

**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.