NAME

fido_dev_info_manifest, **fido_dev_info_new**, **fido_dev_info_free**, **fido_dev_info_ptr**, **fido_dev_info_path**, **fido_dev_info_product**, **fido_dev_info_vendor**, **fido_dev_info_manufacturer_string**, **fido_dev_info_product_string**, **fido_dev_info_set** - FIDO2 device discovery functions

SYNOPSIS

#include <fido.h>

int
fido_dev_info_manifest(*fido_dev_info_t *devlist, size_t ilen, size_t *olen*);

*fido_dev_info_t **
fido_dev_info_new(*size_t n*);

void
fido_dev_info_free(*fido_dev_info_t **devlist_p, size_t n*);

*const fido_dev_info_t **
fido_dev_info_ptr(*const fido_dev_info_t *devlist, size_t i*);

*const char **
fido_dev_info_path(*const fido_dev_info_t *di*);

int16_t
fido_dev_info_product(*const fido_dev_info_t *di*);

int16_t
fido_dev_info_vendor(*const fido_dev_info_t *di*);

*const char **
fido_dev_info_manufacturer_string(*const fido_dev_info_t *di*);

*const char **
fido_dev_info_product_string(*const fido_dev_info_t *di*);

int
fido_dev_info_set(*fido_dev_info_t *devlist, size_t i, const char *path, const char *manufacturer, const char *product, const fido_dev_io_t *io, const fido_dev_transport_t *transport*);

DESCRIPTION

The **fido_dev_info_manifest**() function fills *devlist* with up to *ilen* FIDO2 devices found by the underlying operating system. Currently only USB HID devices are supported. The number of discovered devices is returned in *olen*, where *olen* is an addressable pointer.

The **fido_dev_info_new**() function returns a pointer to a newly allocated, empty device list with *n* available slots. If memory is not available, NULL is returned.

The **fido_dev_info_free**() function releases the memory backing **devlist_p*, where **devlist_p* must have been previously allocated by **fido_dev_info_new**(). The number *n* of allocated slots must also be provided. On return, **devlist_p* is set to NULL. Either *devlist_p* or **devlist_p* may be NULL, in which case **fido_dev_info_free**() is a NOP.

The **fido_dev_info_ptr**() function returns a pointer to slot number *i* of *devlist*. It is the caller's responsibility to ensure that *i* is bounded. Please note that the first slot has index 0.

The **fido_dev_info_path**() function returns the filesystem path or subsystem-specific identification string of *di*.

The **fido_dev_info_product**() function returns the product ID of *di*.

The **fido_dev_info_vendor**() function returns the vendor ID of *di*.

The **fido_dev_info_manufacturer_string**() function returns the manufacturer string of *di*. If *di* does not have an associated manufacturer string, **fido_dev_info_manufacturer_string**() returns an empty string.

The **fido_dev_info_product_string**() function returns the product string of *di*. If *di* does not have an associated product string, **fido_dev_info_product_string**() returns an empty string.

An example of how to use the functions described in this document can be found in the *examples/manifest.c* file shipped with *libfido2*.

The **fido_dev_info_set**() function initializes an entry in a device list allocated by **fido_dev_info_new**() with the specified path, manufacturer, and product strings, and with the specified I/O handlers and, optionally, transport functions, as described in fido_dev_set_io_functions(3). The *io* argument must be specified; the *transport* argument may be NULL. The path, I/O handlers, and transport functions will be used automatically by fido_dev_new_with_info(3) and fido_dev_open_with_info(3). An application can use this, for example, to substitute mock FIDO2 devices in testing for the real ones that **fido_dev_info_manifest**() would discover.

RETURN VALUES

The **fido_dev_info_manifest**() function always returns FIDO_OK. If a discovery error occurs, the *olen* pointer is set to 0.

On success, the **fido_dev_info_set**() function returns FIDO_OK. On error, a different error code defined in <*fido/err.h*> is returned.

The pointers returned by **fido_dev_info_ptr**(), **fido_dev_info_path**(), **fido_dev_info_manufacturer_string**(), and **fido_dev_info_product_string**() are guaranteed to exist until **fido_dev_info_free**() is called on the corresponding device list.