NAME

fn_ctx_destroy_subcontext – destroy the named context and remove its binding from the parent context

SYNOPSIS

cc
[*flag* ...] *file* ...
–lxfn
[*library* ...]

#include <xfn/xfn.h>

int fn_ctx_destroy_subcontext(FN_ctx_t **ctx*, **const FN_composite_name_t** **name*,
  FN_status_t **status***);**

MT-LEVEL

Safe.

DESCRIPTION

This operation destroys the subcontext named by *name* relative to *ctx*, and unbinds the name.

As with **fn_ctx_unbind( ),** this operation succeeds even if the terminal atomic name is not bound in the target context — the context named by all but the terminal atomic name in *name*.

RETURN VALUE

fn_ctx_destroy_subcontext( ) returns **1** on success and **0** on failure.

ERRORS

fn_ctx_destroy_subcontext( ) sets *status* as described in **FN_status_t**(3N) and **xfn_status_codes**(3N). Of special relevance for **fn_ctx_destroy_subcontext( )** are the following status codes:

FN_E_CTX_NOT_A_CONTEXT
  name does not name a context.

FN_E_CTX_NOT_EMPTY
  The naming system being asked to do the destroy does not support removal of a context that still contains bindings.

APPLICATION USAGE

Some aspects of this operation are not specified by XFN, but are determined by the target context and its naming system. For example, XFN does not specify what happens if the named subcontext is non-empty when the operation is invoked.

In naming systems that support attributes, and store the attributes along with names or contexts, this operation removes the name, the context, and its associated attributes.

Normal resolution always follows links. In a **fn_ctx_destroy_subcontext( )** operation, resolution of *name* continues to the target context; the terminal atomic name is not resolved. If the terminal atomic name is bound to a link, the link is not followed and the operation fails with **FN_E_CTX_NOT_A_CONTEXT** because the name is not bound to a context.

SEE ALSO

FN_ctx_t(3N), **FN_composite_name_t**(3N), **FN_status_t**(3N), **fn_ctx_create_subcontext**(3N), **fn_ctx_unbind**(3N), **xfn_status_codes**(3N), **xfn**(3N)