NAME

fn_attr_get_values, FN_valuelist_t, fn_valuelist_next, fn_valuelist_destroy – return values of an attribute

SYNOPSIS

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

#include <xfn/xfn.h>

FN_valuelist_t ∗**fn_attr_get_values(FN_ctx_t** ∗*ctx*, **const FN_composite_name_t** ∗*name*, **const FN_identifier_t** ∗*attribute_id*, **FN_status_t** ∗*status*);

FN_attrvalue_t ∗**fn_valuelist_next(FN_valuelist_t** ∗*vl*, **FN_identifier_t** ∗∗*attr_syntax*, **FN_status_t** ∗*status*);

void fn_valuelist_destroy(FN_valuelist_t∗*vl*, **FN_status_t** ∗*status*);

MT-LEVEL

Safe.

DESCRIPTION

This set of operations is used to obtain the values of a single attribute, identified by *attribute_id*, associated with the object named *name*, resolved in the context *ctx*. If *name* is empty, the attribute values associated with *ctx* are obtained.

The operation **fn_attr_get_values( )** initiates the enumeration process. It returns a handle to an **FN_valuelist_t** object that can be used to enumerate the values of the specified attribute.

The operation **fn_valuelist_next( )** returns a new **FN_attrvalue_t** object containing the next value in the attribute and may be called multiple times until all values are retrieved. The syntax of the attribute is returned in *attr_syntax.*

The operation **fn_valuelist_destroy( )** is used to release the resources used during the enumeration. This may be invoked before the enumeration has completed to terminate the enumeration.

These operations work in a similar fashion as the **fn_ctx_list_names( )** operations.

RETURN VALUE

fn_attr_get_values( ) returns a pointer to an **FN_valuelist_t** object if the enumeration process is successfully initiated; it returns a **NULL** pointer if the process failed.

fn_valuelist_next ( ) returns a **NULL** pointer if no more attribute value can be returned.

In the case of a failure, these operations set *status* to indicate the nature of the failure.

ERRORS

Each successful call to **fn_valuelist_next( )** returns an attribute value. *status* is set to **FN_SUCCESS.**

When **fn_valuelist_next( )** returns a **NULL** pointer, it indicates that no more values can be returned. *status* is set in the following way:

FN_SUCCESS
The enumeration has completed successfully.

FN_E_INVALID_ENUM_HANDLE
The given enumeration handle is not valid. Possible reasons could be that the handle was from another enumeration, or the context being enumerated no

longer accepts the handle (due to such events as handle expiration or updates to the context).

FN_E_PARTIAL_RESULT
The enumeration is not yet complete but cannot be continued.

In addition to these status codes, other status codes are also possible in calls to these operations. In such cases, *status* is set as described in **FN_status_t**(3N) and **xfn_status_codes**(3N).

APPLICATION USAGE

This interface should be used instead of **fn_attr_get( )** if the combined size of all the values is expected to be too large to be returned by **fn_attr_get( ).**

There may be a relationship between the *ctx* argument supplied to **fn_attr_get_values( )** and the **FN_valuelist_t** object it returns. For example, some implementations may store the context handle *ctx* within the **FN_valuelist_t** object for subsequent **fn_valuelist_next( )** calls. In general, a **fn_ctx_handle_destroy( )** should not be invoked on *ctx* until the enumeration has terminated.

SEE ALSO

FN_attribute_t(3N), **FN_attrvalue_t**(3N), **FN_composite_name_t**(3N), **FN_ctx_t**(3N), **FN_identifier_t**(3N), **FN_status_t**(3N), **fn_attr_get**(3N), **fn_ctx_list_names**(3N), **xfn_attributes**(3N), **xfn_status_codes**(3N), **xfn**(3N)