microsoft OLE signature. If so, it processes file, otherwise it just copies it to stdin. It is intended to use **catdoc** as filter for viewing all files with *.doc* extension.

-d*charset*
- specifies destination charset name. Charset file has format described in CHARACTER SETS below and should have **.txt** extension and reside in **catdoc library directory (normally /usr/local/lib/catdoc).**

-f *format*
- specifies output format as described in CHARACTER SUBSTITUTION below. **catdoc** comes with two output formats - ascii and tex. You can add your own if you wish.

-l Causes **catdoc** to list names of available charsets to the stdout and exit successfully.

-m*number*
Specifies right margin for text (default 72). **-m 0** is equivalent to **-w**

-s*charset*
Specifies source charset. (one used in Word document), if Word document doesn't contain UTF-16 text.

-t - shortcut for **-f tex**
converts all printable chars, which have special meaning for **LaTeX**(1) into appropriate control sequences. Separates table columns by **&.**

-u - declares that Word document contain UNICODE (UTF-16) represntation of text (as some Word-97 documents). If catdoc fails to correct Word document with default charset, try this option.

-8 - declares is Word document is 8 bit. Just in case that catdoc
recognizes file format incorrectly.

-w disables word wrapping. By default **catdoc** output is splitted into lines not longer than 72 (or number, specified by -m option) characters and paragraphs are separated by blank line. With this option each paragraph is one long line.

-x causes catdoc to output unknown UNICODE characher as \xNNNN, instead of question marks.

-v causes catdoc to print some useless information about word document structure to stdout before actual start of text.

CHARACTER SETS

When processing MS-Word file **catdoc** uses information about two character sets, typically different
- input and output. They are stored in plain text files in **catdoc** library directory. Character set files should contain two whitespace-separated hexadecimal numbers - 8-bit code in character set and 16-bit unicode code. Anything from hash mark to end of line is ignored, as well as blank lines.

catdoc distribution includes some of these character sets. Additional character set definitions, directly usable by **catdoc** can be obtained from ftp.unicode.org. Charset files have **.txt** suffix, which shouldn't be specified in command-line or configuration files.

Note that **catdoc** is distributed with Cyrillic charsets as default. If you are not Russian, you probably don't want it, an should reconfigure catdoc at compile time or in runtime configuration file.

When dealing with documents with charsets other than default, remember that Microsoft never uses ISO charsets. While letters in, say cp1252 are at the same position as in ISO-8859-1, some punctuation signs would be lost, if you specify ISO-8859-1 as input charset. If you use cp1252, catdoc would deal with those signs as described in CHARACTER SUBSTITUTION below.

CHARACTER SUBSTITUTION

catdoc converts MS-Word file into following internal unicode representation:

1. Paragraphs are separated by ASCII Line Feed symbol (0x000A)

2. Table cells within row are separated by ASCII Field Separator symbol
(0x001C)

3. Table rows are separated by ASCII Record Separator (0x001E)

4. All printable characters, including whitespace are represented with their
respective UNICODE codes.

This UNICODE representation is subsequentely converted into 8-bit text in target character set using following four-step algorithm:

1. List of special characters is searched for given unicode character.
If found, then appropriate multi-character sequence is output instead of character.

2. If there is an equivalent in target character set, it is output.

3. Otherwise, replacement list is searched and, if there is multi-character
substitution for this UNICODE char, it is output.

4. If all above fails, "Unknown char" symbol (question mark) is output.

Lists of special characters and list of substitution are character set-independent, becouse special chars should be escaped regardless of their existense in target character set (usially, they are parts of US-ASCII, and therefore exist in any character set) and replacement list is searched only for those characters, which are not found in target character set.

These lists are stored in **catdoc** library directory in files with prefix of format name. These files have following format:

Each line can be either comment (starting with hash mark) or contain hexadecimal UNICODE value, separated by whitespace from string, which would be substituted instead of it. If string contain no whitespace it can be used as is, otherwise it should be enclosed in single or double quotes. Usial backslash sequences like *'\n','\t'* can be used in these string.

RUNTIME CONFIGURATION

Upon startup catdoc reads its system-wide configuration file (**catdocrc in catdoc** library directory) and then user-specific configuration file **${HOME}/.catdocrc.**

These files can contain following directives:

source_charset = *charset-name*
: Sets default source charset, which would be used if no **-s** option specified. Consult configuration of nearby windows workstation to find one you need.

target_charset = *charset-name*
: Sets default output charset. You probably know, which one you use.

charset_path = *directory-list*
: colon-separated list of directories, which are searched for charset files. This allows you to install additional charsets in your home directory.

map_path = *directory-list*
: colon-separated list of directories, which are searched for special character map and replacement map.

format = *format name*
: Output format which would be used by default. **catdoc** comes with two formats - **ascii** and **tex** but nothing prevents you from writing your own format (set two map files - special character map and replacement map).

unknown_char = *character specification*
: sets characher to output instead of unknown unicode character (default '?') Character specification can have one of two form - character enclosed in single quotes or hexadecimal code.

BUGS

Can produce garbage, if file contain embedded illustrations. Doesn't handle fast-saves properly. Prints footnotes as separate paragraphs at the end of file, instead of producing correct latex commands. Cannot distinguish between empty table cell and end of table row.

SEE ALSO

xls2csv(1), **cat**(1), **strings**(1), **utf**(4), **unicode**(4)

AUTHOR

V.B.Wagner <vitus@ice.ru>