NAME

xmlfixup — Prettyprint XML documents or document fragments

SYNOPSIS

xmlfixup [**−−?**] [**−−configfile** *infile*] [**−−debug**] [**−−dumpconfig** *outfile*] [**−−help**]
[**−−indentation** *n*] [**−−level** *n*] [**−−version**] [**−−width** *n*] [**−−**] [*XML-file(s)*]

DESCRIPTION

xmlfixup formats an XML input stream from one or more input files, or from *stdin*, and writes a prettyprinted output stream on *stdout*, with tag environments indented according to user-specifiable rules.

Unlike SGML and XML parsers, **xmlfixup** can be applied to incomplete XML documents, and to document fragments, making it a convenient tool for reformatting XML markup in a text-editor session, when the editor provides for filtering text regions by an external program.

Suitable indentation usually helps to clarify document structure dramatically, and can also reveal instances of improper tag nesting. However, a strict grammatical analysis of XML documents with an SGML parser, such as **nsgmls**(1) or **sgmls**(1), or with an XML parser, such as **xmllint**(1) or **xmlwf**(1), to ensure well-formed XML is always advisable before submitting an XML stream to other XML processing tools.

To achieve the requested linewidth goal, **xmlfixup** normally breaks the input stream at whitespace or a newline when the next token would cause the requested linewidth to be exceeded. However, it will not do so at space inside an XML tag, or where there is no existing breakpoint. Thus, a long punctuation-terminated sequence like

```
<systemitem role="url">http://www.xml.org/long/path/</systemitem>.
```

will *not* be broken, and may exceed the specified linewidth.

OPTIONS

xmlfixup options can be prefixed with either one or two hyphens, and can be abbreviated to any unique prefix. Thus, **−v**, **−ver**, and **−−version** are equivalent.

−− All following command-line arguments are files, even if they look like options.

−−? Same as **−−help**.

−−configfile *infile* Supply alternate XML tag indentation rules, as described in the **CONFIGURATION FILES** section. If this option is specified more than once, **xmlfixup** uses the last one. If it is not specified at all, then **xmlfixup** supplies a default set of rules suitable for a subset of DocBook/XML markup.

−−debug Write voluminous debug output on *stderr*.

This option may be abbreviated to **−d**.

−−dumpconfig *outfile* Print the current XML tag indentation rules on *outfile*. That file can be used on a subsequent run with the **−−configfile** option.

This option may be abbreviated to **−du**.

−−help Display a brief help message on *stdout*, giving a usage description, and then terminate immediately with a success return code.

−−indentation *n* Set the number of indentation spaces for each logical tag level to *n* (default: 2).

−−level *n* Set the initial logical tag level to *n* (default: 0).

−−version Display the program version number and release date on *stdout*, and then terminate immediately with a success return code.

−−width *n* Set the output linewidth to *n* (default: 60). A negative or zero value is treated as infinite.

CONFIGURATION FILES

Because XML markup is entirely defined by the XML grammar file that is loaded at the start of any document, unlike HTML tags, XML tags do not form a small known set. Consequently, an XML prettyprinter

must be given rules that describe the desired formatting of all tags used in the input stream.

The default XML markup assumed by **xmlfixup** is a small subset of the DocBook/XML markup tags, but alternative indentation rules for other tag sets can be easily defined in a configuration file supplied as a command-line argument.

The formatting required for documents in XML markup can be specified by assigning XML tags to one of several formatting classes, illustrated by the default configuration file output with the **--dumpconfig** option:

```
# xmlfixup version 1.01 [02-Dec-2003]

Begin_Verbatim    : # clear existing list
Begin_Verbatim    : <screen>

Break_Before      : # clear existing list
Break_Before      : <colspec> <systemitem> <xref>

Empty_Line_After  : # clear existing list
Empty_Line_After  : </entry> </item> </listitem> </para> </row>
Empty_Line_After  : </screen> </tbody> </thead> </varlistentry>

Empty_Line_Before : # clear existing list
Empty_Line_Before : <entry> <item> <listitem> <para> <row>
Empty_Line_Before : <screen> <tbody> <thead> <varlistentry>
Empty_Line_Before : <xref>

End_Verbatim      : # clear existing list
End_Verbatim      : </screen>

No_Break_Before   : # clear existing list
No_Break_Before   : <footnote>

Ordinary          : # clear existing list
Ordinary          : <!> </citetitle> </command> </emphasis>
Ordinary          : </envvar> </filename> </firstterm>
Ordinary          : </indexterm> </literal> </option> </quote>
Ordinary          : </replaceable> </screen> </secondary>
Ordinary          : </subscript> </superscript> </systemitem>
Ordinary          : </term> </tertiary> </title> <?> <citetitle>
Ordinary          : <colspec> <command> <emphasis> <envvar>
Ordinary          : <filename> <firstterm> <footnoteref>
Ordinary          : <indexterm> <literal> <option> <primary>
Ordinary          : <quote> <replaceable> <screen> <secondary>
Ordinary          : <spanspec> <subscript> <superscript>
Ordinary          : <systemitem> <term> <tertiary> <title> <xref>
```

Unlike in HTML and most SGML document types, markup tags in XML are *case-sensitive*, and **xmlfixup** follows that practice. DocBook/XML tags are all lowercase, but tags for other XML document types may use mixed lettercase.

In a configuration file, comments run from sharp to end of line, and are removed before further processing. Blank or empty lines, and whitespace at start or end of line, are ignored.

The formatting class is defined by the name before the colon, which may optionally be surrounded by whitespace. The colon is followed by a whitespace-separated list of zero or more tags in that class. An empty list clears the class list, and otherwise, tags for repeated class names simply augment the list for that class.

The `Begin_Verbatim` and `End_Verbatim` classes contain tag environments that must be copied verbatim, without any change whatsoever in indentation or spacing.

The `Break_Before` class contains tags which should begin on a new line, and thus, require a line break preceding the tag.

The `No_Break_Before` class contains tags, like those for footnotes, which must be attached to preceding text, and thus, must not be preceded by a line break. However, they are otherwise treated like normal tags with indented bodies. Existing whitespace preceding the open tag will not be eliminated.

The `Empty_Line_After` and `Empty_Line_Before` classes contain tags that should appear alone on their line, with an empty line after or before them, respectively.

The `Ordinary` class contains tags that behave like ordinary text, and thus require no special formatting; in particular, they do not cause line breaks.

Two kinds of tags receive special treatment. Formatter directives of the form `<?name?>` are classed according to the reduced form `<?>`. Comments (`<!-- ...-->`) and SGML directives (`<!NAME ...>`) are classed according to the reduced form `<!>`. Consequently, only those reduced forms need be used in the classification rules.

Tags may belong to more than one class.

Tag attributes are ignored when tags are classified: thus, `<chapter id="mybook-ch-3">` is formatted just like `<chapter>` is.

Outside the seven classes, all other tags are assumed to define environments with indented bodies. This unnamed class thus serves as a catch-all for an unbounded set of unclassified tags. The *open* tag (`<name>`) appears on a line by itself at the current indentation level, the level is incremented by one for the body, and the *close* tag (`</name>`) appears on a line by itself at the same level as the open tag:

```
<tag>
  Text text text...
  Text text text...
  <tag>
    More text text text...
    <tag>
      Even more text text text...
    </tag>
  </tag>
  Text text text...
</tag>
```

xmlfixup adjusts indentation levels according to whether the tag is an open tag or a close tag: it issues a warning on *stderr*, and also inside an XML comment on *stdout*, if the tag names do not match. If this happens, it means that either the XML input is not well-formed, or else there is an inconsistency in the markup rules in a user-supplied configuration file.

SEE ALSO

gmat(1), **html-check**(1), **html-ncheck**(1), **html-pretty**(1), **html-spam**(1), **nsgmls**(1), **sgmlnorm**(1), **sgmls**(1), **xmlcatalog**(1), **xmllint**(1), **xmlwf**(1).

BUGS

The default formatting matches that found useful in a single computer-related book project. For other documents, even ones using more of the DocBook/XML markup scheme, it will almost certainly be necessary to supply a configuration file to define how you want the markup handled: use the **−−dumpconfig** option to get a starting configuration file that you can then modify as needed.

xmlfixup does not currently offer any control over the formatting of the internals of SGML comments, declarations, and directives (all of which look like `<!...>`), or of processor directives (which look like `<?...>`).

It may even prove necessary to extend **xmlfixup** with additional tag classes to handle a wider variety of

XML document types.

xmlfixup does not know anything about SGML tag minimization (use of unnamed end tags </>, and omission of end tags where their implied positions can be determined by the SGML parser from the markup grammar). A tag normalizer, such as **sgmlnorm**(1) or **spam**(1), would likely be needed to standardize SGML input before **xmlfixup** could format it sensibly.

xmlfixup does not currently offer an option to ignore lettercase in tags, making it less useful for HTML and SGML documents with inconsistent lettercase in tags. A tag normalizer can fix such problems. For HTML files, **html-pretty**(1) has extensive knowledge of HTML variants, and can do an excellent job of prettyprinting such files.

FURTHER READING

These books may be helpful in understanding DocBook/XML markup, and SGML and XML markup in general:

- Peter Flynn, *Understanding SGML and XML tools: practical programs for handling structured text*, Kluwer, 1998, ISBN 0-7923-8169-6.
- Charles F. Goldfarb and Yuri Rubinsky, *The SGML handbook*, Clarendon Press, 1990, ISBN 0-19-853737-9.
- Charles F. Goldfarb and Paul Prescod, *Charles F. Goldfarb's XML handbook*, fourth edition, Prentice-Hall PTR, 2001, ISBN 0-13-065198-2.
- Michel Goossens and Sebastian Rahtz, *The LaTeX Web companion: integrating TeX, HTML, and XML*, Addison-Wesley Longman, 1999, ISBN 0-201-43311-7.
- Norman Walsh and Leonard Muellner, *DocBook: the definitive guide*, O'Reilly & Associates, 1999, ISBN 1-56592-580-7.

References to hundreds of other books on HTML, SGML, and XML can be found in the bibliographies at:

```
http://www.math.utah.edu/pub/tex/bib/index-table.html#sgml
http://www.math.utah.edu/pub/tex/bib/index-table.html#sgml2000
```

AUTHOR

Nelson H. F. Beebe
University of Utah
Department of Mathematics, 110 LCB
155 S 1400 E RM 233
Salt Lake City, UT 84112-0090
Tel: +1 801 581 5254
FAX: +1 801 581 4148
Email: beebe@math.utah.edu, beebe@acm.org,
beebe@computer.org (Internet)
WWW URL: http://www.math.utah.edu/~beebe

AVAILABILITY

xmlfixup is freely available; its master distribution can be found at

```
ftp://ftp.math.utah.edu/pub/xmlfixup/
```

in the file *xmlfixup-x.yy.tar.gz* where *x.yy* is the current version. Other distribution formats are usually available in the same location.