Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'xmlwf.1' command

$ man xmlwf.1

XMLWF(1) [FIXME: manual] XMLWF(1)

NAME

xmlwf - Determines if an XML document is well-formed

SYNOPSIS

xmlwf [OPTIONS] [FILE ...]

xmlwf -h

xmlwf -v

DESCRIPTION

xmlwf uses the Expat library to determine if an XML document is well-formed. It is non-validating.

If you do not specify any files on the command-line, and you have a recent version of xmlwf, the input file will be read from standard input.

WELL-FORMED DOCUMENTS

A well-formed document must adhere to the following rules:

? The file begins with an XML declaration. For instance, <?xml version="1.0" standalone="yes"?>. NOTE: xmlwf does not currently check for a valid XML declaration.

? Every start tag is either empty (<tag/>) or has a corresponding end tag.

? There is exactly one root element. This element must contain all other elements in the document. Only comments, white space, and processing instructions may come after the close of the root

element.

? All elements nest properly.

? All attribute values are enclosed in quotes (either single or double).

If the document has a DTD, and it strictly complies with that DTD, then the document is also considered valid. xmlwf is a non-validating parser -- it does not check the DTD. However, it does support external entities (see the -x option).

OPTIONS

When an option includes an argument, you may specify the argument either separately ("-d output") or concatenated with the option ("-doutput"). xmlwf supports both.

-a factor

Sets the maximum tolerated amplification factor for protection against billion laughs attacks (default: 100.0). The amplification factor is calculated as ..

amplification := (direct + indirect) / direct

.. while parsing, whereas <direct> is the number of bytes read from the primary document in parsing and <indirect> is the number of bytes added by expanding entities and reading of external DTD files, combined.

NOTE: If you ever need to increase this value for non-attack payload, please file a bug report.

-b bytes

Sets the number of output bytes (including amplification) needed to activate protection against billion laughs attacks (default: 8 MiB). This can be thought of as an "activation threshold".

NOTE: If you ever need to increase this value for non-attack payload, please file a bug report.

-c

If the input file is well-formed and xmlwf doesn't encounter any errors, the input file is simply copied to the output directory unchanged. This implies no namespaces (turns off -n) and requires

-d to specify an output directory.

-d output-dir

Specifies a directory to contain transformed representations of the input files. By default, -d outputs a canonical representation (described below). You can select different output formats using -c, -m and -N.

The output filenames will be exactly the same as the input filenames or "STDIN" if the input is coming from standard input. Therefore, you must be careful that the output file does not go into the same directory as the input file. Otherwise, xmlwf will delete the input file before it generates the output file (just like running cat < file > file in most shells).

Two structurally equivalent XML documents have a byte-for-byte identical canonical XML representation. Note that ignorable white space is considered significant and is treated equivalently to data. More on canonical XML can be found at http://www.jclark.com/xml/canonxml.html .

-e encoding

Specifies the character encoding for the document, overriding any document encoding declaration. xmlwf supports four built-in encodings: US-ASCII, UTF-8, UTF-16, and ISO-8859-1. Also see the -w option.

-k

When processing multiple files, xmlwf by default halts after the the first file with an error. This tells xmlwf to report the error but to keep processing. This can be useful, for example, when testing a filter that converts many files to XML and you want to quickly find out which conversions failed.

-m

Outputs some strange sort of XML file that completely describes the input file, including character positions. Requires -d to specify an output file.

-n

Turns on namespace processing. (describe namespaces) -c disables namespaces.

-N

Adds a doctype and notation declarations to canonical XML output. This matches the example output used by the formal XML test cases. Requires -d to specify an output file.

-p

Tells xmlwf to process external DTDs and parameter entities. Normally xmlwf never parses parameter entities. -p tells it to always parse them. -p implies -x.

-r

Normally xmlwf memory-maps the XML file before parsing; this can result in faster parsing on many platforms. -r turns off memory-mapping and uses normal file IO calls instead. Of course, memory-mapping is automatically turned off when reading from standard input.
Use of memory-mapping can cause some platforms to report substantially higher memory usage for xmlwf, but this appears to be a matter of the operating system reporting memory in a strange way; there is not a leak in xmlwf.

-s

Prints an error if the document is not standalone. A document is standalone if it has no external subset and no references to parameter entities.

-t

Turns on timings. This tells Expat to parse the entire file, but not perform any processing. This gives a fairly accurate idea of the raw speed of Expat itself without client overhead. -t turns off most of the output options (-d, -m, -c, ...).

-v

Prints the version of the Expat library being used, including some information on the compile-time configuration of the library, and then exits.

-w

Enables support for Windows code pages. Normally, xmlwf will throw an error if it runs across an encoding that it is not equipped to handle itself. With -w, xmlwf will try to use a Windows code page. See also -e.

-x

Turns on parsing external entities.

Non-validating parsers are not required to resolve external entities, or even expand entities at all. Expat always expands internal entities (?), but external entity parsing must be enabled explicitly.

External entities are simply entities that obtain their data from outside the XML file currently being parsed.

This is an example of an internal entity:

```
<!ENTITY vers '1.0.2'>
```

And here are some examples of external entities:

```
<!ENTITY header SYSTEM "header-&vers;.xml">  (parsed)
<!ENTITY logo SYSTEM "logo.png" PNG>         (unparsed)
```

--

(Two hyphens.) Terminates the list of options. This is only needed if a filename starts with a hyphen. For example:

```
xmlwf -- -myfile.xml
```

will run xmlwf on the file -myfile.xml.

Older versions of xmlwf do not support reading from standard input.

OUTPUT

xmlwf outputs nothing for files which are problem-free. If any input file is not well-formed, or if the output for any input file cannot be opened, xmlwf prints a single line describing the problem to standard output.

If the -k option is not provided, xmlwf halts upon encountering a well-formedness or output-file error. If -k is provided, xmlwf continues processing the remaining input files, describing problems found with any of them.

EXIT STATUS

For option -v or -h, xmlwf always exits with status code 0. For other cases, the following exit status codes are returned:

0
: The input files are well-formed and the output (if requested) was written successfully.

1
: An internal error occurred.

2
: One or more input files were not well-formed or could not be parsed.

3
: If using the -d option, an error occurred opening an output file.

4
: There was a command-line argument error in how xmlwf was invoked.

BUGS

The errors should go to standard error, not standard output.

There should be a way to get -d to send its output to standard output rather than forcing the user to send it to a file.

I have no idea why anyone would want to use the -d, -c, and -m options. If someone could explain it to me, I'd like to add this information to this manpage.

SEE ALSO

The Expat home page: https://libexpat.github.io/

The W3 XML 1.0 specification (fourth edition): https://www.w3.org/TR/2006/REC-xml-20060816/

Billion laughs attack: https://en.wikipedia.org/wiki/Billion_laughs_attack

AUTHOR

This manual page was originally written by Scott Bronson <bronson@rinspin.com> in December 2001 for the Debian GNU/Linux(TM) system (but may be used by others). Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1.

Scott Bronson

Author.

COPYRIGHT