

**Yubico Universal 2nd Factor C Library**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i>	
	Yubico Universal 2nd Factor C Library	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		March 24, 2022
		<i>SIGNATURE</i>

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Yubico Universal 2nd Factor C Library</b>	<b>1</b>
1.1	u2f-host . . . . .	1
1.2	u2f-host-types . . . . .	1
1.3	u2f-host-version . . . . .	3
<b>2</b>	<b>Index</b>	<b>5</b>

## Chapter 1

# Yubico Universal 2nd Factor C Library

This is a C library that implements the host-side of the U2F protocol. More precisely, it provides an API for applications that wishes to talk to a U2F device and perform the U2F Register and U2F Authenticate operations.

### 1.1 u2f-host

u2f-host —

#### Description

#### Functions

### 1.2 u2f-host-types

u2f-host-types —

#### Types and Values

enum	<a href="#">u2fh_rc</a>
enum	<a href="#">u2fh_initflags</a>
enum	<a href="#">u2fh_cmdflags</a>
typedef	<a href="#">u2fh_devs</a>

#### Description

#### Functions

#### Types and Values

enum `u2fh_rc`

Error codes.

---

**Members**

U2FH_OK	Success.
U2FH_MEMORY_ERROR	Memory error.
U2FH_TRANSPORT_ERROR	Transport (e.g., USB) error.
U2FH_JSON_ERROR	Json error.
U2FH_BASE64_ERROR	Base64 error.
U2FH_NO_U2F_DEVICE	Missing U2F device.
U2FH_AUTHENTICATOR_ERROR	Authenticator error.
U2FH_TIMEOUT_ERROR	Timeout error.
U2FH_SIZE_ERROR	

**enum u2fh\_initflags**

Flags passed to `u2fh_global_init()`.

**Members**

U2FH_DEBUG	Print debug messages.
------------	-----------------------

**enum u2fh\_cmdflags**

Flags passed to `u2fh_register()` and `u2fh_authenticate()`.

**Members**

U2FH_REQUEST_USER_PRESENCE	Request user presence.
----------------------------	------------------------

**u2fh\_devs**

```
typedef struct u2fh_devs u2fh_devs;
```

**1.3 u2f-host-version**

u2f-host-version —

**Functions**

const char\* | [u2fh\\_check\\_version\(\)](#)

**Types and Values**

#define	<a href="#">U2FH_VERSION_STRING</a>
#define	<a href="#">U2FH_VERSION_NUMBER</a>
#define	<a href="#">U2FH_VERSION_MAJOR</a>
#define	<a href="#">U2FH_VERSION_MINOR</a>
#define	<a href="#">U2FH_VERSION_PATCH</a>

**Description****Functions****u2fh\_check\_version()**

```
const char~*
u2fh_check_version (const char *req_version);
```

Check that the version of the library is at minimum the requested one and return the version string; return NULL if the condition is not satisfied. If a NULL is passed to this function, no check is done, but the version string is simply returned.

See [U2FH\\_VERSION\\_STRING](#) for a suitable *req\_version* string.

**Parameters**

req\_version | Required version number, or NULL. |

**Returns**

Version string of run-time library, or NULL if the run-time library does not meet the required version number.

**Types and Values****U2FH\_VERSION\_STRING**

```
#define U2FH_VERSION_STRING "1.1.10"
```

Pre-processor symbol with a string that describe the header file version number. Used together with [u2fh\\_check\\_version\(\)](#) to verify header file and run-time library consistency.

**U2FH\_VERSION\_NUMBER**

```
#define U2FH_VERSION_NUMBER 0x01010a
```

Pre-processor symbol with a hexadecimal value describing the header file version number. For example, when the header version is 1.2.3 this symbol will have the value 0x01020300. The last two digits are only used between public releases, and will otherwise be 00.

**U2FH\_VERSION\_MAJOR**

```
#define U2FH_VERSION_MAJOR 1
```

Pre-processor symbol with a decimal value that describe the major level of the header file version number. For example, when the header version is 1.2.3 this symbol will be 1.

**U2FH\_VERSION\_MINOR**

```
#define U2FH_VERSION_MINOR 1
```

Pre-processor symbol with a decimal value that describe the minor level of the header file version number. For example, when the header version is 1.2.3 this symbol will be 2.

**U2FH\_VERSION\_PATCH**

```
#define U2FH_VERSION_PATCH 10
```

Pre-processor symbol with a decimal value that describe the patch level of the header file version number. For example, when the header version is 1.2.3 this symbol will be 3.

---

## Chapter 2

# Index

### U

u2fh\_check\_version, 3

u2fh\_cmdflags, 2

u2fh\_devs, 3

u2fh\_initflags, 2

u2fh\_rc, 1

U2FH\_VERSION\_MAJOR, 4

U2FH\_VERSION\_MINOR, 4

U2FH\_VERSION\_NUMBER, 4

U2FH\_VERSION\_PATCH, 4

U2FH\_VERSION\_STRING, 3