NAME

libxo - library for emitting text, XML, JSON, or HTML output

LIBRARY

Text, XML, JSON, and HTML Output Emission Library (libxo, -lxo)

SYNOPSIS

#include <libxo/xo.h>

DESCRIPTION

The functions defined in **libxo** are used to generate a choice of *TEXT*, *XML*, *JSON*, or *HTML* output. A common set of functions are used, with command line switches passed to the library to control the details of the output.

Most commands emit text output aimed at humans. It is designed to be parsed and understood by a user. Humans are gifted at extracting details and pattern matching. Often programmers need to extract information from this human-oriented output. Programmers use tools like grep(1), awk(1), and regular expressions to ferret out the pieces of information they need. Such solutions are fragile and require updates when output contents change or evolve, requiring testing and validation.

Modern tool developers favor encoding schemes like XML and JSON, which allow trivial parsing and extraction of data. Such formats are simple, well understood, hierarchical, easily parsed, and often integrate easier with common tools and environments.

In addition, modern reality means that more output ends up in web browsers than in terminals, making HTML output valuable.

libxo allows a single set of function calls in source code to generate traditional text output, as well as XML and JSON formatted data. HTML can also be generated; "<div>" elements surround the traditional text output, with attributes that detail how to render the data.

There are four encoding styles supported by **libxo**:

- TEXT output can be display on a terminal session, allowing compatibility with traditional command line usage.

- XML output is suitable for tools like XPath and protocols like NETCONF.

- JSON output can be used for RESTful APIs and integration with languages like Javascript and Python.

- HTML can be matched with a small CSS file to permit rendering in any HTML5 browser.

In general, XML and JSON are suitable for encoding data, while TEXT is suited for terminal output and HTML is suited for display in a web browser (see xohtml(1)).

libxo uses command line options to trigger rendering behavior. The following options are recognised:

```
--libxo <options>

--libxo=<options>

--libxo:<brief-options>
```

Options is a comma-separated list of tokens that correspond to output styles, flags, or features:

Token	**Action**
dtrt	Enable "Do The Right Thing" mode
html	Emit HTML output
indent=xx	Set the indentation level
info	Add info attributes (HTML)
json	Emit JSON output
keys	Emit the key attribute for keys (XML)
log-gettext	Log (via stderr) each gettext(3) string lookup
log-syslog	Log (via stderr) each syslog message (via xo_syslog(3))
no-humanize	Ignore the {h:} modifier (TEXT, HTML)
no-locale	Do not initialize the locale setting

no-retain	Prevent retaining formatting information
no-top	Do not emit a top set of braces (JSON)
not-first	Pretend the 1st output item was not 1st (JSON)
pretty	Emit pretty-printed output
retain	Force retaining formatting information
text	Emit TEXT output
underscores	Replace XML-friendly "-"s with JSON friendly "_"s e
units	Add the 'units' (XML) or 'data-units (HTML) attribute
warn	Emit warnings when libxo detects bad calls
warn-xml	Emit warnings in XML
xml	Emit XML output
xpath	Add XPath expressions (HTML)

The "brief-options" are single letter commands, designed for those with too little patience to use real tokens. No comma separator is used.

Token	**Action**
H	Enable HTML output (XO_STYLE_HTML)
I	Enable info output (XOF_INFO)
i<num>	Indent by <number>
J	Enable JSON output (XO_STYLE_JSON)
P	Enable pretty-printed output (XOF_PRETTY)
T	Enable text output (XO_STYLE_TEXT)
W	Enable warnings (XOF_WARN)
X	Enable XML output (XO_STYLE_XML)
x	Enable XPath data (XOF_XPATH)

Refer to xo_options(7) for additional option information.

The **libxo** library allows an application to generate text, XML, JSON, and HTML output using a common set of function calls. The application decides at run time which output style should be produced. The application calls a function xo_emit(3) to product output that is described in a format string. A "field descriptor" tells **libxo** what the field is and what it means. Each field descriptor is placed in braces with a printf-like format string:

```
xo_emit(" {:lines/%7ju} {:words/%7ju} "
      "{:characters/%7ju}{d:filename/%s}\n",
      linect, wordct, charct, file);
```

Each field can have a role, with the 'value' role being the default, and the role tells **libxo** how and when to render that field, as well as a printf(3)-like format string.

Output can then be generated in various style, using the "--libxo" option.

DEFAULT HANDLE

Handles give an abstraction for **libxo** that encapsulates the state of a stream of output. Handles have the data type "xo_handle_t" and are opaque to the caller.

The library has a default handle that is automatically initialized. By default, this handle will send text style output to standard output. The xo_set_style(3) and xo_set_flags(3) functions can be used to change this behavior.

Many **libxo** functions take a handle as their first parameter; most that do not use the default handle. Any function taking a handle can be passed NULL to access the default handle.

For the typical command that is generating output on standard output, there is no need to create an explicit handle, but they are available when needed, e.g., for daemons that generate multiple streams of output.

FUNCTION OVERVIEW

The **libxo** library includes the following functions:

Function	**Description**
xo_attr()	
xo_attr_h()	
xo_attr_hv()	Allows the caller to emit XML attributes with the next open element.

xo_create()

xo_create_to_file() Allow the caller to create a new handle. Note that **libxo** has a default handle that allows the caller to avoid use of an explicitly created handle. Only callers writing to files other than stdout would need to call **xo_create**().

xo_destroy() Frees any resources associated with the handle, including the handle itself.

xo_emit()

xo_emit_h()

xo_emit_hv() Emit formatted output. The *fmt* string controls the conversion of the remaining arguments into formatted output. See xo_format(5) for details.

xo_emit_warn()

xo_emit_warnx()

xo_emit_warn_c()

xo_emit_warn_hc()

xo_emit_err()

xo_emit_errc()

xo_emit_errx() These functions are mildly compatible with their standard libc namesakes, but use the format string defined in xo_format(5). While there is an increased cost for converting the strings, the output provided can be richer and more useful. See also xo_err(3)

xo_warn()

xo_warnx()

xo_warn_c()

xo_warn_hc()

xo_err()

xo_errc()

xo_errx()

xo_message()

xo_message_c()

xo_message_hc()

xo_message_hcv() These functions are meant to be compatible with their standard libc namesakes.

xo_finish()

xo_finish_h() Flush output, close open construct, and complete any pending operations.

xo_flush()

xo_flush_h() Allow the caller to flush any pending output for a handle.

xo_no_setlocale() Direct **libxo** to avoid initializing the locale. This function should be called before any other **libxo** function is called.

xo_open_container()

xo_open_container_h()

xo_open_container_hd()

xo_open_container_d()

xo_close_container()

xo_close_container_h()

xo_close_container_hd()

xo_close_container_d()

Containers a singleton levels of hierarchy, typically used to organize related content.

xo_open_list_h()

xo_open_list()

xo_open_list_hd()

xo_open_list_d()

xo_open_instance_h()

xo_open_instance()

xo_open_instance_hd()

xo_open_instance_d()

xo_close_instance_h()

xo_close_instance()

xo_close_instance_hd()

xo_close_instance_d()

xo_close_list_h()

xo_close_list()

xo_close_list_hd()

xo_close_list_d() Lists are levels of hierarchy that can appear multiple times within the same parent. Two calls are needed to encapsulate them, one for the list and one for each instance of that list. Typically **xo_open_list**() and **xo_close_list**() are called outside a for-loop, where **xo_open_instance**() it called at the top of the loop, and **xo_close_instance**() is called at the bottom of the loop.

xo_parse_args() Inspects command line arguments for directions to **libxo**. This function should be called before *argv* is inspected by the application.

xo_set_allocator() Instructs **libxo** to use an alternative memory allocator and deallocator.

xo_set_flags()

xo_clear_flags() Change the flags set for a handle.

xo_set_info() Provides additional information about elements for use with HTML rendering.

xo_set_options() Changes formatting options used by handle.

xo_set_style()

xo_set_style_name() Changes the output style used by a handle.

xo_set_writer() Instructs **libxo** to use an alternative set of low-level output functions.

SEE ALSO

libxo-csv(7,) xo(1), xolint(1), xo_attr(3), xo_create(3), xo_emit(3), xo_emit_err(3), xo_err(3), xo_finish(3), xo_flush(3), xo_no_setlocale(3), xo_open_container(3), xo_open_list(3), xo_options(7,) xo_parse_args(3), xo_set_allocator(3), xo_set_flags(3), xo_set_info(3), xo_set_options(3), xo_set_style(3), xo_set_writer(3), xo_format(5)

HISTORY

The **libxo** library first appeared in FreeBSD 11.0.

AUTHORS

libxo was written by Phil Shafer <*phil@freebsd.org*>.