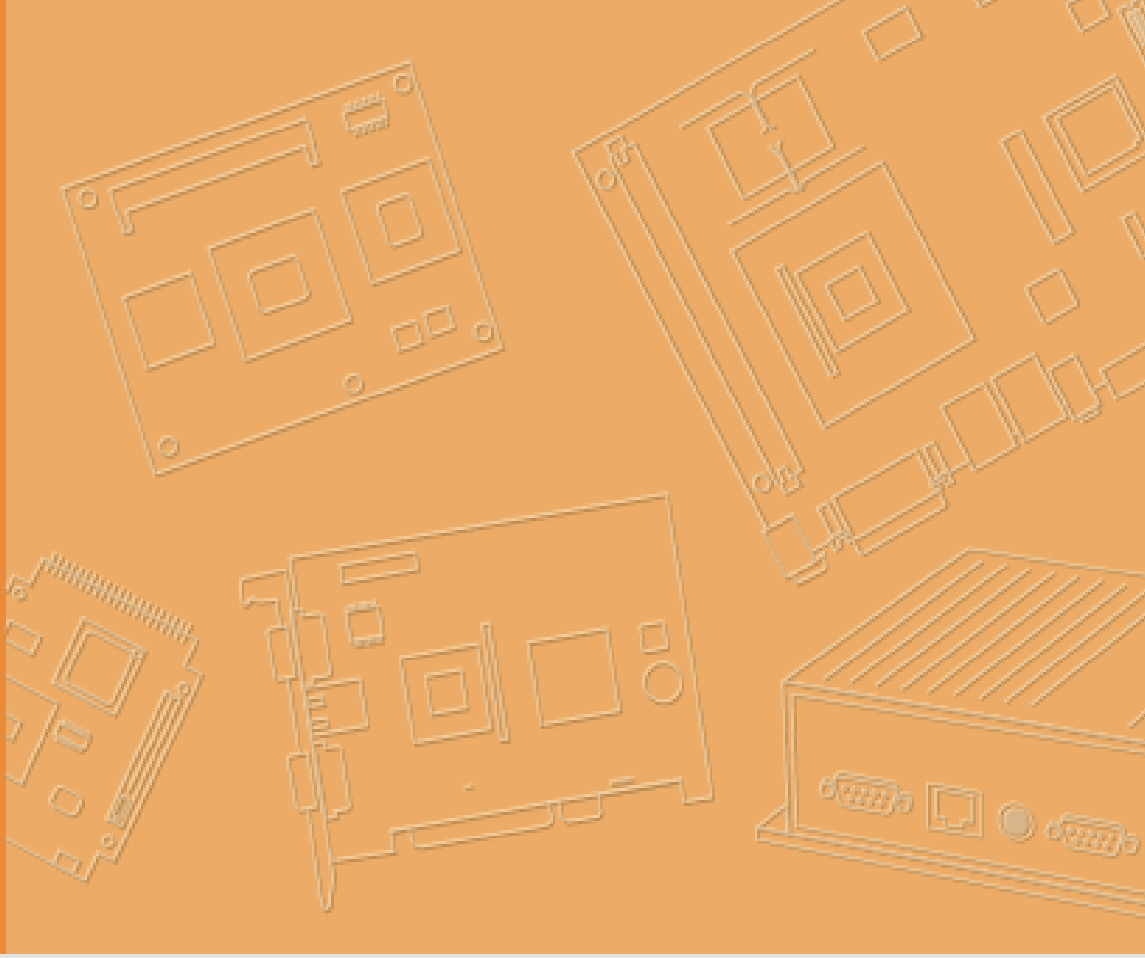
Manual



# DLoG Software Keyboard

English V3.11

ADVANTECH DLOG

This manual contains a detailed description of the product and we have made every effort to make it as accurate as possible. However, this is not a guarantee of the features or the functionality of the product. We reserve the right to modify the contents of this document at any time and without prior notice. Because we at Advantech are constantly striving to improve this product, we cannot guarantee that previous or subsequent releases of the product will coincide in every respect with the product description given in this manual.

Advantech Co., Ltd. assumes no liability for technical inaccuracies, typographic errors or faults in this documentation. Advantech also assumes no liability for damages, which refer directly or indirectly to the delivery, performance or usage of this material.

The software and hardware designations used in this documentation are in most cases also registered trademarks and are thus subject to law.

Windows® is a registered trademark of Microsoft Corporation in the United Sates and other countries.



| | |
|---|---|
| Title of documentation | DLoG Software Keyboard |
| Documentation completed on | August 05, 2022 |
| Version | V 3.11 |
| Article number | Not available! |



Advantech Co., Ltd.
No. 1, Alley 20, Lane 26, Rueiguang Road,
Neihu District, Taipei 11491, Taiwan, R.O.C.

# Table of contents

# List of figures

# 1. About this manual

This manual has been designed to make using your Software Keyboard as simple as possible and provide qualified assistance if problems occur. It contains important information on using the software safely and properly.

| | |
|---|---|
| **NOTICE: Property damage** | Please read this manual thoroughly before configuring your Software keyboard. |

Advantech Co., Ltd. will not assume responsibility for any damage caused by the improper use of the software and/or in disregard of the instructions in this manual.

Within this manual, Advantech Co., Ltd. strives to provide all the information required for using your software product. However, because this is a versatile product that can be used in many different scenarios, we cannot guarantee that the information contained in this manual will cover every single aspect.

Should you require further information or if you have questions or issues needing clarification, please contact your nearest Advantech representative.

## 1.1. For qualified personnel

This manual was written for qualified personnel. The information is intended exclusively to complement existing professional expertise, not to replace it.

## 1.2. Keep this manual

Retain this manual carefully. It should be readily accessible when the DLoG Software keyboard is configured.

## 1.3. Current information on the internet

Current manuals and additional useful information can be found on the internet at

www.advantech.com

## 1.4. Design method

### 1.4.1. Property damage

These tips warn you of possible property damage:

| | |
|---|---|
| **NOTICE: Property damage** | This symbol warns you of any dangers or hazards that could potentially cause damage to the terminal or system (such as malfunctions, data loss, equipment damage, etc.). |

### 1.4.2. Hints

| | |
|---|---|
|  | This symbol indicates hints that help you to understand how to use the product or the manual. |

# 2. Functional description

The DLoG Software Keyboard brings the complete standard keyboard with function keys and numeric pad directly to your Advantech-DLoG Industrial PC screen – with easy touch operation. The user can work with an alphanumeric keypad without needing to connect a keyboard.

Overview of the most important functions:

- All of the keys found on a standard keyboard are available (alphanumeric, function keys etc.).
- Several keyboards can be defined.
- Switching between various keyboards is possible.
- The keyboard can be quickly shown and hidden again.

# 3. Installation

## 3.1. Operating systems

The following operating systems are supported:

- Microsoft Windows XP
- Microsoft Windows XP Embedded
- Microsoft Windows 7
- Microsoft Windows Embedded Standard 7
- Windows CE

## 3.2. Pre-installed on the Industrial PC

If the software keyboard was ordered together with a new Advantech-DLoG Industrial PC, then the program will have been fully pre-licensed by the Advantech-DLoG production department.

The associated files are located in directory **c:\dlog**.

## 3.3. Subsequent installation

For subsequent installation of the software on an Advantech-DLoG Industrial PC, an installation program is available. The software is automatically installed in directory **c:\dlog**, but this can be changed manually.

### 3.3.1. Licence key

A few minutes after the launch of the software keyboard, the licence key is checked. This licence key matches the serial number of the Advantech-DLoG hardware and may be found in file **DKEY_<SNR>_SWKB.TXT**. This file must be located in directory **C:\DLoG\Keys** or in the system search path. Copy the file to this directory.

The software keyboard can also be started without this file; however, it then runs for just a few minutes until the licence key is checked.

## 3.4. Subsequent installation under Windows CE

In subsequent purchase and installation of the **Windows CE-based** software keyboard you will receive a separate installation instruction with the software and the associated license file.

## 3.5. Standard delivery version

### 3.5.1. Important Files

| | |
|---|---|
| DLOGKEYBOARD.EXE | Main program, which can be automatically started using the Autostart folder or a registry run key. Alternatively, the program can be launched using a desktop shortcut or keyboard shortcut.<br>More on this in *Chapter 4: Starting the program* |
| KEYBOARD.CFG | In the configuration file (Default name: Keyboard.cfg), the layout and functionality of the software keyboard are defined. The program searches for this file in the same directory as the EXE file. |
| DKEY_<SNR>_SWKB.TXT | A licence key that matches the serial number <SNR> of the Advantech-DLoG hardware. This file must be located in directory “C:\DLoG\Keys” or in the system search path.<br>The software keyboard can also be started without this file; however, it then runs for just a few minutes until the licence key is checked. |

### 3.5.2. Languages

The default CFG file is the German layout file **Keyboard.cfg**. This file may also be found in directory **c:\dlog\gr**.

CFG files are currently available in the following additional languages:

- English
- French
- Finnish/Swedish
- Spanish
- Italian
- Russian

To activate a different CFG file from the German one:

1. Rename the file **keyboard.cfg** in directory **c:\dlog**, for example, to **keyboard_german.cfg**.
2. Copy the required CFG file, for example, the file **keyboard-fr.cfg** to directory **c:\dlog**.
3. Rename the file **keyboard-fr.cfg** to **keyboard.cfg**.
4. Restart the software keyboard.
   The new layout is displayed.

# 4. Starting the program

The DLoG Software Keyboard can be started in the following ways:

- Using the Windows **Start** menu
- By creating a registry key
- With a desktop shortcut
- Using the Autostart folder
- Using special keys on the Industrial PC

## 4.1. Call parameters

The following call parameters can be set; here, settings are case insensitive:

SHOWCTRL — Displays the basic window of the software keyboard (which is generally not visible), allowing you to exit easily by pressing **Terminate**:

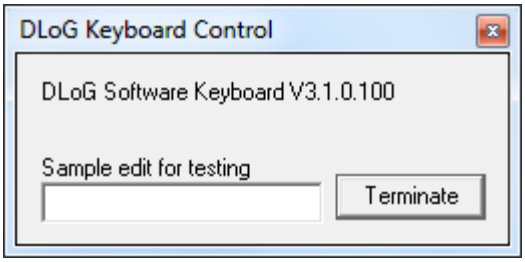



Figure 4.1: Terminate button

In addition, the version number of the program is displayed. This function should only be used for test purposes when configuring.

CFG=Filename — You use this parameter if the configuration file (Keyboard.cfg) for the software keyboard is not located in the same directory as the EXE file, or if you want a different name to be used. File names and directories that contain spaces should be placed in inverted commas.

| | |
|---|---|
| SWITCHVIEW | Allows an active software keyboard to be enabled/disabled on the screen using a desktop shortcut, which can be invoked with a keyboard shortcut. |
| EXIT | Ends an active software keyboard program. Thus, a shortcut for exiting the program can be created on the desktop, for example. |

**Call example:**

DLoGKeyboard.exe SHOWCTRL CFG=C:\MyKeyboard.txt

## 4.2. Creating a registry key

If a registry key is created for the software keyboard, then the program will launch automatically when your computer is started. Here, various call parameters can be set.

Procedure:

1. On the Windows **Start** menu, select the **Run** command.
2. Type `regedit` and confirm with **OK**.
3. Switch to the following key:
   HK_LOCAL_MACHINE / SOFTWARE / MICROSOFT / WINDOWS / CURRENTVERSION / RUN
4. Right-click with your mouse, and then select **New > String Value**.
5. Create the new **RegSZ** text **DLoGSWKeyboard**.
6. Double-click this text to open the **Edit String** dialog box.
7. Enter the EXE file and the required call parameters.

Result:

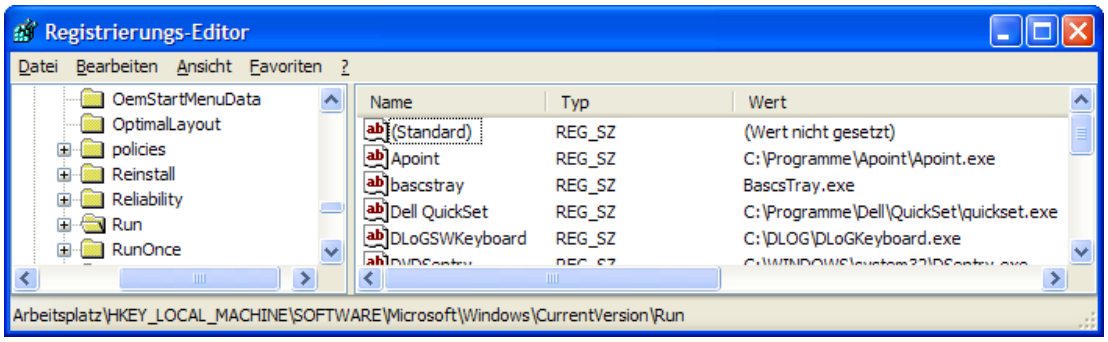



Figure 4.2: Create registry key

8. To check the new program call, restart your computer.

## 4.3. Desktop shortcut

The software keyboard can be started using a desktop shortcut. Here, various call parameters (see section of the same name) can be set.

> In order to start the software keyboard with a shortcut, at least one software keyboard must be activated with the **ShowOnStart=1** switch in the **Keyboard.cfg** file.

To create the shortcut:

1. Go to the **DLoGKeyboard** program using **Start | Programs | DLoG**.
2. Right-click **DLoGKeyboard**. A shortcut menu opens.
3. Select the **Properties** command.
4. Switch to the **Shortcut** tab.
5. For **Target**, enter the call parameter `switchview` after the EXE file (and after one space).
6. **Apply** the settings.

### 4.3.1. Set path name

In the **DLoG Keyboard Properties** dialog box, the path and name of the configuration file can also be set for the **Target** parameter. The default name **Keyboard.cfg** in the default directory **c:\dlog** then no longer has to be used.

### 4.3.2. Define keyboard shortcut

In the **DLoG Keyboard Properties** dialog box, you can specify a key combination for the **Keyboard shortcut** parameter. The software keyboard can be started and ended again with this keyboard shortcut.

## 4.4. Autostart folder

If you want the software keyboard to launch immediately at every computer start, put it in the Autostart folder:

1. Right-click the **Start** menu.
2. Open Explorer.
3. Drag the **DLoGKeyboard.exe** program to the Autostart folder using your right mouse button, and then select **Create Shortcut**.
4. If necessary, set the required call parameters in the **Properties** menu.

## 4.5. Special keys on the Industrial PC

If your Advantech-DLoG Industrial PC is equipped with special keys, you can start the software keyboard using one of them.

To prepare this:

1. Go to the **DLoG Keyboard** program using **Start | Programs | DLoG**.
2. Right-click the program, and then select the **Properties** command on the shortcut menu.
3. Switch to the **Keyboard shortcut** parameter and press the required special key on the Industrial PC.
4. **Apply** the setting.
5. If necessary, set the required call parameters in the **Properties** menu.

## 4.6. Exiting the program

The software keyboard can be exited in the following ways:

- With a keyboard shortcut that contains the **exit** call parameter in the **Properties** definition.
- With the keyboard shortcut that was defined in the **Properties** menu.

### 4.6.1. Minimize program to icon size

The software keyboard can be hidden and displayed as an icon with this key:

## 4.7. Software keyboard and taskbar

The software keyboard (as of V1.2) can be automatically adjusted to the size and position of the taskbar in order not to conceal it. This depends however on the taskbar options set.

To set the taskbar options:

1. Right-click the taskbar.
2. Select the **Properties** command on the shortcut menu.
3. On the **Taskbar** tab, you can hide the taskbar using the **Auto-hide** feature; you can choose whether you want the taskbar to be covered by other windows using the **Keep the taskbar on top of other windows** check box.

**Recommendation for taskbar configuration:**

If you want the taskbar to be **always** visible:

Auto-hide the taskbar = OFF
Keep the taskbar on top of other windows = ON

If you do **not** want the taskbar to be always visible:

Auto-hide the taskbar = ON
Keep the taskbar on top of other windows = ON

Please note the following special features:

Auto-hide the taskbar = OFF
Keep the taskbar on top of other windows = OFF

The software keyboard positions itself over the visible taskbar. If, however, an application is maximized or moved over the taskbar, the keyboard appears to be hanging in the air, since the taskbar does not automatically change its position. To prevent this, one of the two taskbar options must be enabled.

Auto-hide the taskbar = ON

If automatic hiding of the taskbar is enabled, the software keyboard always positions itself along the extreme edge of the screen. If the taskbar is then automatically shown on the screen, the position of the software keyboard remains unchanged.

With the option **Keep the taskbar on top of other windows = OFF,** the software keyboard covers the shown taskbar; the covered parts cannot be selected.

With the option **Keep the taskbar on top of other windows = ON**, the shown taskbar covers the software keyboard. With the next keypress on the area of the software keyboard that is still visible, the software keyboard returns to the foreground again.

# 5. Operation

The software keyboard is controlled simply using a touch pen or your finger tips on the touch screen. Any entries made, for example, letters and numbers, are passed to the application program currently active.

## 5.1. Keyboards included in the standard delivery version

The standard delivery version offers the user the following software keyboards:

**Standard blue color keyboard:**


Figure 5.1: Software keyboard with standard keyboard layout

**Black/White color terminal emulation keyboard:**


Figure 5.2: Software keyboard with terminal emulation keyboard layout

**Blue color 24-key keypad:**

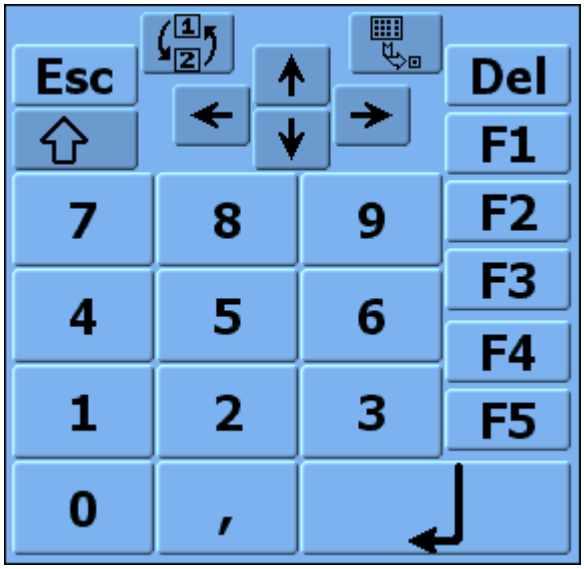



Figure 5.3: Software keyboard with 24-key layout

### 5.1.1. Keys with special functions

Alongside the standard keys, such as letters, numbers and function keys, the following keys are available, which have been designed by Advantech-DLoG:

Hide software keyboard and minimize to icon size
(function **VK_KB_HIDE**, bitmap **BMP_Minimize**)

Move position of the software keyboard up/down on the screen
(function **VK_KB_UPDOWN**, bitmap **BMP_MoveUpDown**)

Switch between the keyboards available
(function **VK_KB_SWITCHNEXT**, bitmap **BMP_SwitchNext**)

# 6. Configuring the software keyboard

The design of the software keyboard is defined in the configuration file **Keyboard.cfg**. The keys are defined and many other settings are made in the various sections and keys of this file.

Whilst the CFG file largely conforms to the INI standard, separate parsing is carried out specifically for the definitions of the individual keys.

## 6.1. Structure of the Keyboard.cfg file

In the **Keyboard.cfg** file, several software keyboards can be defined, which can be shown on the screen as required.

Every software keyboard requires:

1. A **[Keyboard_xxx]** section with general settings for transparency, size and position of the software keyboard.
2. A **[Keys]** section that follows, in which the individual keys of this software keyboard are defined.

## 6.2. Rules for editing the Keyboard.cfg file

- To edit the CFG file, editors such as Windows **Notepad** can be used.
- Redundant spaces should be avoided as they can lead to errors.
- Incorrect lines are ignored.
- As is customary, a semicolon in the first column of a line can be used for comment lines.

## 6.3. General options

Name of the section: **[Keyboard_xxx]**

In the **[Keyboard_xxx]** section, you make general settings for the respective software keyboard. This section must be explicitly available for every software keyboard.

**xxx** stands for the name of the software keyboard. This name should indicate the function of the respective keyboard.

The entry specified after the keys displays the default value or an example value.

### 6.3.1. Keys of the [Keyboard_xxx] section

**StdKeySize**=45.40

Defines the standard key size X,Y in pixels. This size is used as the default size for all keys of this software keyboard.

**Transparency**=150

Setting the transparency level.
255 = non-transparent, 0 = invisible.
If you do **not** want a software keyboard to be displayed transparently, it is recommended that for performance reasons you comment out the entry with a semicolon: **;Transparency=150**
In this case, the software keyboard window is generated with no transparent capability.

This function cannot be used in combination with Microsoft Windows CE.

**Font**=Tahoma,24,B

Here, you can set font name, font size and boldface type for the key labelling. Boldface type can be disabled by omitting the **B**.
In general, the fonts are generated with activated anti-aliasing. With small font sizes, however, anti-aliasing is ignored by the system.

**ShowOnStart**=0

Normally the software keyboards should not be visible at program start, so the default value here is **0**. With the value **1**, the software keyboard is displayed immediately when the program launches.

**BorderSize**=3.3

Defines the width of the border between the keys and the Windows border in the X,Y direction.

**Pattern**=L

Four standard and one optional keyboard layouts are pre-defined:
Pattern**=W** (white), **=B** (blue), **=L** (blue with a narrow border),

**=T** (white with a narrow border), **=X** (transparent background without border)

Pattern=W:

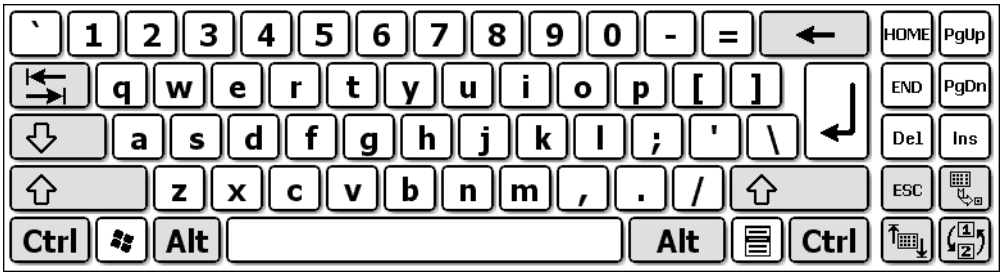



Figure 6.1: Key layout with Pattern=W

Pattern=B:


Figure 6.2: Key layout with Pattern=B

Pattern=L:


Figure 6.3: Key layout with Pattern=L

Pattern=T:

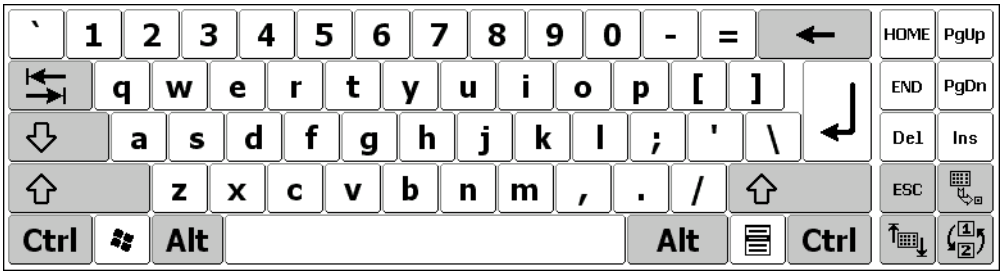



Figure 6.4: Key layout with Pattern=T

**Optional:** (Windows 7 \ Windows Embedded Standard 7)

Pattern=X:


Figure 6.5: Key layout with Pattern=X

Setting requires the following additional entries in the “keyboard.cfg” file:

```
Pattern=X
InvisibleCol=21,107,193
NoBorder=1
```

**Position**=24

Here, you can define the position of the software keyboard. The following position specifications are possible as numeric values:

24 Bottom edge, centre
17 Top edge, centre
18 Left edge, centre
20 Right edge, centre
3 Top left
5 Top right
10 Bottom left
12 Bottom right

The value **32 (=Fixed Position)** can be added to these values (specify total). In this case, the position of the keyboard is not changed when a switch is made to the next keyboard (see **VK_KB_SWITCHNEXT**). Otherwise, the newly activated keyboard by default is displayed with the same position code as the previous one.

**ShowTitle**=0

**1** = Show title bar of the keyboard window
**0** = Do not show title bar

In the title bar, the name of the software keyboard, which was specified for **Keyboard_xxx** in the section designation, is also displayed.
It is recommended, however, not to show the title bar. With the title bar shown, the software keyboard can gain focus if you click the title bar – the current input window would lose focus.

**ShowTaskWin**=0

**1** = Show in the taskbar
**0** = Do not show

Here, you can make the individual software keyboard windows visible in the taskbar and in task switching with ALT-TAB.

**Background**=R,G,B

Definition of the window background with RGB values; this entry has priority over the **Pattern** specification.

**OverTaskbar**=0

**1** = fade taskbar
**0** = Do not fade taskbar

Define if taskbar should be faded in case of displayed keyboard.

## 6.4. Defining the keys

Name of the section: **[Keys]**

In the **[Keys]** section, you define the individual keys of the software keyboard. The overall size of the keyboard results from the arrangement of the keys.

An entry for a key definition contains the following, mostly optional fields:

| Labelling | Key code | Pattern | Size | Position |
|---|---|---|---|---|
| Normal/Shift/AltGr\| | VirtualKeycode\| | PatternBitmap, | X-Len,Y-Len, | X-Pos,Y-Pos |

### 6.4.1. Labelling

The first three fields (up to the first | character) are for the labelling of the keys:

1. **Normal** field, labelling without <Shift> and without <Alt Gr>
2. **Shift** field, labelling with the <Shift> key pressed
3. **AltGr** field, labelling with the <Alt Gr> key pressed

The following combinations are permissible, for example:

**A|** In all three modes, the upper-case A is always displayed

**a/A/|** Normal = a, with <Shift> = A, with <Alt Gr> = no display

**A//|** Normal = A, with <Shift> or <Alt Gr> = no display (blank key)

If the **/** or **|** character, or other characters that are difficult to enter, are being specified for the labelling, then this can be done with the hexadecimal operator $:

**Character /** displayed through $2F

**Character |** displayed through $24

### 6.4.2. Bitmaps

Bitmaps can also be used for the key display. The pre-defined standard keyboard symbols may already be found in the resources of the software keyboard program.

Alternatively, the bitmap files can also be specified directly:

| | |
|---|---|
| **$BMP_Shift\|** | Display of the Shift arrow from the program resource |
| **$BMP_MySymbol.bmp\|** | Display of the bitmap file **MySymbol.bmp** |

Please note the following:

- The special character **$** must be specified in front of the bitmap name.
- The bitmap files must either be located in the current working directory of the program or they must be specified with full path name.
- For all bitmap files, the **Color RGB = 140,195,255** is permanently defined as the transparent color.

**Bitmap definitions for label symbols (Examples)**

| | | |
|---|---|---|
| BMP_RETURN |  | RETURN symbol |
| BMP_TAB |  | TAB symbols |
| BMP_CAPS |  | CAPS LOCK, continuous Shift symbol |
| BMP_SHIFT |  | SHIFT symbol |
| BMP_BACK |  | BACKSPACE symbol |

| | | |
|---|---|---|
| BMP_APPKEY |  | Application key |
| BMP_INSERT |  | Small “INSERT” text |
| BMP_END |  | Small “END” text |
| BMP_DELETE |  | Small “DELETE” text |
| BMP_NEXT |  | Small “PAGE DOWN” text |
| BMP_PRIOR |  | Small “PAGE UP” text |
| BMP_HOME |  | Small “HOME” text |
| BMP_MoveUpDown |  | Special symbol for SWITCH POSITION (see **VK_KB_UPDOWN**) |
| BMP_SwitchNext |  | Special symbol for SWITCH KEYBOARD (see **VK_KB_SWITCHNEXT**) |
| BMP_Keyboard |  | Special symbol for the MAXIMIZE function |
| BMP_Minimize |  | Special symbol for the MINIMIZE function |

### 6.4.3. Virtual-key codes

Apart from certain special codes, the Windows virtual-key code (alternatively the scan code with **#**, see below) is specified as the text name here.

At present, only one basic key code can be set for each key; in combination with <Shift>+<Alt Gr>, this then automatically gives the other assignments. Therefore, for the key code **E**, the euro symbol **€** is automatically produced with the <Alt Gr> key pressed.

The virtual-key code names can be obtained from Microsoft MSDN (Microsoft Developer Network).

Examples:

e/E/€|E

Assignment of label and code = **E** to a key
For the assignment of the standard ASCII characters, no special virtual-key code is necessary; the character can be specified directly as a code.

$BMP_Return|VK_RETURN

Assignment of bitmap and virtual-key code name to the <Return> key

,/;/|VK_COMMA

Assignment to a standard comma key

As an alternative to the virtual-key code, the numerical scan code of a key can also be entered directly. This is made possible through the **#** character.

If, for example, you want to specify the scan code of the <Esc> key, you must specify the value **#1** in the **VirtualKeycode** field; for the A key, you must specify the value **#30**.

This is necessary in particular for keys with national special characters, since no virtual-key code definitions generally exist for these keys.

### 6.4.4. Key codes for special functions

**VK_KB_UPDOWN**

This key code is not used as a keyboard value, but instructs the program internally to change the keyboard position (from bottom edge to top edge, and vice versa).

**VK_KB_SWITCHNEXT**

Keys with this key code cause a switch to the next defined software keyboard. See also the **ExcludeChain** parameter.

**VK_KB_HIDE**

If a key with this key code is pressed, then the associated keyboard window is removed from the screen. In addition, the **Maximize** keyboard (**Keyboard_Maximize**) is activated if available.

### 6.4.5. Key codes for special characters

| Virtual-key codes | Hardware key codes |
|---|---|
| VK_NUMRET | <Return> on numeric keypad |
| VK_CIRCUMFLEX | 220 |
| VK_SHARP_S | 219 |
| VK_ACCENT | 221 |
| VK_PLUS | 187 |
| VK_GER_UE | 186 |
| VK_GER_OE | 192 |
| VK_GER_AE | 222 |
| VK_NUMSIGN | 191 |
| VK_COMMA | 188 |
| VK_POINT | 190 |
| VK_SMALLER | 226 |
| VK_MINUS | 189 |
| VK_ALTGR | VK_RMENU |

### 6.4.6. PatternBitmap

Here, you can set a different bitmap pattern for the key; at present, only the entry **$ExtBmp1** is possible for keys highlighted in color, for example, the <Return> or TAB key. If this entry is not used, then the entry **including comma** must be omitted.

### 6.4.7. X-Len,Y-Len,X-Pos,Y-Pos

Here, you define the size and positioning of the key. For parameters left blank (,,,) or parameters that are not available, the default value is always set.

The default value for the size is defined through the **StdKeySize** entry in the **Keyboard_xxx** section. The default value for the positioning is always the next X-position after the previously defined key in accordance with its size.

A minus sign in front of the value is a reference to the standard key size:

- An X-Len of **–2** produces a key that is twice as wide. A Y-Pos of **–5** positions the key in the fifth key row.
- A plus sign in front of the value adds the value relative to the current standard position; with **X-Pos = +5**, the key is positioned at a distance of 5 pixels from the previous one.
- Values with no **-/+** sign are used as absolute pixel positions within the keyboard window.

## 6.5. Keyboard_Maximize section

**[Keyboard_Maximize]** is reserved as a special section for the keyboard definition. This software keyboard contains only one key, to which the **VK_KB_SWITCHVIEW** function should be assigned.

All the parameters listed in Section *6.3 General options* can be used for the definition of this software keyboard.

Example of a Maximize Keyboard definition:

```
[Keyboard_Maximize]
StdKeySize=40,40
Transparency=150
Font=Tahoma,22,B
Pattern=W
ExcludeChain=1
BorderSize=0,0
Position=12
;MaxiMode=0
[Keys]
$BMP_Keyboard|VK_KB_SWITCHVIEW
```

### 6.5.1. Options MaxiMode and ExcludeChain

The options **MaxiMode** and **ExcludeChain** were introduced specially for this software keyboard. They allow specific configurations to be made.

**MaxiMode**=1

This is a special parameter valid only for **Keyboard_Maximize**.

**1** = AutoHide (default). If any software keyboard is visible on the screen, then **Keyboard_Maximize** is automatically made invisible. If there is no longer any (normal) software keyboard visible on the screen, then **Keyboard_Maximize** reappears automatically.

**0** = Always visible. **Keyboard_Maximize** is always visible on the screen, irrespective of the other software keyboards.

**ExcludeChain**=0

If you set this parameter to **1**, the associated software keyboard is excluded from general commands; the software keyboard remains unaffected by this. This applies, for example, to the successive switching through of the keyboards with **VK_KB_SWITCHNEXT**.

# 7. Settings for Windows CE

The following is valid as the DLoG software keyboard:

- The integration of external bitmaps shall **only** be supported for 256-color non-compressed bitmaps (BMP) 256 (Windows CE).
- The call parameters (see Chapter 4) can be given in the **Keyboard.arg** text file. The file (optional) must be placed in the same directory as the .exe.
  The following call parameters shall be supported and must be written in capitals, be on one line and be separated by spaces or commas: SHOWCTRL, CFG, EXIT, SWITCHVIEW.
  Example of the file contents: **SHOWCTRL CFG=MyTestKeys.cfg**“
- Transparency functionality is not supported.

## 7.1. Keyboard.cfg

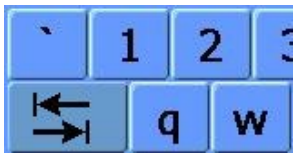
### 7.1.1. Key patterns

T=Thin, white with a narrow border for CE

Pattern **T** (Thin White)


L=Blue with a narrow border CE

Pattern **L** (Thin Blue)




The Options (W = white, B = blue) can be set optionally.

### 7.1.2. AutoHideSIP Parameter=0

If these parameters are set to 1 for a keyboard the WinCE-Input Panel shall be closed automatically when this keyboard is displayed on the screen. Any input panels that are re-opened shall be automatically closed again.

This parameter shall only be supported by Windows CE.

### 7.1.3. Displaying the Shift-Alt-Ctrl Status

An explicit status for specific keys may also be provided for each virtual keycode. Please see the table below for details:

| Key Description | Status Code indicating that the key has been "pressed" | Status Code indicating that the key has been released |
|---|---|---|
| Shift (and CapsLock) | S | s |
| Control | C | c |
| Alt (left Alt Key) | A | a |
| AltGr (right Alt key) | G | g |

The status code must be placed before the keycode and introduced with a $ and followed by a"

e.g.

E/E/€

Here if a lowercase e is pressed, the AltGR code automatically causes the Euro sign to be displayed and a lowercase e to be displayed if shift+lowercase e is pressed. To be 100% sure, the user can also combine the options together e.g.

The combinations and status codes for the keys shall currently only be taken into consideration for **one** software keyboard. Therefore, the pressing of status keys on a given hardware keyboard or an additional SW keyboard on the display will lead to internal status handling errors and thus issue an incorrect keycode.

## 7.2. Bitmap definitions of Label Symbols

| | |
|---|---|
| BMP_RETURN | Return Character |
| BMP_TAB | Tab Character |
| BMP_CAPS | CapsLock, maintains the shift sign⇩ |
| BMP_SHIFT | Shift Sign⇧ |
| BMP_BACK | Backspace Character← |
| BMP_BACKSMALL | Backspace Characters (small for WinCE) |
| BMP_APPKEY | Windows Application Key |
| BMP_INSERT | Small text INS |
| BMP_END | Small text END |
| BMP_DELETE | Small text DEL |
| BMP_NEXT | Small ↓text icon |
| BMP_PRIOR | Small↑ text icon |
| BMP_HOME | Small Pos1 Text |
| BMP_MoveUpDown | Special Character<br>Switch Position ➔ VK_KB_UPDOWN |
| BMP_MoveUpDownSm | reduced for WinCE |
| BMP_SwitchNext | Special Character.<br>Switch Keyboard ➔ VK_KB_SWITCHNEXT |

| | |
|---|---|
| BMP_SwitchNextSm | reduced for WinCE |
| BMP_Keyboard | Special Character. for the maximize function, for example |
| BMP_KeyboardSm | reduced for WinCE |
| BMP_Minimize | Special Character. for the minimize function |
| BMP_MinimizeSm | reduced for WinCE |

### 7.2.1. Bitmaps as of V1.4:

| | |
|---|---|
| BMP_Ins | Small Text INS in English (insert) |
| BMP_Del | Small Text DEL in English (Delete) |
| BMP_Up | Upward-pointing cursor arrow |
| BMP_Down | Downward-pointing cursor arrow |
| BMP_Left | Left cursor arrow |
| BMP_Right | Right cursor arrow |
| BMP_PageUp | Symbol for the ↑ icon, 2 upward-pointing arrows |
| BMP_PageDown | Symbol for the ↓ icon, 2 downward-pointing arrows |
| BMP_ArrowPos1 | Symbol for Pos1. Left or Right arrow. |
| BMP_ArrowEnd | Symbol for end, left or right arrow. |

# Index