

# **Yubico YubiKey NEO CCID Manager C Library**

---



**COLLABORATORS**

	<i>TITLE :</i> Yubico YubiKey NEO CCID Manager C Library		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		April 9, 2015	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME



# Contents

<b>1</b>	<b>Yubico YubiKey NEO CCID Manager C Library</b>	<b>1</b>
1.1	ykneomgr . . . . .	1
1.2	ykneomgr-types . . . . .	10
1.3	ykneomgr-version . . . . .	11
<b>2</b>	<b>Index</b>	<b>14</b>



## Chapter 1

# Yubico YubiKey NEO CCID Manager C Library

This is a C library to interact with the CCID-part of the YubiKey NEO. There is a command line tool "ykneomgr" for interactive use. It supports querying the YubiKey NEO for firmware version, operation mode (OTP/CCID) and serial number. You may also mode switch the device and manage applets (list, delete and install).

For more information about Yubico and the YubiKey, see: <https://www.yubico.com/>

### 1.1 ykneomgr

ykneomgr —

#### Functions

ykneomgr_rc	ykneomgr_global_init ()
void	ykneomgr_global_done ()
const char *	ykneomgr_strerror ()
const char *	ykneomgr_strerror_name ()
ykneomgr_rc	ykneomgr_init ()
void	ykneomgr_done ()
ykneomgr_rc	ykneomgr_list_devices ()
ykneomgr_rc	ykneomgr_connect ()
ykneomgr_rc	ykneomgr_discover ()
ykneomgr_rc	ykneomgr_discover_match ()
uint8_t	ykneomgr_get_version_major ()
uint8_t	ykneomgr_get_version_minor ()
uint8_t	ykneomgr_get_version_build ()
uint8_t	ykneomgr_get_mode ()
uint32_t	ykneomgr_get_serialno ()
ykneomgr_rc	ykneomgr_modeswitch ()
ykneomgr_rc	ykneomgr_authenticate ()
ykneomgr_rc	ykneomgr_applet_list ()
ykneomgr_rc	ykneomgr_applet_delete ()
ykneomgr_rc	ykneomgr_applet_install ()
ykneomgr_rc	ykneomgr_send_apdu ()

#### Object Hierarchy



## Description

## Functions

### ykneomgr\_global\_init ()

```
ykneomgr_rc  
ykneomgr_global_init (ykneomgr_initflags flags);
```

Initialize the library. This function is not guaranteed to be thread safe and must be invoked on application startup.

#### Parameters

flags	initialization flags, ORed ykneomgr_initflags.	
-------	---	--

#### Returns

On success **YKNEOMGR\_OK** (integer 0) is returned, and on errors an **ykneomgr\_rc** error code.

### ykneomgr\_global\_done ()

```
void  
ykneomgr_global_done (void);
```

Release all resources from the library. Call this function when no further use of the library is needed.

### ykneomgr\_strerror ()

```
const char~*  
ykneomgr_strerror (int err);
```

Convert return code to human readable string explanation of the reason for the particular error code.

This string can be used to output a diagnostic message to the user.

This function is one of few in the library that can be used without a successful call to **ykneomgr\_global\_init()**.

#### Parameters

err	error code	
-----	------------	--

#### Returns

Returns a pointer to a statically allocated string containing an explanation of the error code *err* .

### ykneomgr\_strerror\_name ()

```
const char~*  
ykneomgr_strerror_name (int err);
```

Convert return code to human readable string representing the error code symbol itself. For example, **ykneomgr\_strerror\_name(YKNEOMGR\_OK)** returns the string "YKNEOMGR\_OK".

---



This string can be used to output a diagnostic message to the user.

This function is one of few in the library that can be used without a successful call to `ykneomgr_global_init()`.

### Parameters

<code>err</code>	error code	
------------------	------------	--

### Returns

Returns a pointer to a statically allocated string containing a string version of the error code `err`, or NULL if the error code is not known.

### `ykneomgr_init ()`

```
ykneomgr_rc  
ykneomgr_init (ykneomgr_dev **dev);
```

Create a YubiKey NEO device handle. The handle must be deallocated using `ykneomgr_done()` when you no longer need it.

### Parameters

<code>dev</code>	pointer to newly allocated device handle.	
------------------	--	--

### Returns

On success, `YKNEOMGR_OK` (integer 0) is returned, or another `ykneomgr_rc` error code.

### `ykneomgr_done ()`

```
void  
ykneomgr_done (ykneomgr_dev *dev);
```

Release all resources allocated to a YubiKey NEO device handle.

### Parameters

<code>dev</code>	device handle to deallocate, created by <code>ykneomgr_init()</code> .	
------------------	---	--

### `ykneomgr_list_devices ()`

```
ykneomgr_rc  
ykneomgr_list_devices (ykneomgr_dev *dev,  
                        char *devicestr,  
                        size_t *len);
```

List devices.

### Parameters

---



dev	a <b>ykneomgr_dev</b> device handle.	
devicestr	output buffer to hold string, or <b>NULL</b> .	
len	on input length of <i>devicestr</i> buffer, on output holds output length	

### Returns

On success, **YKNEOMGR\_OK** (integer 0) is returned, or another **ykneomgr\_rc** error code.

### ykneomgr\_connect ()

```
ykneomgr_rc  
ykneomgr_connect (ykneomgr_dev *dev,  
                  const char *name);
```

Establish connection to a named PCSC device and verify that it has the YubiKey OTP applet. The *name* string should be a PCSC device name; you can use **ykneomgr\_list\_devices()** to list connected devices.

### Parameters

dev	a <b>ykneomgr_dev</b> device handle.	
name	input string with device name to connect to.	

### Returns

On success, **YKNEOMGR\_OK** (integer 0) is returned, when no device could be found **YKNEOMGR\_NO\_DEVICE** is returned, or another **ykneomgr\_rc** error code.

### ykneomgr\_discover ()

```
ykneomgr_rc  
ykneomgr_discover (ykneomgr_dev *dev);
```

Discover and establish connection to the first found YubiKey NEO. A YubiKey NEO is identified by having the YubiKey OTP applet installed, i.e., a connect followed by attempting to select the YubiKey OTP applet.

### Parameters

dev	a <b>ykneomgr_dev</b> device handle.	
-----	--------------------------------------	--

### Returns

On success, **YKNEOMGR\_OK** (integer 0) is returned, when no device could be found **YKNEOMGR\_NO\_DEVICE** is returned, otherwise another **ykneomgr\_rc** error code is returned.



### ykneomgr\_discover\_match ()

```
ykneomgr_rc  
ykneomgr_discover_match (ykneomgr_dev *dev,  
                          const char *match);
```

Discover and establish connection to the first found YubiKey NEO that has a card reader name matching *match* . A YubiKey NEO is identified by having the YubiKey OTP applet installed, i.e., a connect followed by attempting to select the YubiKey OTP applet. If *match* is NULL, then the first YubiKey NEO device detected will be used.

#### Parameters

dev	a <b>ykneomgr_dev</b> device handle.	
match	substring to match card reader for, or <b>NULL</b> .	

#### Returns

On success, **YKNEOMGR\_OK** (integer 0) is returned, when no device could be found **YKNEOMGR\_NO\_DEVICE** is returned, or another **ykneomgr\_rc** error code.

Since 0.1.4

### ykneomgr\_get\_version\_major ()

```
uint8_t  
ykneomgr_get_version_major (ykneomgr_dev *dev);
```

Get major version of a YubiKey NEO. Versions are in the form of MAJOR.MINOR.BUILD, for example 3.0.4, in which case this function would return 3.

#### Parameters

dev	a <b>ykneomgr_dev</b> device handle.	
-----	--------------------------------------	--

#### Returns

the YubiKey NEO major version number.

### ykneomgr\_get\_version\_minor ()

```
uint8_t  
ykneomgr_get_version_minor (ykneomgr_dev *dev);
```

Get minor version of a YubiKey NEO. Versions are in the form of MINOR.MINOR.BUILD, for example 3.0.4, in which case this function would return 0.

#### Parameters

---



dev

a **ykneomgr\_dev** device handle.

### Returns

the YubiKey NEO minor version number.

### ykneomgr\_get\_version\_build ()

```
uint8_t  
ykneomgr_get_version_build (ykneomgr_dev *dev);
```

Get build version of a YubiKey NEO. Versions are in the form of BUILD.MINOR.BUILD, for example 3.0.4, in which case this function would return 4.

### Parameters

dev

a **ykneomgr\_dev** device handle.

### Returns

the YubiKey NEO build version number.

### ykneomgr\_get\_mode ()

```
uint8_t  
ykneomgr_get_mode (ykneomgr_dev *dev);
```

Get mode of a YubiKey NEO.

### Parameters

dev

a **ykneomgr\_dev** device handle.

### Returns

the YubiKey NEO device mode.

### ykneomgr\_get\_serialno ()

```
uint32_t  
ykneomgr_get_serialno (ykneomgr_dev *dev);
```

Get serial number of a YubiKey NEO, if visible.

### Parameters

dev

a **ykneomgr\_dev** device handle.



## Returns

the YubiKey NEO serial number, or 0 if not visible.

## ykneomgr\_modeswitch ()

```
ykneomgr_rc  
ykneomgr_modeswitch (ykneomgr_dev *dev,  
                     uint8_t mode);
```

Mode switch a YubiKey NEO.

## Parameters

dev	a <b>ykneomgr_dev</b> device handle.	
mode	new mode to switch the device into	

## Returns

On success, **YKNEOMGR\_OK** (integer 0) is returned, or another **ykneomgr\_rc** error code.

## ykneomgr\_authenticate ()

```
ykneomgr_rc  
ykneomgr_authenticate (ykneomgr_dev *dev,  
                      const uint8_t *key);
```

Authenticate to the device, to prepare for privileged function access.

## Parameters

dev	a <b>ykneomgr_dev</b> device handle.	
key	Double-DES key in binary, 16 bytes	

## Returns

On success, **YKNEOMGR\_OK** (integer 0) is returned, or another **ykneomgr\_rc** error code.

## ykneomgr\_applet\_list ()

```
ykneomgr_rc  
ykneomgr_applet_list (ykneomgr_dev *dev,  
                     char *appletstr,  
                     size_t *len);
```

List installed applets.

---



**Parameters**



dev	a <b>ykneomgr_dev</b> device handle.	
appletstr	output buffer to hold string, or <b>NULL</b> .	
len	on input length of <i>appletstr</i> buffer, on output holds output length	

### Returns

On success, **YKNEOMGR\_OK** (integer 0) is returned, or another **ykneomgr\_rc** error code.

### ykneomgr\_applet\_delete ()

```
ykneomgr_rc
ykneomgr_applet_delete (ykneomgr_dev *dev,
                        const uint8_t *aid,
                        size_t aidlen);
```

Delete specified applet.

### Parameters

dev	a <b>ykneomgr_dev</b> device handle.	
aid	aid to delete.	
aidlen	length of <i>aid</i> buffer.	

### Returns

On success, **YKNEOMGR\_OK** (integer 0) is returned, or another **ykneomgr\_rc** error code.

### ykneomgr\_applet\_install ()

```
ykneomgr_rc
ykneomgr_applet_install (ykneomgr_dev *dev,
                        const char *capfile);
```

Install specified applet.

### Parameters

dev	a <b>ykneomgr_dev</b> device handle.	
capfile	string with path filename to CAP file	

### Returns

On success, **YKNEOMGR\_OK** (integer 0) is returned, or another **ykneomgr\_rc** error code.

---



**ykneomgr\_send\_apdu ()**

```
ykneomgr_rc
ykneomgr_send_apdu (ykneomgr_dev *dev,
                    const uint8_t *send,
                    size_t sendlen,
                    uint8_t *recv,
                    size_t *recvlen);
```

Send an arbitrary apdu to the device.

**Parameters**

dev	a <b>ykneomgr_dev</b> device handle.	
send	apdu to send	
sendlen	length of send buffer	
recv	response apdu	
recvlen	length of recv buffer	

**Returns**

On success, **YKNEOMGR\_OK** (integer 0) is returned, or another **ykneomgr\_rc** error code. *recvlen* will be set to the length of the data in *recv*.

**Types and Values**

**1.2 ykneomgr-types**

ykneomgr-types —

**Types and Values**

enum	<b>ykneomgr_rc</b>
enum	<b>ykneomgr_initflags</b>
typedef	<b>ykneomgr_dev</b>

**Object Hierarchy**

**Description**

**Functions**

**Types and Values**

**enum ykneomgr\_rc**

Error codes.



Members

YKNEOMGR_OK	Success.
YKNEOMGR_MEMORY_ERROR	Memory error.
YKNEOMGR_NO_DEVICE	No device found.
YKNEOMGR_TOO_MANY_DEVICES	Too many devices found.
YKNEOMGR_BACKEND_ERROR	Input/Output error.

enum ykneomgr\_initflags

Flags passed to `ykneomgr_global_init()`.

Members

YKNEOMGR_DEBUG	Print debug messages.
----------------	-----------------------

ykneomgr\_dev

```
typedef struct ykneomgr_dev ykneomgr_dev;
```

1.3 ykneomgr-version

ykneomgr-version —

Functions

`const char *` | `ykneomgr_check_version ()`

Types and Values

#define	YKNEOMGR_VERSION_STRING
#define	YKNEOMGR_VERSION_NUMBER
#define	YKNEOMGR_VERSION_MAJOR
#define	YKNEOMGR_VERSION_MINOR
#define	YKNEOMGR_VERSION_PATCH



## Object Hierarchy

---

### Description

### Functions

#### **ykneomgr\_check\_version ()**

```
const char~*
ykneomgr_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 **YKNEOMGR\_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

#### **YKNEOMGR\_VERSION\_STRING**

```
#define YKNEOMGR_VERSION_STRING "0.1.7"
```

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

#### **YKNEOMGR\_VERSION\_NUMBER**

```
#define YKNEOMGR_VERSION_NUMBER 0x000107
```

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.

#### **YKNEOMGR\_VERSION\_MAJOR**

```
#define YKNEOMGR_VERSION_MAJOR 0
```

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.

---



**YKNEOMGR\_VERSION\_MINOR**

```
#define YKNEOMGR_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.

**YKNEOMGR\_VERSION\_PATCH**

```
#define YKNEOMGR_VERSION_PATCH 7
```

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

### Y

ykneomgr\_applet\_delete, 9  
ykneomgr\_applet\_install, 9  
ykneomgr\_applet\_list, 7  
ykneomgr\_authenticate, 7  
ykneomgr\_check\_version, 12  
ykneomgr\_connect, 4  
ykneomgr\_dev, 11  
ykneomgr\_discover, 4  
ykneomgr\_discover\_match, 5  
ykneomgr\_done, 3  
ykneomgr\_get\_mode, 6  
ykneomgr\_get\_serialno, 6  
ykneomgr\_get\_version\_build, 6  
ykneomgr\_get\_version\_major, 5  
ykneomgr\_get\_version\_minor, 5  
ykneomgr\_global\_done, 2  
ykneomgr\_global\_init, 2  
ykneomgr\_init, 3  
ykneomgr\_initflags, 11  
ykneomgr\_list\_devices, 3  
ykneomgr\_modeswitch, 7  
ykneomgr\_rc, 10  
ykneomgr\_send\_apdu, 10  
ykneomgr\_strerror, 2  
ykneomgr\_strerror\_name, 2  
YKNEOMGR\_VERSION\_MAJOR, 12  
YKNEOMGR\_VERSION\_MINOR, 13  
YKNEOMGR\_VERSION\_NUMBER, 12  
YKNEOMGR\_VERSION\_PATCH, 13  
YKNEOMGR\_VERSION\_STRING, 12

---