NAME

FN_ref_addr_t, fn_ref_addr_create, fn_ref_addr_destroy, fn_ref_addr_copy, fn_ref_addr_assign, fn_ref_addr_type, fn_ref_addr_length, fn_ref_addr_data, fn_ref_addr_description – an address in an XFN reference

SYNOPSIS

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

#include <xfn/xfn.h>

FN_ref_addr_t ∗**fn_ref_addr_create(const FN_identifier_t** ∗*type*, **size_t** *length*,
    const void ∗*data***);**

void fn_ref_addr_destroy(FN_ref_addr_t ∗*addr***);**

FN_ref_addr_t ∗**fn_ref_addr_copy(const FN_ref_addr_t** ∗*addr***);**

FN_ref_addr_t ∗**fn_ref_addr_assign(FN_ref_addr_t** ∗*dst*, **const FN_ref_addr_t** ∗*src***);**

const FN_identifier_t ∗**fn_ref_addr_type(const FN_ref_addr_t** ∗*addr***);**

size_t fn_ref_addr_length(const FN_ref_addr_t ∗*addr***);**

const void∗ **fn_ref_addr_data(const FN_ref_addr_t** ∗*addr***);**

FN_string_t ∗**fn_ref_addr_description(const FN_ref_addr_t** ∗*addr*, **unsigned int** *detail*,
    unsigned int ∗*more_detail***);**

MT-LEVEL

Safe.

DESCRIPTION

An XFN reference is represented by the type **FN_ref_t**. An object of this type contains a reference type and a list of addresses. Each address in the list is represented by an object of type **FN_ref_addr_t**. An address consists of an opaque data buffer and a type field, of type **FN_identifier_t**.

fn_ref_addr_create( ) creates and returns an address with the given type and data. *length* indicates the size of the data. **fn_ref_addr_destroy( )** releases the storage associated with the given address. **fn_ref_addr_copy( )** returns a copy of the given address object. **fn_ref_addr_assign( )** makes a copy of the address pointed to by *src* and assigns it to *dst*, releasing any old contents of *dst*. A pointer to the same object as *dst* is returned.

fn_ref_addr_type( ) returns the type of the given address. **fn_ref_addr_length( )** returns the size of the address in bytes. **fn_ref_addr_data( )** returns the contents of the address.

fn_ref_addr_description( ) returns the implementation-defined textual description of the address. It takes as arguments a number, *detail*, and a pointer to a number *more_detail*. *detail* specifies the level of detail for which the description should be generated; the higher the number, the more detail is to be provided. If *more_detail* is zero, it is ignored. If *more_detail* is non-zero, it is set by the description operation to indicate the next level of detail available, beyond that specified by *detail*. If no higher level of detail is available, *more_detail* is set to *detail*.

APPLICATION USAGE

The address type of an **FN_ref_addr_t** object is intended to identify the mechanism that should be used to reach the object using that address. The client must interpret the contents of the opaque data buffer of the address based on the type of the address, and on the type of the reference that the address is in. However, this interpretation is intended to occur below the application layer. Most applications developers should not have to manipulate the contents of either address or reference objects themselves. These interfaces would generally be used within service libraries.

Multiple addresses in a single reference are intended to identify multiple communication endpoints for the same conceptual object. Multiple addresses may arise for various reasons, such as the object offering interfaces over more than one communication mechanism.

Manipulation of addresses using the operations described in this manual page does not affect their representation in the underlying naming system. Changes to addresses in the underlying naming system can only be effected through the use of the interfaces described in **FN_ctx_t**(3N).

SEE ALSO **FN_ctx_t**(3N), **FN_identifier_t**(3N), **FN_ref_t**(3N), **FN_string_t**(3N), **xfn**(3N)