NAME

xo_open_marker, xo_open_marker_h, xo_close_marker, xo_close_marker_h - prevent and allow closing
of open constructs

LIBRARY

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

SYNOPSIS

#include <libxo/xo.h>

xo_ssize_t
xo_open_marker(*const char *name*);

xo_ssize_t
xo_open_marker_h(*xo_handle_t *handle*, *const char *name*);

xo_ssize_t
xo_close_marker(*const char *name*);

xo_ssize_t
xo_close_marker_h(*xo_handle_t *handle*, *const char *name*);

DESCRIPTION

libxo represents hierarchy using two constructs: "containers" and "lists". A marker can be used to affect
how open constructs are closed, either by preventing their (implicit or explicit) closure or by forcing
their closure. While a marker is open, no other open constructs can be closed. When a marker is closed,
all constructs open since the marker was opened will be closed. A marker is used to "freeze" any open
constructs. Calls to **xo_close_***() functions that would normally close them will be ignored, effectively
blocking their closure. However when **xo_close_marker**() is called, any containers, lists, or leaf-lists
open since the matching **xo_open_marker**() call will be close and the marker discarded. Markers use
names which are not user-visible, allowing the caller to choose appropriate internal names. The marker
has no value and is not emitted in any form.

To open a marker, call **xo_open_marker**() or **xo_open_marker_h**(). The former uses the default handle
and the latter accepts a specific handle.

To close a marker, use the **xo_close_marker**() or **xo_close_marker_h**() functions.

Each open call must have a matching close call.

LIBXO(3)

In this example, the **xo_close_container**() call on line [1] will be ignored, since the open marker "outer" will prevent close of any open constructs that precede it. The **xo_close_marker**() call on line [2] will close the "system" container, since it was opened after the "outer" marker.

Example:

```
xo_open_container("top");
    xo_open_marker("outer");
xo_open_container("system");
xo_emit("{:host-name/%s%s%s", hostname,
    domainname ? "." : "", domainname ?: "");
xo_close_container("top");   /* [1] */
    xo_close_marker("outer");   /* [2] */
xo_close_container("top");
```

In this example, the code whiffles through a list of fish, calling a function to emit details about each fish. The marker "fish-guts" is used to ensure that any constructs opened by the function are closed properly.

```
for (i = 0; fish[i]; i++) {
    xo_open_instance("fish");
    xo_open_marker("fish-guts");
    dump_fish_details(i);
    xo_close_marker("fish-guts");
}
```

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>*.