

# BatteryMonitor

Generated by Doxygen 1.7.3

Thu Oct 6 2011 13:42:06



# Contents

<b>1</b>	<b>BatteryMonitor License</b>	<b>1</b>
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Namespace Documentation</b>	<b>7</b>
4.1	BatteryMonitor Namespace Reference . . . . .	7
4.1.1	Detailed Description . . . . .	9
4.1.2	Function Documentation . . . . .	9
4.1.2.1	addNotifyIcon . . . . .	9
4.1.2.2	cleanUp . . . . .	9
4.1.2.3	errOut . . . . .	9
4.1.2.4	getBatteryInformation . . . . .	9
4.1.2.5	initDevice . . . . .	10
4.1.2.6	setupWindow . . . . .	10
4.1.2.7	setupWindowClass . . . . .	10
4.1.2.8	shutdown . . . . .	10
4.1.2.9	wndProc . . . . .	11
4.1.3	Variable Documentation . . . . .	11
4.1.3.1	battery . . . . .	11
4.1.3.2	batterylifetime . . . . .	11
4.1.3.3	classname . . . . .	11
4.1.3.4	devicepath . . . . .	12
4.1.3.5	instance . . . . .	12
4.1.3.6	WM_TASKBARCREATED . . . . .	12
<b>5</b>	<b>File Documentation</b>	<b>13</b>
5.1	main.h File Reference . . . . .	13
5.1.1	Detailed Description . . . . .	15
5.2	mainres.h File Reference . . . . .	15
5.2.1	Detailed Description . . . . .	16



# Chapter 1

## BatteryMonitor License

Copyright (c) 2011, Florian Satzger <http://batterymonitor.de.vu> All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the author nor the names of the contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL FLORIAN SATZGER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

<a href="#">BatteryMonitor</a> (Root namespace) . . . . .	7
---	---





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

<b>commandline.cpp</b>	??
<b>commandline.h</b>	??
<b>fileconfig.cpp</b>	??
<b>fileconfig.h</b>	??
<b>main.cpp</b>	??
<a href="#">main.h</a> (Public funktions and variables )	13
<a href="#">mainres.h</a> (Resources of <a href="#">BatteryMonitor</a> )	15



## Chapter 4

# Namespace Documentation

### 4.1 BatteryMonitor Namespace Reference

Root namespace.

#### Functions

- void [addNotifyIcon](#) (HWND hwnd)  
*Adds an icon to the taskbar.*
- void [cleanUp](#) (HWND hwnd)  
*Memory and handler cleanups, taskbar icon removal.*
- void [errOut](#) (void)  
*Error output.*
- BATTERY\_INFORMATION [getBatteryInformation](#) ()  
*retrieves battery informations from the system.*
- void [initDevice](#) (void)  
*Retrieves the path of the battery device.*
- HWND [setupWindow](#) (int nCmdShow)  
*Shows the main window.*
- long [setupWindowClass](#) (void)  
*registers the window class stored in *classname*.*
- void [showBatteryInfo](#) (HWND hwnd)  
*Shows a balloon tip with battery information.*

- void `shutdown` (void)  
*Tries to shut down the computer.*
- LRESULT CALLBACK `wndProc` (HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam)  
*The window procedure.*

### Variables

- double `battery` = 0  
*Battery charge.*
- long `batterylifetime` = -1  
*Remaining battery lifetime in seconds.*
- HBRUSH `blue`  
*Blue brush.*
- char `classname` [14]  
*Name of registered window class.*
- char \* `devicepath`  
*Path to battery device.*
- HBRUSH `green`  
*Green brush.*
- HINSTANCE `instance`  
*Application instance.*
- HBRUSH `red`  
*Red brush.*
- bool `shutdown_enabled` = false  
*Indicates, if shutdown was invoked by `BatteryMonitor`.*
- UINT `WM_TASKBARCREATED` = 0  
*Window message for taskbarcreated-event.*
- HBRUSH `yellow`  
*Yellow brush.*

### 4.1.1 Detailed Description

Root namespace.

### 4.1.2 Function Documentation

#### 4.1.2.1 void BatteryMonitor::addNotifyIcon ( HWND *hwnd* )

Adds an icon to the taskbar.

##### Parameters

<i>in</i>	<i>hwnd</i>	The window handle of the window whose window procedure receives the events occurring.
-----------	-------------	---

##### See also

Windows API

Definition at line 234 of file main.cpp.

#### 4.1.2.2 void BatteryMonitor::cleanUp ( HWND *hwnd* )

Memory and handler cleanups, taskbar icon removal.

##### Parameters

<i>in</i>	<i>hwnd</i>	The window handle of the window associated with the taskbar icon.
-----------	-------------	---

Definition at line 92 of file main.cpp.

#### 4.1.2.3 void BatteryMonitor::errOut ( void )

Error output.

If [BatteryMonitor](#) is compiled in debug configuration (with a `DEBUG` defined) , a messagebox containing the error text is shown.

Definition at line 123 of file main.cpp.

#### 4.1.2.4 BATTERY\_INFORMATION BatteryMonitor::getBatteryInformation ( )

retrieves battery informations from the system.

This funktion retrieves battery information from the system. If any error occurs, [errOut\(\)](#) will be called and the `DesignedCapacity` member of the returned struct is set to 0.

**Returns**

Battery informations

Definition at line 328 of file main.cpp.

**4.1.2.5 void BatteryMonitor::initDevice ( void )**

Retrieves the path of the battery device.

This function retrieves the path of the battery device and stores it into [devicepath](#).

Definition at line 301 of file main.cpp.

**4.1.2.6 HWND BatteryMonitor::setupWindow ( int nCmdShow )**

Shows the main window.

This funktion creates a window of the class stored in [classname](#). This window is automatically set click-through, transparent and topmost.

**Parameters**

<code>in</code>	<code>nCmdShow</code>	Can override the default visibility-parameters
-----------------	-----------------------	--

**Returns**

The window handle of the newly created window

**See also**

Windows API

Definition at line 66 of file main.cpp.

**4.1.2.7 long BatteryMonitor::setupWindowClass ( void )**

registers the window class stored in [classname](#).

**Returns**

The error code of RegisterClassEx, the corresponding windows api call.

Definition at line 105 of file main.cpp.

**4.1.2.8 void BatteryMonitor::shutdown ( void )**

Tries to shut down the computer.

This function is called if remaining battery lifetime is less than 5 minutes.

**See also**

[addNotifyIcon\(HWND hwnd\)](#)

Definition at line 273 of file main.cpp.

**4.1.2.9 LRESULT CALLBACK BatteryMonitor::wndProc ( HWND *hwnd*, UINT *msg*, WPARAM *wParam*, LPARAM *lParam* )**

The window procedure.

Here, all window logic is done such as mouse handling and so on.

**See also**

Windows API for further information

Definition at line 133 of file main.cpp.

**4.1.3 Variable Documentation****4.1.3.1 double BatteryMonitor::battery = 0**

Battery charge.

Battery charge, positive if battery is discharging, negative if charging. (e.g. if 75% and changing, battery is 0.75). If battery status cannot be determined, battery is 2.

Definition at line 142 of file main.h.

**4.1.3.2 long BatteryMonitor::batterylifetime = -1**

Remaining battery lifetime in seconds.

Remaining battery lifetime in seconds. -1 is status cannot be determined.

Definition at line 148 of file main.h.

**4.1.3.3 char BatteryMonitor::classname[14]**

Name of registered window class.

**See also**

Windows API

Definition at line 154 of file main.h.

#### 4.1.3.4 `char* BatteryMonitor::devicepath`

Path to battery device.

The battery information can be read from a pseudo-file, which path is stored here.

##### See also

Windows API

Definition at line 115 of file main.h.

#### 4.1.3.5 `HINSTANCE BatteryMonitor::instance`

Application instance.

##### See also

Windows API

Definition at line 134 of file main.h.

#### 4.1.3.6 `UINT BatteryMonitor::WM_TASKBARCREATED = 0`

Window message for taskbarcreated-event.

This message must be registered with windows, so that the application gets informed if the taskbar is (re-)created. This is necessary to update the taskbar-icon.

Definition at line 162 of file main.h.



## Chapter 5

# File Documentation

### 5.1 main.h File Reference

public funktions and variables

```
#include <initguid.h>
#include <stdlib.h>
#include <windows.h>
#include <winuser.h>
#include <stdio.h>
#include <setupapi.h>
#include <ddk/batclass.h>
#include "mainres.h"
```

#### Namespaces

- namespace [BatteryMonitor](#)  
*Root namespace.*

#### Defines

- #define [\\_WIN32\\_IE](#) 0x600  
*Sets windows version information for some Apis.*
- #define [UPDATE\\_BATTERY](#) 1  
*Window message for updating battery status.*
- #define [WINVER](#) 0x0501

*Sets windows version information for some Apis.*

## Functions

- void [BatteryMonitor::addNotifyIcon](#) (HWND hwnd)  
*Adds an icon to the taskbar.*
- void [BatteryMonitor::cleanUp](#) (HWND hwnd)  
*Memory and handler cleanups, taskbar icon removal.*
- void [BatteryMonitor::errOut](#) (void)  
*Error output.*
- BATTERY\_INFORMATION [BatteryMonitor::getBatteryInformation](#) ()  
*retrieves battery informations from the system.*
- void [BatteryMonitor::initDevice](#) (void)  
*Retrieves the path of the battery device.*
- HWND [BatteryMonitor::setupWindow](#) (int nCmdShow)  
*Shows the main window.*
- long [BatteryMonitor::setupWindowClass](#) (void)  
*registers the window class stored in *classname*.*
- void [BatteryMonitor::showBatteryInfo](#) (HWND hwnd)  
*Shows a balloon tip with battery information.*
- void [BatteryMonitor::shutdown](#) (void)  
*Tries to shut down the computer.*
- LRESULT CALLBACK [BatteryMonitor::wndProc](#) (HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam)  
*The window procedure.*

## Variables

- double [BatteryMonitor::battery](#) = 0  
*Battery charge.*
- long [BatteryMonitor::batterylifetime](#) = -1  
*Remaining battery lifetime in seconds.*
- HBRUSH [BatteryMonitor::blue](#)

*Blue brush.*

- char [BatteryMonitor::classname](#) [14]  
*Name of registered window class.*
- char \* [BatteryMonitor::devicepath](#)  
*Path to battery device.*
- HBRUSH [BatteryMonitor::green](#)  
*Green brush.*
- HINSTANCE [BatteryMonitor::instance](#)  
*Application instance.*
- HBRUSH [BatteryMonitor::red](#)  
*Red brush.*
- bool [BatteryMonitor::shutdown\\_enabled](#) = false  
*Indicates, if shutdown was invoked by [BatteryMonitor](#).*
- UINT [BatteryMonitor::WM\\_TASKBARCREATED](#) = 0  
*Window message for taskbarcreated-event.*
- HBRUSH [BatteryMonitor::yellow](#)  
*Yellow brush.*

### 5.1.1 Detailed Description

public funktions and variables

#### Author

F.Satzger

#### Date

22. Oktober 2010, 10:59

Definition in file [main.h](#).

## 5.2 mainres.h File Reference

Resources of [BatteryMonitor](#).

**Defines**

- #define [MAIN\\_ICON](#) 1  
*The icon of [BatteryMonitor](#).*

**5.2.1 Detailed Description**

Resources of [BatteryMonitor](#).

**Author**

F.Satzger

**Date**

23. Oktober 2010, 11:11

Definition in file [mainres.h](#).

# Index

- addNotifyIcon
  - BatteryMonitor, 9
- battery
  - BatteryMonitor, 11
- batterylifetime
  - BatteryMonitor, 11
- BatteryMonitor, 7
  - addNotifyIcon, 9
  - battery, 11
  - batterylifetime, 11
  - classname, 11
  - cleanUp, 9
  - devicepath, 11
  - errOut, 9
  - getBatteryInformation, 9
  - initDevice, 10
  - instance, 12
  - setupWindow, 10
  - setupWindowClass, 10
  - shutdown, 10
  - WM\_TASKBARCREATED, 12
  - wndProc, 11
- classname
  - BatteryMonitor, 11
- cleanUp
  - BatteryMonitor, 9
- devicepath
  - BatteryMonitor, 11
- errOut
  - BatteryMonitor, 9
- getBatteryInformation
  - BatteryMonitor, 9
- initDevice
  - BatteryMonitor, 10
- instance
  - BatteryMonitor, 12
- main.h, 13
- mainres.h, 15
- setupWindow
  - BatteryMonitor, 10
- setupWindowClass
  - BatteryMonitor, 10
- shutdown
  - BatteryMonitor, 10
- WM\_TASKBARCREATED
  - BatteryMonitor, 12
- wndProc
  - BatteryMonitor, 11