

NAME

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** - open (and close) container constructs

LIBRARY

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

SYNOPSIS

#include <libxo/xo.h>

xo_ssize_t

xo_open_container(*const char *name*);

xo_ssize_t

xo_open_container_h(*xo_handle_t *handle, const char *name*);

xo_ssize_t

xo_open_container_hd(*xo_handle_t *handle, const char *name*);

xo_ssize_t

xo_open_container_d(*const char *name*);

xo_ssize_t

xo_close_container(*const char *name*);

xo_ssize_t

xo_close_container_h(*xo_handle_t *handle, const char *name*);

xo_ssize_t

xo_close_container_hd(*xo_handle_t *handle*);

xo_ssize_t

xo_close_container_d(*void*);

DESCRIPTION

libxo represents two types of hierarchy: "containers" and "lists". A container appears once under a given parent where a list contains instances that can appear multiple times. A container is used to hold related fields and to give the data organization and scope. The container has no value, but serves to contain other nodes.

To open a container, call `xo_open_container()` or `xo_open_container_h()`. The former uses the default handle and the latter accepts a specific handle.

To close a level, use the `xo_close_container()` or `xo_close_container_h()` functions.

Each open call should have a matching close call. If the `XOF_WARN` flag is set and the name given does not match the name of the currently open container, a warning will be generated.

Example:

```
xo_open_container("top");
xo_open_container("system");
xo_emit("{:host-name/%s%s%s", hostname,
        domainname ? "." : "", domainname ? : "");
xo_close_container("system");
xo_close_container("top");
```

Sample Output:

Text:

```
my-host.example.org
```

XML:

```
<top>
  <system>
    <host-name>my-host.example.org</host-name>
  </system>
</top>
```

JSON:

```
"top" : {
  "system" : {
    "host-name": "my-host.example.org"
  }
}
```

HTML:

```
<div class="data"
  data-tag="host-name">my-host.example.org</div>
```

EMITTING HIERARCHY

To create a container, use the `xo_open_container()` and `xo_close_container()` set of functions. The *handle* parameter contains a handle such as returned by `xo_create(3)` or `NULL` to use the default handle. The *name* parameter gives the name of the container, encoded in *UTF-8*. Since *ASCII* is a proper subset of *UTF-8*, traditional C strings can be used directly.

The close functions with the "_d" suffix are used in "Do The Right Thing" mode, where the name of the open containers, lists, and instances are maintained internally by **libxo** to allow the caller to avoid keeping track of the open container name.

Use the XOF_WARN flag to generate a warning if the name given on the close does not match the current open container.

For TEXT and HTML output, containers are not rendered into output text, though for HTML they are used when the XOF_XPATH flag is set.

EXAMPLE:

```
xo_open_container("system");
xo_emit("The host name is {:host-name}\n", hn);
xo_close_container("system");
```

XML:

```
<system><host-name>foo</host-name></system>
```

DTRT MODE

Some users may find tracking the names of open containers, lists, and instances inconvenient. **libxo** offers a "Do The Right Thing" mode, where **libxo** will track the names of open containers, lists, and instances so the close function can be called without a name. To enable *DTRT* mode, turn on the XOF_DTRT flag prior to making any other **libxo** output.

```
xo_set_flags(NULL, XOF_DTRT);
```

Each open and close function has a version with the suffix "_d", which will close the open container, list, or instance:

```
xo_open_container("top");
...
xo_close_container_d();
```

Note that the XOF_WARN flag will also cause **libxo** to track open containers, lists, and instances. A warning is generated when the name given to the close function and the name recorded do not match.

SEE ALSO

xo_emit(3), libxo(3)

HISTORY

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

AUTHORS

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