Utils API provide a variety of frequently used general purpose functionalities for the Talaria TWO platform.

# Features and Limitations

This component provides the following features:

1. File system initialization and file access.
2. System utility function like software reset.
3. Wi-Fi connection profile initialization.

# Header file/s

1. components/utils/inc/fs\_defines.h
2. components/utils/inc/fs\_utils.h
3. components/utils/inc/utils.h
4. components/utils/inc/wifi\_utils.h
5. components/utils/inc/net\_utils.h

# API Reference

## utils\_mount\_rootfs

### Overview

This API mounts the root FS. Root FS is used by the user application to store application specific files like certificates, partition file (if SSBL is used), FOTA configuration file and others.

This is in-fact a wrapper function for os\_mount() API.

### Definition

|  |
| --- |
| int  utils\_mount\_rootfs (void) |

### Parameters

None

### Return

Success: 0

Error: -1

## utils\_is\_file\_present

### Overview

Used to check if a file is present in at a specified path or not in root FS.

### Definition

|  |
| --- |
| int  utils\_is\_file\_present (char \*path) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *path* | Path of the file |

Table : utils\_is\_file\_present – parameters

### Return

If the file is present – 1

If the file is not present - 0

## utils\_file\_size\_get

### Overview

Used to get the size of the file at a specified path in root FS.

### Definition

|  |
| --- |
| int  utils\_file\_size\_get (char \*path) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *path* | Path of the file |

Table : utils\_file\_size\_get - parameters

### Return

File size: >= 0

Error: < 0

## utils\_file\_get

### Overview

Used to get the content of a file into a buffer.

**Note**: If the buffer return is not freed, it will result in a memory leak.

### Definition

|  |
| --- |
| char \*  utils\_file\_get (char \*path, int \*len) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *path* | Path of the file |
| *Len* | Used to return the length of the file |

Table : utils\_file\_get - parameters

### Return

Success : Pointer to buffer having the content of the file.

Failure : NULL

## utils\_file\_store

### Overview

Used to store the content of a buffer into a file.

**Note**: There is no way to amend the data with this API. This will store the data from the beginning of the file, replacing any older data if present.

### Definition

|  |
| --- |
| int  utils\_file\_store (char \*path, char \*buf, int \*len) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *path* | Path of the file |
| *buf* | Buffer having data to be store in the specified file |
| *Len* | Lenth/size of data present in the buffer |

Table : utils\_file\_store – parameters

### Return

Success : Pointer to buffer having the content of the file.

Failure : NULL

## reset\_device

### Overview

Used to soft reset the device.

### Definition

|  |
| --- |
| void  reset\_device (void) |

## show\_heap

### Overview

Used to print the heap available runtime and is used for debugging.

### Definition

|  |
| --- |
| void  show\_heap (const char \*function, int line\_number) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *function* | Name of the function from where this is getting called |
| *line\_number* | Line number at which this function is getting called |

Table : show\_heap – parameters

## network\_profile\_new\_from\_ssid\_pw

### Overview

Used to allocate new network profile structure from SSID and Passphrase.

### Definition

|  |
| --- |
| int  network\_profile\_new\_from\_ssid\_pw(struct network\_profile \*\*np\_ret, char \*ssid, char \*passphrase) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *np\_ret* | Newly allocated network\_profile structure |
| *ssid* | SSID of AP |
| *passphrase* | Passphrase of AP |

Table : network\_profile\_new\_from\_ssid\_pw – parameters

### Return

Success: 0

Error: Negative error number

## network\_profile\_new\_from\_ssid\_bssid\_pw

### Overview

Used to allocate new network profile structure from SSID, BSSID and passphrase.

### Definition

|  |
| --- |
| int  network\_profile\_new\_from\_ssid\_bssid\_pw(struct network\_profile \*\*np\_ret, char \*ssid, uint8\_t \*bssid, char \*passphrase) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *np\_ret* | Newly allocated network\_profile structure |
| *ssid* | SSID of AP |
| *bssid* | BSSID of AP |
| *passphrase* | Passphrase of AP |

Table : network\_profile\_new\_from\_ssid\_bssid\_pw – parameters

### Return

Success: 0

Error: Negative error number

## network\_profile\_new\_from\_boot\_args

### Overview

Used to allocate new network profile structure from bootargs.

### Definition

|  |
| --- |
| int  network\_profile\_new\_from\_boot\_args(struct network\_profile \*\*np\_ret) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *np\_ret* | Newly allocated network\_profile structure |

Table : network\_profile\_new\_from\_boot\_args – parameters

### Return

Success: 0

Error: Negative error number

## is\_valid\_ip

### Overview

Used to verify whether the argument passed is a valid IP address or not.

### Definition

|  |
| --- |
| int  is\_valid\_ip(char \*ip\_str\_in); |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *ip\_str\_in* | String form of IP address, which is to be validated |

Table : is\_valid\_ip – parameters

### Return

Success: 0

Error: 1

## wifi\_connect\_to\_network

### Overview

Used to connect to a network.

### Definition

|  |
| --- |
| int wifi\_connect\_to\_network (struct wcm\_handle \*\*p\_wcm, int timeout\_secs, bool \*conn\_status); |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *p\_wcm* | Double pointer to @ref wcm\_handle |
| *timeout\_secs* | Timeout to wait for wifi connection.  -1 = infite, 0 = no wait, >0 = wait\_secs |
| *conn\_status* | Status of connection: Connected or Disconnected |

Table : wifi\_connect\_to\_network – parameters

### Return

Success: 0

Error: Negative Error Code

# Application Example

For the example code, refer: *components/utils*.