NAME

docbook2man − Convert DocBook to man pages

SYNOPSIS

docbook2man [*options*] *xml-document*

DESCRIPTION

docbook2man converts the given DocBook XML document into man pages. By default, the man pages will be output to the current directory.

Only the `refentry` content in the DocBook document is converted. (To convert content outside of a `refentry`, stylesheet customization is required. See the docbook2X package for details.)

The **docbook2man** command is a wrapper script for a two-step conversion process. See the section "CONVERSION PROCESS" below for details.

OPTIONS

The available options are essentially the union of the options from **db2x_xsltproc**(1) and **db2x_manxml**(1).

Some commonly-used options are listed below:

`--encoding=`*encoding*
: Sets the character encoding of the output.

`--string-param` *parameter***`=`***value*
: Sets a stylesheet parameter (options that affect how the output looks). See "Stylesheet parameters" below for the parameters that can be set.

`--sgml`
: Accept an SGML source document as input instead of XML.

`--solinks`
: Make stub pages for alternate names for an output man page.

STYLESHEET PARAMETERS

`uppercase-headings`
: **Brief**. Make headings uppercase?

  Default setting. `1` (boolean true)

  Headings in man page content should be or should not be uppercased.

`manvolnum-cite-numeral-only`
: **Brief**. Man page section citation should use only the number

  Default setting. `1` (boolean true)

  When citing other man pages, the man-page section is either given as is, or has the letters stripped from it, citing only the number of the section (e.g. section `3x` becomes `3`). This option specifies which style.

`quotes-on-literals`
: **Brief**. Display quotes on `literal` elements?

  Default setting. `0` (boolean false)

  If true, render `literal` elements with quotes around them.

`show-comments`
: **Brief**. Display `comment` elements?

  Default setting. `1` (boolean true)

If true, comments will be displayed, otherwise they are suppressed. Comments here refers to the `comment` element, which will be renamed `remark` in DocBook V4.0, not XML comments (<-- like this -->) which are unavailable.

`function-parens`

Brief. Generate parentheses after a function?

Default setting. `0` (boolean false)

If true, the formatting of a `<function>` element will include generated parenthesis.

`xref-on-link`

Brief. Should `link` generate a cross-reference?

Default setting. `1` (boolean true)

Man pages cannot render the hypertext links created by `link`. If this option is set, then the stylesheet renders a cross reference to the target of the link. (This may reduce clutter). Otherwise, only the content of the `link` is rendered and the actual link itself is ignored.

`header-3`

Brief. Third header text

Default setting. (blank)

Specifies the text of the third header of a man page, typically the date for the man page. If empty, the `date` content for the `refentry` is used.

`header-4`

Brief. Fourth header text

Default setting. (blank)

Specifies the text of the fourth header of a man page. If empty, the `refmiscinfo` content for the `refentry` is used.

`header-5`

Brief. Fifth header text

Default setting. (blank)

Specifies the text of the fifth header of a man page. If empty, the 'manual name', that is, the title of the `book` or `reference` container is used.

`default-manpage-section`

Brief. Default man page section

Default setting. `1`

The source document usually indicates the sections that each man page should belong to (with `manvolnum` in `refmeta`). In case the source document does not indicate man-page sections, this option specifies the default.

`custom-localization-file`

Brief. URI of XML document containing custom localization data

Default setting. (blank)

This parameter specifies the URI of a XML document that describes text translations (and other

locale-specific information) that is needed by the stylesheet to process the DocBook document.

The text translations pointed to by this parameter always override the default text translations (from the internal parameter `localization-file`). If a particular translation is not present here, the corresponding default translation is used as a fallback.

This parameter is primarily for changing certain punctuation characters used in formatting the source document. The settings for punctuation characters are often specific to the source document, but can also be dependent on the locale.

To not use custom text translations, leave this parameter as the empty string.

`custom-l10n-data`

Brief. XML document containing custom localization data

Default setting. `document($custom-localization-file)`

This parameter specifies the XML document that describes text translations (and other locale-specific information) that is needed by the stylesheet to process the DocBook document.

This parameter is internal to the stylesheet. To point to an external XML document with a URI or a file name, you should use the `custom-localization-file` parameter instead.

However, inside a custom stylesheet (*not on the command-line*) this paramter can be set to the XPath expression `document('')`, which will cause the custom translations directly embedded inside the custom stylesheet to be read.

`author-othername-in-middle`

Brief. Is `othername` in `author` a middle name?

Default setting. `1`

If true, the `othername` of an `author` appears between the `firstname` and `surname`. Otherwise, `othername` is suppressed.

EXAMPLES

```
$ docbook2man --solinks manpages.xml
$ docbook2man --solinks --encoding=utf-8//TRANSLIT manpages.xml
$ docbook2man --string-param header-4="Free Recode 3.6" document.xml
```

CONVERSION PROCESS

Converting to man pages

DocBook documents are converted to man pages in two steps:

1. The DocBook source is converted by a XSLT stylesheet into an intermediate XML format, Man-XML.

   Man-XML is simpler than DocBook and closer to the man page format; it is intended to make the stylesheets' job easier.

   The stylesheet for this purpose is in *`xslt/man/docbook.xsl`*. For portability, it should always be referred to by the following URI:

   http://docbook2x.sourceforge.net/latest/xslt/man/docbook.xsl

   Run this stylesheet with **db2x_xsltproc**(1).

Customizing. You can also customize the output by creating your own XSLT stylesheet — changing parameters or adding new templates — and importing *xslt/man/docbook.xsl*.

2. Man-XML is converted to the actual man pages by **db2x_manxml**(1).

The **docbook2man** command does both steps automatically, but if any problems occur, you can see the errors more clearly if you do each step separately:

```
$ db2x_xsltproc -s man mydoc.xml -o mydoc.mxml
$ db2x_manxml mydoc.mxml
```

Options to the conversion stylesheet are described in the man-pages stylesheets reference.

Pure XSLT conversion. An alternative to the **db2x_manxml** Perl script is the XSLT stylesheet in *xslt/backend/db2x_manxml.xsl*. This stylesheet performs a similar function of converting Man-XML to actual man pages. It is useful if you desire a pure XSLT solution to man-page conversion. Of course, the quality of the conversion using this stylesheet will never be as good as the Perl **db2x_manxml**, and it runs slower. In particular, the pure XSLT version currently does not support tables in man pages, but its Perl counterpart does.

Character set conversion

When translating XML to legacy ASCII-based formats with poor support for Unicode, such as man pages and Texinfo, there is always the problem that Unicode characters in the source document also have to be translated somehow.

A straightforward character set conversion from Unicode does not suffice, because the target character set, usually US-ASCII or ISO Latin-1, do not contain common characters such as dashes and directional quotation marks that are widely used in XML documents. But document formatters (man and Texinfo) allow such characters to be entered by a markup escape: for example, `\(lq` for the left directional quote “. And if a markup-level escape is not available, an ASCII transliteration might be used: for example, using the ASCII less-than sign < for the angle quotation mark ⟨.

So the Unicode character problem can be solved in two steps:

1. **utf8trans**(1), a program included in docbook2X, maps Unicode characters to markup-level escapes or transliterations.

   Since there is not necessarily a fixed, official mapping of Unicode characters, **utf8trans** can read in user-modifiable character mappings expressed in text files and apply them. (Unlike most character set converters.)

   In *charmaps/man/roff.charmap* and *charmaps/man/texi.charmap* are character maps that may be used for man-page and Texinfo conversion. The programs **db2x_manxml**(1) and **db2x_texixml**(1) will apply these character maps, or another character map specified by the user, automatically.

2. The rest of the Unicode text is converted to some other character set (encoding). For example, a French document with accented characters (such as é) might be converted to ISO Latin 1.

   This step is applied after **utf8trans** character mapping, using the **iconv**(1) encoding conversion tool. Both **db2x_manxml**(1) and **db2x_texixml**(1) can call **iconv**(1) automatically when producing their output.

FILES

/usr/local/share/docbook2X/xslt/man/docbook.xsl
/usr/local/share/docbook2X/xslt/backend/db2x_manxml.xsl
/usr/local/share/docbook2X/xslt/catalog.xml
/usr/local/share/docbook2X/charmaps/roff.charmap
/usr/local/share/docbook2X/charmaps/roff.charmap.xml

The above files are distributed and installed by the docbook2X package.

NOTES

The **docbook2man** or the **docbook2texi** command described in this manual page come from the docbook2X package. It should not be confused with the command of the same name from the obsoleted docbook-utils package.

LIMITATIONS

- Internally there is one long pipeline of programs which your document goes through. If any segment of the pipeline fails (even trivially, like from mistyped program options), the resulting errors can be difficult to decipher — in this case, try running the components of docbook2X separately.

AUTHOR

Steve Cheng <stevecheng@users.sourceforge.net>.

SEE ALSO

db2x_xsltproc(1), **db2x_manxml**(1), **utf8trans**(1)

The docbook2X manual (in Texinfo or HTML format) fully describes how to convert DocBook to man pages and Texinfo.

Up-to-date information about this program can be found at the docbook2X Web site ⟨http://docbook2x.sourceforge.net/⟩.

NAME

docbook2texi − Convert DocBook to Texinfo

SYNOPSIS

docbook2texi [*options*] *xml-document*

DESCRIPTION

docbook2texi converts the given DocBook XML document into one or more Texinfo documents. By default, these Texinfo documents will be output to the current directory.

The **docbook2texi** command is a wrapper script for a two-step conversion process. See the section "CONVERSION PROCESS" below for details.

OPTIONS

The available options are essentially the union of the options for **db2x_xsltproc**(1) and **db2x_texixml**(1).

Some commonly-used options are listed below:

--encoding=*encoding*

Sets the character encoding of the output.

--string-param *parameter***=***value*

Sets a stylesheet parameter (options that affect how the output looks). See "Stylesheet parameters" below for the parameters that can be set.

--sgml

Accept an SGML source document as input instead of XML.

STYLESHEET PARAMETERS

`captions-display-as-headings`

Brief. Use heading markup for minor captions?

Default setting. `0` (boolean false)

If true, `title` content in some (formal) objects are rendered with the Texinfo `@heading` commands.

If false, captions are rendered as an emphasized paragraph.

`links-use-pxref`

Brief. Translate `link` using `@pxref`

Default setting. `1` (boolean true)

If true, `link` is translated with the hypertext followed by the cross reference in parentheses.

Otherwise, the hypertext content serves as the cross-reference name marked up using `@ref`. Typically info displays this contruct badly.

`explicit-node-names`

Brief. Insist on manually constructed Texinfo node names

Default setting. `0` (boolean false)

Elements in the source document can influence the Texinfo node name generation specifying either a `xreflabel`, or for the sectioning elements, a `title` with `role='texinfo-node'` in the `*info` container.

However, for the majority of source documents, explicit Texinfo node names are not available, and the stylesheet tries to generate a reasonable one instead, e.g. from the normal title of an element. The generated name may not be optimal. If this option is set and the stylesheet needs to generate a

name, a warning is emitted and **generate-id** is always used for the name.

When the hashtable extension is not available, the stylesheet cannot check for node name collisions, and in this case, setting this option and using explicit node names are recommended.

This option is not set (i.e. false) by default.
Note

The absolute fallback for generating node names is using the XSLT function **generate-id**, and the stylesheet always emits a warning in this case regardless of the setting of explicit-node-names.

show-comments
Brief. Display comment elements?

Default setting. 1 (boolean true)

If true, comments will be displayed, otherwise they are suppressed. Comments here refers to the comment element, which will be renamed remark in DocBook V4.0, not XML comments (<-- like this -->) which are unavailable.

funcsynopsis-decoration
Brief. Decorate elements of a FuncSynopsis?

Default setting. 1 (boolean true)

If true, elements of the FuncSynopsis will be decorated (e.g. bold or italic). The decoration is controlled by functions that can be redefined in a customization layer.

function-parens
Brief. Generate parentheses after a function?

Default setting. 0 (boolean false)

If true, the formatting of a <function> element will include generated parenthesis.

refentry-display-name
Brief. Output NAME header before 'RefName'(s)?

Default setting. 1 (boolean true)

If true, a "NAME" section title is output before the list of 'RefName's.

manvolnum-in-xref
Brief. Output manvolnum as part of refentry cross-reference?

Default setting. 1 (boolean true)

if true, the manvolnum is used when cross-referencing refentrys, either with xref or citerefentry.

prefer-textobjects
Brief. Prefer textobject over imageobject?

Default setting. 1 (boolean true)

If true, the textobject in a mediaobject is preferred over any imageobject.

(Of course, for output formats other than Texinfo, you usually want to prefer the `imageobject`, but Info is a text-only format.)

In addition to the values true and false, this parameter may be set to `2` to indicate that both the text and the images should be output. You may want to do this because some Texinfo viewers can read images. Note that the Texinfo `@image` command has its own mechanism for switching between text and image output — but we do not use this here.

The default is true.

`semantic-decorations`

Brief. Use Texinfo semantic inline markup?

Default setting. `1` (boolean true)

If true, the semantic inline markup of DocBook is translated into (the closest) Texinfo equivalent. This is the default.

However, because the Info format is limited to plain text, the semantic inline markup is often distinguished by using explicit quotes, which may not look good. You can set this option to false to suppress these. (For finer control over the inline formatting, you can use your own stylesheet.)

`custom-localization-file`

Brief. URI of XML document containing custom localization data

Default setting. (blank)

This parameter specifies the URI of a XML document that describes text translations (and other locale-specific information) that is needed by the stylesheet to process the DocBook document.

The text translations pointed to by this parameter always override the default text translations (from the internal parameter `localization-file`). If a particular translation is not present here, the corresponding default translation is used as a fallback.

This parameter is primarily for changing certain punctuation characters used in formatting the source document. The settings for punctuation characters are often specific to the source document, but can also be dependent on the locale.

To not use custom text translations, leave this parameter as the empty string.

`custom-l10n-data`

Brief. XML document containing custom localization data

Default setting. `document($custom-localization-file)`

This parameter specifies the XML document that describes text translations (and other locale-specific information) that is needed by the stylesheet to process the DocBook document.

This parameter is internal to the stylesheet. To point to an external XML document with a URI or a file name, you should use the `custom-localization-file` parameter instead.

However, inside a custom stylesheet (*not on the command-line*) this paramter can be set to the XPath expression `document('')`, which will cause the custom translations directly embedded inside the custom stylesheet to be read.

`author-othername-in-middle`
Brief. Is `othername` in `author` a middle name?

Default setting. `1`

If true, the `othername` of an `author` appears between the `firstname` and `surname`. Otherwise, `othername` is suppressed.

`output-file`
Brief. Name of the Info file

Default setting. (blank)

This parameter specifies the name of the final Info file, overriding the setting in the document itself and the automatic selection in the stylesheet. If the document is a `set`, this parameter has no effect.
Important

Do *not* include the `.info` extension in the name.

(Note that this parameter has nothing to do with the name of the *Texi-XML output* by the XSLT processor you are running this stylesheet from.)

`directory-category`
Brief. The categorization of the document in the Info directory

Default setting. (blank)

This is set to the category that the document should go under in the Info directory of installed Info files. For example, `General Commands`.
Note

Categories may also be set directly in the source document. But if this parameter is not empty, then it always overrides the setting in the source document.

`directory-description`
Brief. The description of the document in the Info directory

Default setting. (blank)

This is a short description of the document that appears in the Info directory of installed Info files. For example, `An Interactive Plotting Program.`
Note

Menu descriptions may also be set directly in the source document. But if this parameter is not empty, then it always overrides the setting in the source document.

`index-category`
Brief. The Texinfo index to use

Default setting. `cp`

The Texinfo index for `indexterm` and `index` is specified using the `role` attribute. If the above elements do not have a `role`, then the default specified by this parameter is used.

The predefined indices are:

`c`, `cp` Concept index

`f`, `fn` Function index

`v`, `vr` Variable index

`k`, `ky` Keystroke index

`p`, `pg` Program index

`d`, `tp` Data type index

User-defined indices are not yet supported.

`qanda-defaultlabel`
Brief. Sets the default for defaultlabel on QandASet.

Default setting.

If no defaultlabel attribute is specified on a QandASet, this value is used. It must be one of the legal values for the defaultlabel attribute.

`qandaset-generate-toc`
Brief. Is a Table of Contents created for QandASets?

Default setting.

If true, a ToC is constructed for QandASets.

EXAMPLES

```
$ docbook2texi tdg.xml
$ docbook2texi --encoding=utf-8//TRANSLIT tdg.xml
$ docbook2texi --string-param semantic-decorations=0 tdg.xml
```

CONVERSION PROCESS

Converting to Texinfo

DocBook documents are converted to Texinfo in two steps:

1. The DocBook source is converted by a XSLT stylesheet into an intermediate XML format, Texi-XML.

   Texi-XML is simpler than DocBook and closer to the Texinfo format; it is intended to make the stylesheets' job easier.

   The stylesheet for this purpose is in *`xslt/texi/docbook.xsl`*. For portability, it should always be referred to by the following URI:

   http://docbook2x.sourceforge.net/latest/xslt/texi/docbook.xsl

   Run this stylesheet with **db2x_xsltproc**(1).

   Customizing. You can also customize the output by creating your own XSLT stylesheet — changing parameters or adding new templates — and importing *`xslt/texi/docbook.xsl`*.

2. Texi-XML is converted to the actual Texinfo files by **db2x_texixml**(1).

The **docbook2texi** command does both steps automatically, but if any problems occur, you can see the errors more clearly if you do each step separately:

```
$ db2x_xsltproc -s texi mydoc.xml -o mydoc.txml
$ db2x_texixml mydoc.txml
```

Options to the conversion stylesheet are described in the Texinfo stylesheets reference.

Character set conversion

When translating XML to legacy ASCII-based formats with poor support for Unicode, such as man pages and Texinfo, there is always the problem that Unicode characters in the source document also have to be translated somehow.

A straightforward character set conversion from Unicode does not suffice, because the target character set, usually US-ASCII or ISO Latin-1, do not contain common characters such as dashes and directional quotation marks that are widely used in XML documents. But document formatters (man and Texinfo) allow such characters to be entered by a markup escape: for example, `\(lq` for the left directional quote “. And if a markup-level escape is not available, an ASCII transliteration might be used: for example, using the ASCII less-than sign < for the angle quotation mark ⟨.

So the Unicode character problem can be solved in two steps:

1. **utf8trans**(1), a program included in docbook2X, maps Unicode characters to markup-level escapes or transliterations.

   Since there is not necessarily a fixed, official mapping of Unicode characters, **utf8trans** can read in user-modifiable character mappings expressed in text files and apply them. (Unlike most character set converters.)

   In *charmaps/man/roff.charmap* and *charmaps/man/texi.charmap* are character maps that may be used for man-page and Texinfo conversion. The programs **db2x_manxml**(1) and **db2x_texixml**(1) will apply these character maps, or another character map specified by the user, automatically.

2. The rest of the Unicode text is converted to some other character set (encoding). For example, a French document with accented characters (such as é) might be converted to ISO Latin 1.

   This step is applied after **utf8trans** character mapping, using the **iconv**(1) encoding conversion tool. Both **db2x_manxml**(1) and **db2x_texixml**(1) can call **iconv**(1) automatically when producing their output.

FILES

/usr/local/share/docbook2X/xslt/texi/docbook.xsl
/usr/local/share/docbook2X/xslt/backend/db2x_texixml.xsl
/usr/local/share/docbook2X/xslt/catalog.xml
/usr/local/share/docbook2X/charmaps/texi.charmap.xml
/usr/local/share/docbook2X/charmaps/texi.charmap.xml

The above files are distributed and installed by the docbook2X package.

NOTES

The **docbook2man** or the **docbook2texi** command described in this manual page come from the docbook2X package. It should not be confused with the command of the same name from the obsoleted docbook-utils package.

LIMITATIONS

- Internally there is one long pipeline of programs which your document goes through. If any segment of the pipeline fails (even trivially, like from mistyped program options), the resulting errors can be difficult to decipher — in this case, try running the components of docbook2X separately.

AUTHOR

Steve Cheng <stevecheng@users.sourceforge.net>.

SEE ALSO

db2x_xsltproc(1), **db2x_texixml**(1), **utf8trans**(1)

The docbook2X manual (in Texinfo or HTML format) fully describes how to convert DocBook to man pages and Texinfo.

Up-to-date information about this program can be found at the docbook2X Web site ⟨`http://docbook2x.sourceforge.net/`⟩.

NAME

db2x_manxml − Make man pages from Man-XML

SYNOPSIS

db2x_manxml [*options*] [*xml-document*]

DESCRIPTION

db2x_manxml converts a Man-XML document into one or more man pages. They are written in the current directory.

If *xml-document* is not given, then the document to convert is read from standard input.

OPTIONS

`--encoding=`*encoding*

Select the character encoding used for the output files. The available encodings are those of **iconv**(1). The default encoding is `us-ascii`.

The XML source may contain characters that are not representable in the encoding that you select; in this case the program will bomb out during processing, and you should choose another encoding. (This is guaranteed not to happen with any Unicode encoding such as UTF-8, but unfortunately not everyone is able to process Unicode texts.)

If you are using GNU's version of **iconv**(1), you can affix `//TRANSLIT` to the end of the encoding name to attempt transliterations of any unconvertible characters in the output. Beware, however, that the really inconvertible characters will be turned into another of those damned question marks. (Aren't you sick of this?)

The suffix `//TRANSLIT` applied to a Unicode encoding — in particular, `utf-8//TRANSLIT` — means that the output files are to remain in Unicode, but markup-level character translations using **utf8trans** are still to be done. So in most cases, an English-language document, converted using **`--encoding=utf-8//TRANSLIT`** will actually end up as a US-ASCII document, but any untranslatable characters will remain as UTF-8 without any warning whatsoever. (Note: strictly speaking this is not "transliteration".) This method of conversion is a compromise over strict **`--encoding=us-ascii`** processing, which aborts if any untranslatable characters are encountered.

Note that man pages and Texinfo documents in non-ASCII encodings (including UTF-8) may not be portable to older (non-internationalized) systems, which is why the default value for this option is `us-ascii`.

To suppress any automatic character mapping or encoding conversion whatsoever, pass the option **`--encoding=utf-8`**.

`--list-files`

Write a list of all the output files to standard output, in addition to normal processing.

`--output-dir=`*dir*

Specify the directory where the output files are placed. The default is the current working directory.

This option is ignored if the output is to be written to standard output (triggered by the option **`--to-stdout`**).

`--to-stdout`

Write the output to standard output instead of to individual files.

If this option is used even when there are supposed to be multiple output documents, then everything is concatenated to standard output. But beware that most other programs will not accept this concatenated output.

This option is incompatible with **--list-files**, obviously.

--help
Show brief usage information and exit.

--version
Show version and exit.

Some man pages may be referenced under two or more names, instead of just one. For example, **strcpy**(3) and **strncpy**(3) often point to the same man page which describes the two functions together. Choose one of the following options to select how such man pages are to be generated:

--symlinks
For each of all the alternate names for a man page, erect symbolic links to the file that contains the real man page content.

--solinks
Generate stub pages (using `.so` roff requests) for the alternate names, pointing them to the real man page content.

--no-links
Do not make any alternative names available. The man page can only be referenced under its principal name.

This program uses certain other programs for its operation. If they are not in their default installed locations, then use the following options to set their location:

--utf8trans-program=*path*, **--utf8trans-map=***charmap*
Use the character map *charmap* with the **utf8trans**(1) program, included with docbook2X, found under *path*.

--iconv-program=*path*
The location of the **iconv**(1) program, used for encoding conversions.

NOTES

The man pages produced should be compatible with most troff implementations and other tools that process man pages. Some backwards-compatible **groff**(1) extensions are used to make the output look nicer.

AUTHOR

Steve Cheng <stevecheng@users.sourceforge.net>.

SEE ALSO

The docbook2X manual (in Texinfo or HTML format) fully describes how to convert DocBook to man pages and Texinfo.

Up-to-date information about this program can be found at the docbook2X Web site ⟨http://docbook2x.sourceforge.net/⟩.

The input to **db2x_manxml** is defined by the XML DTD present at *dtd/Man-XML* in the docbook2X distribution.

NAME

db2x_texixml − Make Texinfo files from Texi-XML

SYNOPSIS

db2x_texixml [options]... [*xml-document*]

DESCRIPTION

db2x_texixml converts a Texi-XML document into one or more Texinfo documents.

If *xml-document* is not given, then the document to convert comes from standard input.

The filenames of the Texinfo documents are determined by markup in the Texi-XML source. (If the filenames are not specified in the markup, then **db2x_texixml** attempts to deduce them from the name of the input file. However, the Texi-XML source should specify the filename, because it does not work when there are multiple output files or when the Texi-XML source comes from standard input.)

OPTIONS

`--encoding=`*encoding*

Select the character encoding used for the output files. The available encodings are those of **iconv**(1). The default encoding is `us-ascii`.

The XML source may contain characters that are not representable in the encoding that you select; in this case the program will bomb out during processing, and you should choose another encoding. (This is guaranteed not to happen with any Unicode encoding such as UTF-8, but unfortunately not everyone is able to process Unicode texts.)

If you are using GNU's version of **iconv**(1), you can affix `//TRANSLIT` to the end of the encoding name to attempt transliterations of any unconvertible characters in the output. Beware, however, that the really inconvertible characters will be turned into another of those damned question marks. (Aren't you sick of this?)

The suffix `//TRANSLIT` applied to a Unicode encoding — in particular, `utf-8//TRANSLIT` — means that the output files are to remain in Unicode, but markup-level character translations using **utf8trans** are still to be done. So in most cases, an English-language document, converted using **`--encoding=utf-8//TRANSLIT`** will actually end up as a US-ASCII document, but any untranslatable characters will remain as UTF-8 without any warning whatsoever. (Note: strictly speaking this is not "transliteration".) This method of conversion is a compromise over strict **`--encoding=us-ascii`** processing, which aborts if any untranslatable characters are encountered.

Note that man pages and Texinfo documents in non-ASCII encodings (including UTF-8) may not be portable to older (non-internationalized) systems, which is why the default value for this option is `us-ascii`.

To suppress any automatic character mapping or encoding conversion whatsoever, pass the option **`--encoding=utf-8`**.

`--list-files`

Write a list of all the output files to standard output, in addition to normal processing.

`--output-dir=`*dir*

Specify the directory where the output files are placed. The default is the current working directory.

This option is ignored if the output is to be written to standard output (triggered by the option **`--to-stdout`**).

--to-stdout
Write the output to standard output instead of to individual files.

If this option is used even when there are supposed to be multiple output documents, then everything is concatenated to standard output. But beware that most other programs will not accept this concatenated output.

This option is incompatible with **--list-files**, obviously.

--info
Pipe the Texinfo output to **makeinfo**(1), creating Info files directly instead of Texinfo files.

--plaintext
Pipe the Texinfo output to **makeinfo --no-headers**, thereby creating plain text files.

--help
Show brief usage information and exit.

--version
Show version and exit.

This program uses certain other programs for its operation. If they are not in their default installed locations, then use the following options to set their location:

--utf8trans-program=*path*, **--utf8trans-map=***charmap*
Use the character map *charmap* with the **utf8trans**(1) program, included with docbook2X, found under *path*.

--iconv-program=*path*
The location of the **iconv**(1) program, used for encoding conversions.

NOTES

Texinfo language compatibility. The Texinfo files generated by **db2x_texixml** sometimes require Texinfo version 4.7 (the latest version) to work properly. In particular:

- **db2x_texixml** relies on **makeinfo** to automatically add punctuation after a `@ref` if it it not already there. Otherwise the hyperlink will not work in the Info reader (although **makeinfo** will not emit any error).
- The new `@comma{}` command is used for commas (`,`) occurring inside argument lists to Texinfo commands, to disambiguate it from the comma used to separate different arguments. The only alternative otherwise would be to translate `,` to `.` which is obviously undesirable (but earlier docbook2X versions did this).

  If you cannot use version 4.7 of **makeinfo**, you can still use a **sed** script to perform manually the procedure just outlined.

Relation of Texi-XML with the XML output format of makeinfo. The Texi-XML format used by docbook2X is *different* and incompatible with the XML format generated by **makeinfo**(1) with its **--xml** option. This situation arose partly because the Texi-XML format of docbook2X was designed and implemented independently before the appearance of **makeinfo**'s XML format. Also Texi-XML is very much geared towards being *machine-generated from other XML formats*, while there seems to be no non-trivial applications of **makeinfo**'s XML format. So there is no reason at this point for docbook2X to adopt **makeinfo**'s XML format in lieu of Texi-XML.

BUGS

- Text wrapping in menus is utterly broken for non-ASCII text. It is probably also broken everywhere else in the output, but that would be **makeinfo**'s fault.
- **--list-files** might not work correctly with **--info**. Specifically, when the output Info file get too big, **makeinfo** will decide to split it into parts named *abc.info-1*, *abc.info-2*, *abc.info-3*, etc. **db2x_texixml** does not know exactly how many of these files there are, though you can just

do an **ls** to find out.

AUTHOR

Steve Cheng <stevecheng@users.sourceforge.net>.

SEE ALSO

The docbook2X manual (in Texinfo or HTML format) fully describes how to convert DocBook to man pages and Texinfo.

Up-to-date information about this program can be found at the docbook2X Web site ⟨http://docbook2x.sourceforge.net/⟩.

The input to **db2x_texixml** is defined by the XML DTD present at *dtd/Texi-XML* in the docbook2X distribution.

NAME

db2x_xsltproc − XSLT processor invocation wrapper

SYNOPSIS

db2x_xsltproc [*options*] *xml-document*

DESCRIPTION

db2x_xsltproc invokes the XSLT 1.0 processor for docbook2X.

This command applies the XSLT stylesheet (usually given by the **--stylesheet** option) to the XML document in the file *xml-document*. The result is written to standard output (unless changed with **--output**).

To read the source XML document from standard input, specify − as the input document.

OPTIONS

--version

Display the docbook2X version.

TRANSFORMATION OUTPUT OPTIONS

--output *file*, **-o** *file*

Write output to the given file (or URI), instead of standard output.

SOURCE DOCUMENT OPTIONS

--xinclude, **-I**

Process XInclude directives in the source document.

--sgml, **-S**

Indicate that the input document is SGML instead of XML. You need this set this option if *xml-document* is actually a SGML file.

SGML parsing is implemented by conversion to XML via **sgml2xml**(1) from the SP package (or **osx**(1) from the OpenSP package). All tag names in the SGML file will be normalized to lowercase (i.e. the **-xlower** option of **sgml2xml**(1) is used). ID attributes are available for the stylesheet (i.e. option **-xid**). In addition, any ISO SDATA entities used in the SGML document are automatically converted to their XML Unicode equivalents. (This is done by a **sed** filter.)

The encoding of the SGML document, if it is not `us-ascii`, must be specified with the standard SP environment variables: **SP_CHARSET_FIXED=1 SP_ENCODING=***encoding*. (Note that XML files specify their encoding with the XML declaration **<?xml version="1.0" encoding="***encoding***" ?>** at the top of the file.)

The above conversion options cannot be changed. If you desire different conversion options, you should invoke **sgml2xml**(1) manually, and then pass the results of that conversion to this program.

RETRIEVAL OPTIONS

--catalogs *catalog-files*, **-C** *catalog-files*

Specify additional XML catalogs to use for resolving Formal Public Identifiers or URIs. SGML catalogs are not supported.

These catalogs are *not* used for parsing an SGML document under the **--sgml** option. Use the environment variable **SGML_CATALOG_FILES** instead to specify the catalogs for parsing the SGML document.

--network, **-N**

db2x_xsltproc will normally refuse to load external resources from the network, for security reasons. If you do want to load from the network, set this option.

Usually you want to have installed locally the relevent DTDs and other files, and set up catalogs for them, rather than load them automatically from the network.

STYLESHEET OPTIONS

`--stylesheet` *file*, **`-s`** *file*

Specify the filename (or URI) of the stylesheet to use. The special values `man` and `texi` are accepted as abbreviations, to specify that *xml-document* is in DocBook and should be converted to man pages or Texinfo (respectively).

`--param` *name*`=`*expr*, **`-p`** *name*`=`*expr*

Add or modify a parameter to the stylesheet. *name* is a XSLT parameter name, and *expr* is an XPath expression that evaluates to the desired value for the parameter. (This means that strings must be quoted, *in addition* to the usual quoting of shell arguments; use **`--string-param`** to avoid this.)

`--string-param` *name*`=`*string*, **`-g`** *name*`=`*string*

Add or modify a string-valued parameter to the stylesheet.

The string must be encoded in UTF-8 (regardless of the locale character encoding).

DEBUGGING AND PROFILING

`--debug`, **`-d`**

Display, to standard error, logs of what is happening during the XSL transformation.

`--nesting-limit` *n*, **`-D`** *n*

Change the maximum number of nested calls to XSL templates, used to detect potential infinite loops. If not specified, the limit is 500 (libxslt's default).

`--profile`, **`-P`**

Display profile information: the total number of calls to each template in the stylesheet and the time taken for each. This information is output to standard error.

`--xslt-processor` *processor*, **`-X`** *processor*

Select the underlying XSLT processor used. The possible choices for *processor* are: `libxslt`, `saxon`, `xalan-j`.

The default processor is whatever was set when docbook2X was built. libxslt is recommended (because it is lean and fast), but SAXON is much more robust and would be more helpful when debugging stylesheets.

All the processors have XML catalogs support enabled. (docbook2X requires it.) But note that not all the options above work with processors other than the libxslt one.

ENVIRONMENT

XML_CATALOG_FILES

Specify XML Catalogs. If not specified, the standard catalog (*`/etc/xml/catalog`*) is loaded, if available.

DB2X_XSLT_PROCESSOR

Specify the XSLT processor to use. The effect is the same as the **`--xslt-processor`** option. The primary use of this variable is to allow you to quickly test different XSLT processors without having to add **`--xslt-processor`** to every script or make file in your documentation build system.

CONFORMING TO

XML Stylesheet Language – Transformations (XSLT), version 1.0 ⟨`http://www.w3.org/TR/xslt`⟩, a W3C Recommendation.

NOTES

In its earlier versions (< 0.8.4), docbook2X required XSLT extensions to run, and **db2x_xsltproc** was a special libxslt-based processor that had these extensions compiled-in. When the requirement for XSLT extensions was dropped, **db2x_xsltproc** became a Perl script which translates the options to **db2x_xsltproc** to conform to the format accepted by the stock **xsltproc**(1) which comes with libxslt.

The prime reason for the existence of this script is backward compatibility with any scripts or make files that invoke docbook2X. However, it also became easy to add in support for invoking other XSLT processors with a unified command-line interface. Indeed, there is nothing special in this script to docbook2X, or even to DocBook, and it may be used for running other sorts of stylesheets if you desire. Certainly the author prefers using this command, because its invocation format is sane and is easy to use. (e.g. no typing long class names for the Java-based processors!)

AUTHOR

Steve Cheng <stevecheng@users.sourceforge.net>.

SEE ALSO

The docbook2X manual (in Texinfo or HTML format) fully describes how to convert DocBook to man pages and Texinfo.

Up-to-date information about this program can be found at the docbook2X Web site ⟨http://docbook2x.sourceforge.net/⟩.

You may wish to consult the documentation that comes with libxslt, SAXON, or Xalan. The W3C XSLT 1.0 specification would be useful for writing stylesheets.

NAME

sgml2xml-isoent − Convert SGML to XML with support for ISO entities

SYNOPSIS

sgml2xml-isoent [*sgml-document*]

DESCRIPTION

sgml2xml-isoent converts an SGML document to XML, with support for the ISO entities. This is done by using **sgml2xml**(1) from the SP package (or **osx**(1) from the OpenSP package), and the declaration for the XML version of the ISO entities is added to the output. This means that the output of this conversion should work as-is with any XML tool.

This program is often used for processing SGML DocBook documents with XML-based tools. In particular, **db2x_xsltproc**(1) calls this program as part of its **`--sgml`** option. On the other hand, it is probably not helpful for migrating a source SGML text file to XML, since the conversion mangles the original formatting.

Since the XML version of the ISO entities are referred to directly, not via a DTD, this tool also works with document types other than DocBook.

NOTES

The ISO entities are referred using the public identifiers `ISO 8879:1986//ENTITIES//...//EN//XML`. The catalogs used when parsing the converted document should resolve these entities to the appropriate place (on the local filesystem). If the entities are not resolved in the catalog, then the fallback is to get the entity files from the `http://www.docbook.org/` Web site.

AUTHOR

Steve Cheng <`stevecheng@users.sourceforge.net`>.

SEE ALSO

sgml2xml(1), **osx**(1)

NAME

utf8trans − Transliterate UTF-8 characters according to a table

SYNOPSIS

utf8trans *charmap* [*file*]...

DESCRIPTION

utf8trans transliterates characters in the specified files (or standard input, if they are not specified) and writes the output to standard output. All input and output is in the UTF-8 encoding.

This program is usually used to render characters in Unicode text files as some markup escapes or ASCII transliterations. (It is not intended for general charset conversions.) It provides functionality similar to the character maps in XSLT 2.0 (XML Stylesheet Language – Transformations, version 2.0).

OPTIONS

`-m`, **`--modify`**
: Modifies the given files in-place with their transliterated output, instead of sending it to standard output.

  This option is useful for efficient transliteration of many files at once.

`--help`
: Show brief usage information and exit.

`--version`
: Show version and exit.

USAGE

The translation is done according to the rules in the 'character map', named in the file *charmap*. It has the following format:

1. Each line represents a translation entry, except for blank lines and comment lines, which are ignored.
2. Any amount of whitespace (space or tab) may precede the start of an entry.
3. Comment lines begin with #. Everything on the same line is ignored.
4. Each entry consists of the Unicode codepoint of the character to translate, in hexadecimal, followed *one* space or tab, followed by the translation string, up to the end of the line.
5. The translation string is taken literally, including any leading and trailing spaces (except the delimeter between the codepoint and the translation string), and all types of characters. The newline at the end is not included.

The above format is intended to be restrictive, to keep **utf8trans** simple. But if a XML-based format is desired, there is a *`xmlcharmap2utf8trans`* script that comes with the docbook2X distribution, that converts character maps in XSLT 2.0 format to the **utf8trans** format.

LIMITATIONS

- **utf8trans** does not work with binary files, because malformed UTF-8 sequences in the input are substituted with U+FFFD characters. However, null characters in the input are handled correctly. This limitation may be removed in the future.
- There is no way to include a newline or null in the substitution string.

AUTHOR

Steve Cheng <`stevecheng@users.sourceforge.net`>.