



# **CeRegCMD**

**command line CE registry tool**  
**part of CeRegEditor**

**User Manual**  
ver.1.0

## ABOUT

The **CeRegCMD** is command line registry editor for any device (PDA) powered by WindowsMobile (WM5, WM6), Windows CE, PocketPC 2003. It is based on the *CeRegEditor* modules.

The **CeRegCMD** is executed from command line without any GUI. Like the *CeRegEditor*, it runs on standalone PC but operates on PDA registry.

As tool without GUI, **CeRegCMD** can be used in scripts or by programmers in their applications. It is free for private and commercial use.

The **CeRegCMD** is created to make use of PDA's registry as easy as possible ;)

For questions and support please mail me at [ceregeditor@mdsoft.pl](mailto:ceregeditor@mdsoft.pl)

**mdSoft :)**

## PARAMETERS OVERVIEW

The parameters are listed after execution of *ceregcmd.exe* without any parameter.

```
CeRegCMD ver: 1.0.0.0 - CeRegEditor command line tool
Copyright(C) 2007 mdSoft, http://ceregeditor.mdsoft.pl

Usage:
CeRegCMD [-rwh] [-q] [-out FILE_NAME] [-in FILE_NAME] [-msg FILE_NAME]
CeRegCMD [-rwh] [-q] [-kn KEY_NAME] [-vn VALUE_NAME] [-vt [VALUE_TYPE]]
          [-vd [VALUE_DATA]] [-out FILE_NAME] [-msg FILE_NAME]

Commands:
-r                read from the registry
-w                write to the registry

Options:
-q                quiet mode
-out FILE_NAME   output file for read results, in read mode (-r)
-in FILE_NAME    input file to read from, in write mode (-w)
-msg FILE_NAME   redirect messages to FILE_NAME
-oe              output file encoding. Valid encodings: ANSI,UNICODE,
                  UNICODE_B,UTF8
-vk KEY_NAME     key name
-vn VALUE_NAME   value name
-vt VALUE_TYPE   type of value. Valid types: REG_SZ, REG_MULTI_SZ,
                  REG_DWORD, REG_BINARY
-vd VALUE_DATA   data to be stored with the specified VALUE_NAME
```

| PARAMETER             | REQUIRED | DESCRIPTION  |
|-----------------------|----------|--|
| <b>-r</b>             | Yes      | Sets <b>read</b> (from registry) mode  |
| <b>-w</b>             | Yes      | Sets <b>write</b> (to registry) mode   |
| <b>-q</b>             | No       | Quite mode. No messages are displayed.   |
| <b>-out</b> FILE_NAME | No       | Name of file for reading results. Default, results are written to console window.  |
| <b>-in</b> FILE_NAME  | No       | Input file. Can be used in read and write modes. Used for data input.  |
| <b>-msg</b> FILE_NAME | No       | Redirects all messages to file with <i>FILE_NAME</i> .   |
| <b>-oe</b> ENCODING   | No       | Encoding of <b>out</b> file. Valid encodings: <i>ANSI</i> , <i>UNICODE</i> , <i>UNICODE_B</i> (bigEndian), <i>UTF8</i> .<br>Default encoding is ANSI or (if used) encoding used by <b>in</b> file. |

|                              |                                    |   |
|------------------------------|------------------------------------|---|
| <p><b>-kn</b> KEY_NAME</p>   | <p>Yes/<br/>no with <b>-in</b></p> | <p>Registry key name to read from or write to. In <i>read</i> mode key must exist; In <i>write</i> mode, key is created if doesn't exist.</p> <p>If input file (<b>-in</b>) is used, <b>-kn</b> can be omitted. In that case key name is read from <u>first line</u> of input line.</p> <p>Key name is given with root key. Both short and long root key name can be used;<br/>- HKLM or HKEY_LOCAL_MACHINE</p>   |
| <p><b>-vn</b> VALUE_NAME</p> | <p>Yes/<br/>no with <b>-in</b></p> | <p>Value name to read from or write to. In <i>read</i> mode value must exist; In <i>write</i> mode, value is created if doesn't exist.</p> <p>If input file (<b>-in</b>) is used, <b>-vn</b> can be omitted. In that case value name is read from <u>third line</u> of input line.</p> <p>Key name is given with root key. Both short and long root key name can be used;<br/>- HKLM or HKEY_LOCAL_MACHINE</p>  |
| <p><b>-vt</b> VALUE_TYPE</p> | <p>No</p>                          | <p>Type of value. <b>-vt</b> is required only for new values which are going to be created.</p> <p>If omitted in <i>write</i> mode, value type is get from registry.</p> <p>If used in <i>read</i> mode, the CeRegCMD returns value type.</p> <p>If input file (<b>-in</b>) is used, value type is read from <u>second line</u> of input file. For read mode use 0 or REG_UNKNOWN.</p> <p>Valid value types: REG_DWORD, REG_SZ, REG_MULTI_SZ, REG_BINARY.</p> |
| <p><b>-vd</b> VALUE_DATA</p> | <p>No</p>                          | <p>Value data - data to be stored with the specified value name. Used in write mode.</p> <p>If input file (<b>-in</b>) is used, value data is read from <u>fourth and next lines</u> of input file.</p>   |

|  |  |   |
|--|--|---|
|  |  | <p>REG_DWORD: for this type, numbers can be passed as <i>decimal</i> or <i>hexadecimal</i> values. For hexadecimals, syntax of Pascal or C++ can be used - \$FF or 0xFF.</p> <p>REG_BINARY: each byte must given by two characters hex value – 01 0F FF</p> |
|--|--|---|

## EXECUTION EXAMPLES

```
ceregcmd -w -kn HKLM\Test -vn v1 -vt REG_DWORD -vd 0x01 - set value with  
name v1 within key HKLM\Test to number 1
```

same with input file:

```
ceregcmd -w -in d:\in_file.txt
```

```
in_file.txt:  
HKLM\Test1  
REG_DWORD  
v1  
0x01
```

```
ceregcmd -w -kn HKLM\Test -vn v2 -vt REG_SZ -vd test_string - set value with  
name v2 within key HKLM\Test to text "test_string"
```

```
ceregcmd -r -in d:\in_file2.txt -out d:\out_file.txt - read from key and  
value given in in_file2.txt, save results to out_file.txt
```

```
in_file2.txt:  
HKLM\Test  
0  
v2
```

```
out_file.txt:  
test_string
```

## EXIT CODES

After finish, the CeRegCMD always sets *exit code*. If all operations succeed exit code is set to 0 (zero), if not, nonzero value is returned.

Possible error codes:

| NAME                    | VALUE<br>(decimal) | DESCRIPTION   |
|-------------------------|--------------------|---|
| CMD_OK                  | 000                | The operation completed successfully                      |
| CMD_NO_PARAM            | 001                | Missing one of required parameters                        |
| CMD_NO_FILE             | 002                | Specified file is missing on disc                         |
| CMD_NO_DATA             | 003                | Required data is not set                                  |
| CMD_INVALID_PARAM       | 020                | Invalid(unknown) parameter name                           |
| CMD_INVALID_PARAM_VALUE | 021                | Invalid value for parameter                               |
| CMD_INVALID_FILE        | 022                | Invalid file structure                                    |
| CMD_INVALID_DATA        | 023                | Invalid or unsupported data                               |
| CMD_ERROR_REG_OPEN      | 040                | Registry open error                                       |
| CMD_ERROR_REG_READ      | 041                | Error while reading from registry                         |
| CMD_ERROR_REG_WRITE     | 042                | Error while writing to registry                           |
| CMD_PDA_NO_CONNECTION   | 060                | Cannot connect to PDA                                     |
| CMD_UNKNOWN             | 200                | Unknown error, mostly, error raised by internal API calls |