NAME

FN_attrset_t, fn_attrset_create, fn_attrset_destroy, fn_attrset_copy, fn_attrset_assign, fn_attrset_get, fn_attrset_count, fn_attrset_first, fn_attrset_next, fn_attrset_add, fn_attrset_remove – a set of XFN attributes

SYNOPSIS

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

#include <xfn/xfn.h>

FN_attrset_t *fn_attrset_create(void);

void fn_attrset_destroy(FN_attrset_t **aset***);**

FN_attrset_t *fn_attrset_copy(const FN_attrset_t **aset***);**

FN_attrset_t *fn_attrset_assign(FN_attrset_t **dst***, const FN_attrset_t** **src***);**

const FN_attribute_t *fn_attrset_get(const FN_attrset_t **aset***,**
const FN_identifier_t **attr_id***);**

unsigned int fn_attrset_count(const FN_attrset_t **aset***);**

const FN_attribute_t *fn_attrset_first(const FN_attrset_t **aset*,. BI "void " **iter_pos);

const FN_attribute_t *fn_attrset_next(const FN_attrset_t **aset***, void** ***iter_pos***);**

int fn_attrset_add(FN_attrset_t **aset***, const FN_attribute_t** **attr***,**
unsigned int *exclusive***);**

int fn_attrset_remove(FN_attrset_t **aset***, const FN_identifier_t** **attr_id***);**

MT-LEVEL

Safe.

DESCRIPTION

An attribute set is a set of attribute objects with distinct identifiers. The **fn_attr_multi_get( )** operation takes an attribute set as parameter and returns an attribute set. The **fn_attr_get_ids( )** operation returns an attribute set containing the identifiers of the attributes.

Attribute sets are represented by the type **FN_attrset_t**. The following operations are defined for manipulating attribute sets.

fn_attrset_create( ) creates an empty attribute set. **fn_attrset_destroy( )** releases the storage associated with the attribute set *aset*. **fn_attrset_copy( )** returns a copy of the attribute set *aset*. **fn_attrset_assign( )** makes a copy of the attribute set *src* and assigns it to *dst*, releasing any old contents of *dst*. A pointer to the same object as *dst* is returned.

fn_attrset_get( ) returns the attribute with the given identifier *attr_id* from *aset*. **fn_attrset_count( )** returns the number attributes found in the attribute set *aset*.

fn_attrset_first( ) and **fn_attrset_next( )** are functions that can be used to return an enumeration of all the attributes in an attribute set. The attributes are not ordered in any way. There is no guaranteed relation between the order in which items are added to an attribute set and the order of the enumeration. The specification does guarantee that any two enumeration will return the members in the same order, provided that no **fn_attrset_add( )** or **fn_attrset_remove( )** operation was performed on the object in between or during the two enumerations. **fn_attrset_first( )** returns the first attribute from the set and sets *iter_pos* after the first attribute. **fn_attrset_next( )** returns the attribute following *iter_pos* and advances *iter_pos*.

fn_attrset_add( ) adds the attribute *attr* to the attribute set *aset*, replacing the attribute's values if the identifier of *attr* is not distinct in *aset* and *exclusive* is zero. If *exclusive* is nonzero and the identifier of *attr* is not distinct in *aset*, the operation fails. **fn_attrset_remove( )** removes the attribute with the identifier *attr_id* from *aset*. The operation succeeds even if no such attribute occurs in *aset*.

RETURN VALUE

fn_attrset_first( ) returns **0** if the attribute set is empty. **fn_attrset_next( )** returns **0** if there are no more attributes in the set.

fn_attrset_add( ) and **fn_attrset_remove( )** return **1** if the operation succeeds, and **0** if the operation fails.

APPLICATION USAGE

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

SEE ALSO

FN_attribute_t(3N), **FN_attrvalue_t**(3N), **FN_identifier_t**(3N), **fn_attr_multi_get**(3N), **fn_attr_get_ids**(3N), **xfn_attributes**(3N), **xfn**(3N)