

# AllChars for Windows v2.00 for 32-bit Windows



## FREEMWARE



Copyright © 1994-1997:  
Jeroen Laarhoven  
Geleen 12  
8032 GB Zwolle  
Netherlands

[www.whiteants.com](http://www.whiteants.com)  
[jeroen@whiteants.com](mailto:jeroen@whiteants.com)

# Contents

Introduction.....	
What AllChars can not do.....	
Version Two.....	
How does AllChars work ?.....	
Installing AllChars on your system.....	
Removing AllChars from your system.....	
AllChars and deadkeys.....	
Macro's.....	
Delays in macro's.....	
Compatibility.....	
Defining macro's.....	
AllChars in the Taskbar.....	
AllChars popup menu.....	
About AllChars.....	
Help.....	
Macro's.....	
Enable / disable AllChars.....	
Decimal Symbol translation.....	
Configuration.....	
Exit AllChars.....	
Special key-combinations.....	
The writer.....	
Copyright, usage, copying and responsibility.....	
Appendix A - Customising the translation-tables.....	
Upgrading from version v1.2x.....	
Appendix B - Technical data.....	
Appendix C - ANSI-table and AllChars key-combinations.....	

# Introduction

*AllChars for Windows* is an utility which adds a missing functionality to Windows: an easy way to make accents like é Û ç î æ and special characters like f<sup>2</sup> %• © £ ± ß ° ½ ¿ \_ \_ » ™ „, available within every Windows program.

*AllChars for Windows* is very easy and intuitive in usage and it is possible to adapt it to your wishes.

*AllChars for Windows* is FREeware.

*AllChars for Windows* version 2 works with Windows95 and Windows NT 4.0 (upwards).

*AllChars for Windows* version 2 also knows *macro*'s. With this you can make *AllChars* enter any string of characters in your application by just hitting three keys.

As a bonus, *AllChars for Windows* makes it possible to change the point in the numeric keyboard to the decimal symbol as defined in the International settings of Windows. In most European countries this is a comma instead of a point.

## What AllChars can not do

*AllChars for Windows* can only make characters that are available in the *current* font.

So *AllChars* can not (temporary) change to another font like 'Symbol' or 'Wingdings' for special characters of these fonts. It can also not make characters that are not in the font, like special Russian or Turkey's characters.

However if you use special fonts for these languages, you can redefine *AllChars* to make all characters in these fonts available.

See Appendix C for the characters available in the normal 'Latin 1' character-set and Appendix A for customising *AllChars*, e.g. for other character-sets.

## Version Two

*AllChars for Windows* versions 2 is a 32-bit application for Windows95 and Windows NT 4.0 (upwards).

*AllChars for Windows* versions 1.xx are 16-bit applications for Windows 3.11 which work fine under Windows95.

Because *AllChars for Windows* version 2 uses (a lot) more memory than version 1.xx, I suggest that Windows95 users who are short on memory use version 1.xx.

# How does AllChars work ?

When *AllChars for Windows* is started, the program 'hooks' itself into the Windows system and watches all keys pressed.

By pressing the Control-key **and** releasing it again without having pressed another key in the meantime, *AllChars* is activated. It will now try to translate the next two keys pressed into one special character or macro. For example:

<b>Ctrl, ', e</b>	gives an	<b>é</b>
<b>Ctrl, 1, 2</b>	gives an	<b>½</b>
<b>Ctrl, -, f</b>	gives an	<b>f</b>
<b>Ctrl, J, L</b>	gives	<b>Copyright © 1994-1997 by Jeroen Laarhoven</b>

Because the normal usage of the Control-key is always using it together with another key (pressed at the same time) and *AllChars* is activated by pressing and then releasing the Control-key, you can use *AllChars* with all Window programs (including system dialogs etc.) without conflicting with these programs.

Most special characters are built out of keys representing 'parts' of this character, like the examples above.

There are also 'double-keys' defined for fast usage: **f** and **f** gives an **f**, and **L** and **L** gives a **£**.

Some characters can also be formed by there name: **d** and **g** gives a **°** (degree), **c** and **r** gives a **©** (copyright), **m** and **u** gives a **µ** (mu), etc..

Macro's work the same way and can be defined by the user. See the chapter about Macro's for more information.

It doesn't matter which of the two keys you type first. And you can use upper- and lowercase mixed. So: **c** and **r**, or **r** and **c**, or **C** and **r**, or **R** and **C**, etc. all give **©**.

A help screen showing all standard defined key-combinations is available by typing **Ctrl h h**. **Ctrl m m** shows the available user defined macro's.

In Appendix C you can find all the special characters with their standard defined key-combinations and their ANSI-values.

## Installing AllChars on your system

You can install *AllChars for Windows* on your system by copying all *AllChars* files to a directory (any) on your harddisk.

To activate *AllChars*, just start it.

If you wish *AllChars for Windows* to start every time you start Windows, add the program to your startup group of Windows.

*AllChars for Windows* does *not* need to be in your Windows directory. *AllChars* also makes *no* changes to your system files or registration database.

## Removing AllChars from your system

You can remove AllChars from your system by deleting all files in the *AllChars* directory. If you have added *AllChars* to your startup group, you also have to remove it from there.

## AllChars and deadkeys

Windows has for international users (like us in Europe) the so called deadkeys. After typing some keys their characters are not shown. Windows waits for you to type the next key and then tries to see if the combination of these two makes any sense. If so the accent is typed. This way we Europeans got our needed accents.

For example you type ' and you see nothing. If the next key is an e then you see an é , if the next key is a t then you see 't .

This looks like the functionality of *AllChars for Windows*. So why use *AllChars* ?

- *AllChars* is language independent and can make *all* of the font specified characters. All accents: the normal ones, and also some special German, Scandinavian, French and Spanish ones and all other characters not found on the keyboard, like <sup>2</sup> %• © ± ° ½ \_ » etc.
- If you're a programmer or sometimes make spreadsheet macros, the use of these deadkeys is very annoying, because in programming languages accents like ' and " are part of the program/macro language.
- Most programs do not handle deadkeys the way they should. You don't get any feedback and so often just type the key again ...

If you wish, you can use *AllChars* together with deadkeys (but why should you?).

If you wish to switch off the deadkeys, the way to do this is by changing the settings of the keyboard to the standard US layout.

Open the *Control Panel* again, choose *Keyboard* and change (or add if needed) the keyboard layout *English (United States)*.

## Macro's

*AllChars for Windows* can next to translating key-combinations into special characters also translate key-combinations into so called macro's: user defined strings to text, e.g.:

**Ctrl, J, L** gives **Copyright © 1994-1997 by Jeroen Laarhoven**

You can define these macro's yourself and they can be used for:

- Inserting often typed strings like your name, company-name, address, etc.
- Inserting code bits for us programmers, like 'if then begin end else begin end' etc.
- Inserting login and password information. Watch out ! others can use these to ! Use an encrypted macro for this: see below.
- Anything .. you don't want to type again and again (like 'AllChars for Windows' when making this manual).

You can define a maximum of 75 macro's and a macro can hold a maximum of 127 characters. A macro can hold any printable character including special characters made with AllChars for Windows.

*AllChars* special characters (like é Û ç î æ f ² % © £ ± ß ° ½ ¿ \_ \_ » ™ „) are *always* translated before translating macro's. This means that you can *not* redefine a code in use for a special character.

Next to characters, macro's can also hold special codes. These start with a backslash '\` character, followed by letter(s) or numbers.

These special codes are:

- '\b' - inserts a BACKSPACE (ASCII 8) character.
- '\cNNN' - inserts a character with the decimal ASCII value NNN. E.g. '\c072ello' will give 'Hello' because the ASCII value of the 'H' character is 72. Do not use ASCII value 001 or 002, these are used internally by *AllChars*.
- '\d' - DOWN (arrow) key.
- '\e' - END key.
- '\h' - HOME key.
- '\l' - LEFT (arrow) key.
- '\n' - insert a CR character (like pressing the Enter-key).
- '\p' - PAGE UP key.
- '\q' - PAGE DOWN key.
- '\r' - RIGHT (arrow) key.
- '\t' - inserts a TAB character.
- '\u' - UP (arrow) key.
- '\y' - tells *AllChars* this macro is encrypted. Do not insert this code yourself. Use the Encrypt key on the Macro's form instead.
- '\NN' - delay time: wait the specified time before continuing with the macro. See below.
- '\` - insert the '\` character.

*Watch out !* Some applications do not 'like' these special codes. See the Compatibility chapter below.

## Delays in macro's

Sometimes you want to wait a while between sending characters or parts of the macro's. E.g. for a login macro between the name and password.

For this you can define delay times in macro's.

This is done by inserting the special code '\NN' with 'NN' holding any number between '00' and '99'. When reading this code, *AllChars* will wait the specified number in tenths of seconds. So '\15' will delay for 1½ second.

If the delay code is the first code in the macro, *AllChars* will wait the defined time before sending every character. So this will give 'slow typing'.

*Watch out !* Some applications (e.g. Microsoft Word) do not like these delays, see the 'Compatibility' chapter below.

## Compatibility

Using the special codes like ‘\n’ and delays is not always working correctly. To correct some of the problems *AllChars* knows a compatibility section in the ALLCHARS.INI file.

In this section you can define compatibility options for specified programs. For example:

```
[Compatibility]
DELPHI.EXE=1
DELPHI32.EXE=1
WINWORD.EXE=2
```

If the program is not reacting correctly on special codes like ‘\n’ then set it’s compatibility to ‘1’ (like DELPHI32.EXE in the example). The default value is ‘0’.

If the programs (like Word) do not like delays set the compatibility to ‘2’. Delays will be skipped for these programs.

You can add these values, so ‘3’ is also allowed.

To define a non default compatibility you have to know the name of the program receiving the keys. To identify the program: use the program typing in text and then type **Ctrl I I**. *AllChars* will then insert the name to identify the active program in the text followed by its current compatibility setting. E.g. typing **Ctrl I I** here (in Word) gives: ‘WINWORD.EXE=2’.

Use this name if you want to change the compatibility setting.

After changing the compatibility for a program in the ALLCHARS.INI file, you have to stop and start again this program.

If you use programs that need the compatibility option set, please email me so I can include this compatibility setting in future releases of *AllChars*.

## Defining macro’s

The file ALLCHARS.INI in the *AllChars* directory contains the macro’s.

You can edit, add and delete macro’s using the Macro’s screen (see the chapter ‘AllChars Popup menu - Macro’s’ below) or by changing this file. Use the Macro’s screen to make encrypted (secret) macro’s.

After changing the macro’s in the ALLCHARS.INI file you always have to stop *AllChars* and start it again to activate the changes.

The macro’s are below the ‘[macros]’ header and have the format ‘MM=anything’. ‘MM’ stands for the macro code and ‘anything’ stands for the text (including special codes) to type when this macro is activated. For example:

```
[Macros]
ex=Example of a macro\n
sx=\01Example of a slow macro
dx=Example of a macro on ... \n\20...two lines
yx=\YBD53D72DE36CB45A092B27161833E458EF51A9639865
```

In this example:

- Typing **Ctrl e x** will type ‘Example of a macro’ followed by a CR character.
- Typing **Ctrl s x** will type ‘Example of a slow macro’ with a tenth of a second delay between the characters.
- Typing **Ctrl d x** will type ‘Example of a macro on ...’ and a CR then wait two seconds and continue

typing '... two lines'.

- Typing **Ctrl y x** will type the encrypted (secret) text 'This is a SECREd text'.

## AllChars in the Taskbar



Normally *AllChars* works in the background, without you having to pay any attention to it. After starting *AllChars* it will display a icon in the right part of the Windows taskbar near the time. This icon shows you the state of the program and from there you can access the program.

The *AllChars* icon holds two keys numbered one and two. Normally they both are grey. But if you have activated *AllChars* by hitting and releasing the Ctrl-key key one will turn red, telling you that *AllChars* is waiting for the first key of the key-combination.

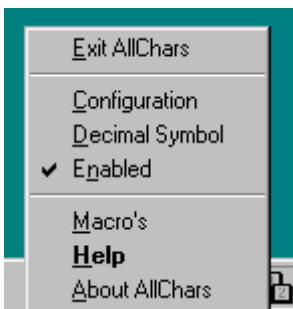
After pressing the first key the second key in the icon will turn red, informing you that *AllChars* is waiting for the second key. After pressing this second key the keys in the icon will turn grey again.

In this icon you can also see if the translation of the numeric keypad dot-key (point) into the decimal symbol is switched on or off. If switched on you see a small dot in the lower left corner of the icon.

You can access the *AllChars* program by this icon:

- Click on the icon with the right mouse button and you will see the *AllChars* popup menu.
- Double click on the icon and *AllChars* will show you a help screen with all special characters and the key-combinations to make these characters with *AllChars*.

## AllChars popup menu



Right click on the icon in the taskbar and you will see the *AllChars* popup menu.

From there you can:

- Show the 'aboutbox'.
- Show the help screen.
- Show the available user defined macro's.
- Enable or disable (temporary) *AllChars*.
- Switch on/off the translation of the dot (point) key in the numeric keypad into the decimal symbol.
- Change the configuration.
- Exit (stopping) *AllChars*.

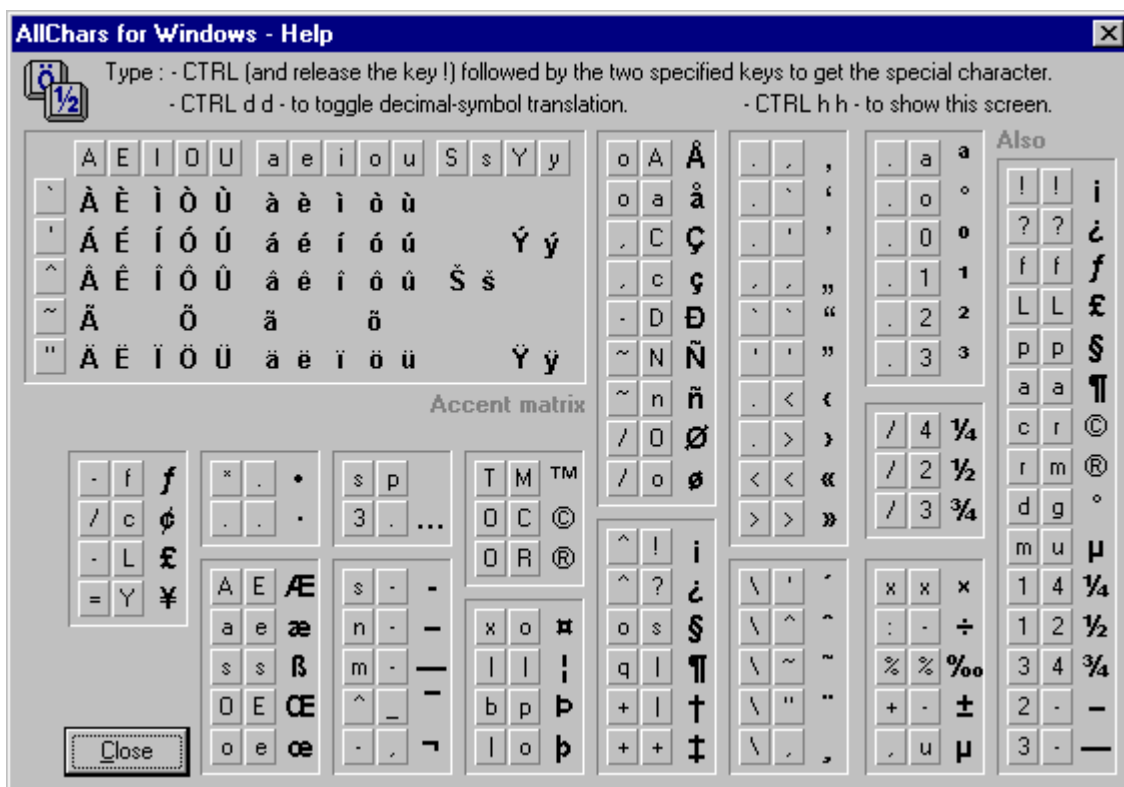


## About AllChars



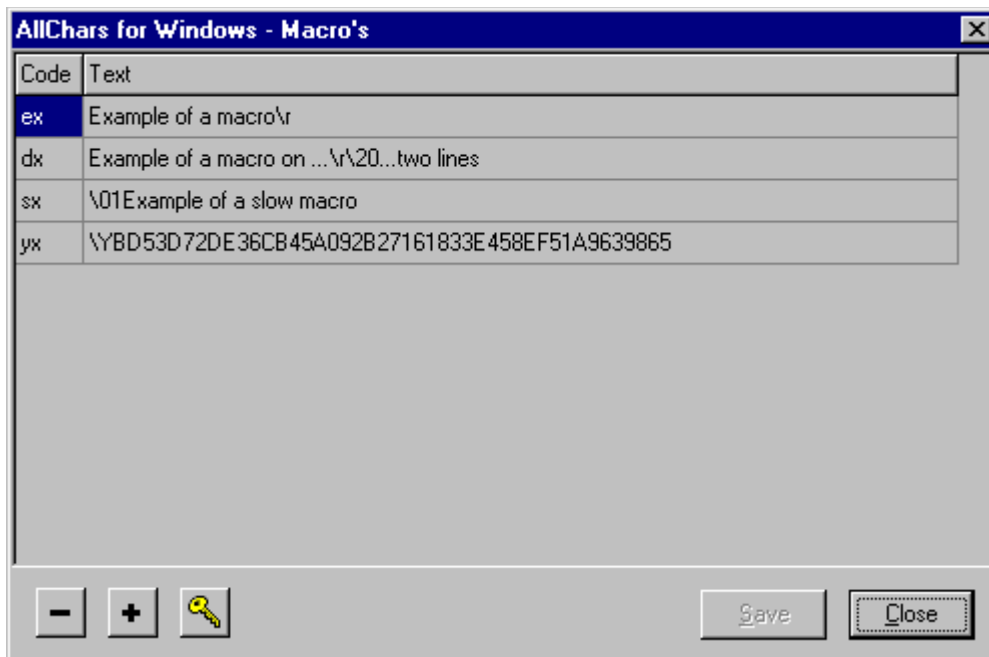
The aboutbox gives you information about the *AllChars* version and the copyright. Here you can also find the home website of *AllChars* on the internet: [www.whiteants.com](http://www.whiteants.com).

## Help



The help screen shows all special characters and the key-combinations to make these characters with *AllChars*.

# Macro's



The macro screen shows the available macro's and gives you an easy way to modify them.

By clicking on the macro text or code you can edit this text or code.

Use the '+' key for adding a macro and the '-' key for deleting the selected macro.

With the Encrypt key (showing a yellow key) you can encrypt the selected macro. This way you can *not* read the code that will be inserted by the macro. Useful for passwords etc..

First type the normal text for the macro in the text-field and then hit the Encrypt key. In the figure the 'yx' macro is the encrypted macro 'This is a SECRETED text'. So typing **Ctrl y x** will give 'This is a SECRETED text'.

See the chapter about Macro's for more information about macro's and special characters in macro's.

## Enable / disable AllChars

With this function you can (temporary) disable or enable *AllChars*. The *AllChars* icon in the Windows taskbar will crossed, showing *AllChars* is disabled.

For specialists: if *AllChars* is disabled, the keyboardhook is unhooked so *AllChars* should not interfere with any other program.

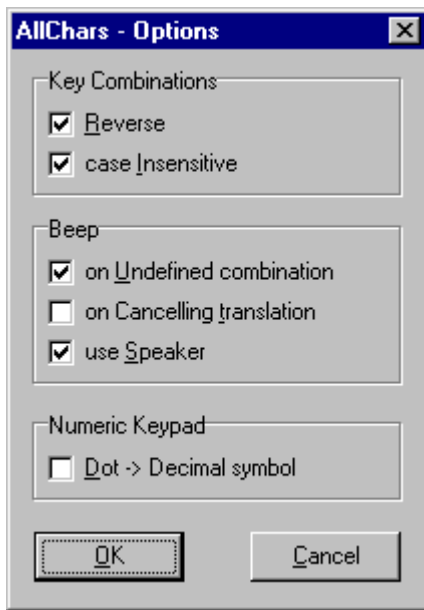
## Decimal Symbol translation

With the 'Decimal Symbol' function you can switch on/off the translation of the dot-key (point) in the numeric keypad into the decimal symbol.

If switched on the *AllChars* icon in the Windows taskbar shows a small dot in the lower left corner. If you can't see it ... you might need glasses.

# Configuration

With the Configuration screen you can change the *AllChars* settings.



You can change the next *AllChars* settings:

## Key Combinations:

- *Reverse* - allows typing the key-combination in reverse order. For example: **c r** gives ©, but also **r c** gives ©. *AllChars* first tests the typed order and if this key-combination is not defined, it will try the reverse order.
- *case insensitive* makes the the key-combination case insensitive. So next to **c r**, also **C R** and **C r** and **c R** gives the ©-sign.  
Here also first the typed key-combination is tested and only if this is not defined switching the case of the characters is tested. So ' **E** will always give **É** and ' **e** will always give **é**.

## Beep:

- By switching on the option *on Undefined combination* you will be warned by a beep if the typed key-combination is not defined. If switched off no beep will be given.
- If the option *on Premature ending* is selected you will hear a beep if the *AllChars* translation is stopped prematurely by typing a non-character key (e.g. space, Alt, Ctrl, Tab, etc.) as one of the keys in the key-combination. If switched off no beep will be given.
- When *use Speaker* is selected, the computer speaker is used for the beep in stead of the default beep.

## Numeric Keypad:

- *Dot -> Decimal symbol* will, if selected, change the dot (point) key in the numeric keypad into the decimal symbol, over here a comma.  
Use the Windows Control Panel to define the decimal symbol.  
By typing **Ctrl d d** you can toggle this translation between on and off.

# Exit AllChars

Choose 'Exit' if you want to stop *AllChars*. The program 'unhooks' itself from the system and removes itself from memory.

## Special key-combinations

*AllChars* knows key-combinations that do not result in a special character or macro, but are used to perform an action. These key-combinations are:

- Ctrl **D D** : to toggle the translation of the numeric keypad to decimal symbol (on/off)
- Ctrl **H H** : to show the help screen with all standard defined key-combinations.
- Ctrl **I I** : to insert the name of the active program: see the chapter ‘Compatibility’.
- Ctrl **M M** : to show the available user defined macro’s.

# The writer

The first version of *AllChars for Windows* has been written in 1994 by Jeroen Laarhoven and was, at first, distributed in the Netherlands only as SHAREWARE.

However the work was too much and the money too little so it became FREEWARE.

The 32-bit version 2 has been written in the winter of 1997. It's has been written in Borland Delphi.

Jeroen Laarhoven is reachable by email at:

- [jeroen@whiteants.com](mailto:jeroen@whiteants.com)

The latest version(s) of AllChars will be available at the White Ants website:

- [www.whiteants.com](http://www.whiteants.com)

White Ants systemhouse BV is specialised in developing software in Borland Delphi for all Microsoft Windows platforms.

Our main activities are:

1. Turnkey building of customer specific software.  
Varying from single applications to complex real-time distributed computer systems and multi-tier client/server solutions.
2. Tools and components for professional colleague software developers.  
Including the Delphi CASE tool White Ants ModelMaker.

Visit our website for information about us and our latest tools and components.

# Copyright, usage, copying and responsibility

*AllChars for Windows* is copyrighted by the writer Jeroen Laarhoven.

*AllChars for Windows* is FREEWARE, you are free to use the program without having to pay a single penny, cent or whatever. You may copy the program and its documentation freely.

Spreading the program and its documentation by electronic networks, diskette, CD-ROM etc. is allowed as long as you're not selling it and the program and documentation is distributed unchanged and completely (see Appendix B).

And ... the developer of *Allchars for Windows* makes no representations about the suitability of this software or about any content or information made accessible by the software, for any purpose. The software is provided 'as is' without express or implied warranties, including warranties of merchantability and fitness for a particular purpose or noninfringement. This software is provided gratuitously and, accordingly, the developer of *Allchars for Windows* shall not be liable under any theory for any damages suffered by you or any user of the software.

The developer of *Allchars for Windows* will offer limited support for this software and is under no obligation to issue updates to this software.

# Appendix A - Customising the translation-tables

The file ALLCHARS.INI in the *AllChars* directory contains the tables used by *AllChars* for translating the key-combinations into special characters.

The file can be changed with any text editor, e.g. notepad.

The tables are built as long strings of characters built up out of groups of three characters separated by a space character.

You can change or add to these tables, but do not change the format.

There are three tables: *Standard*, *Accent* and *User*.

The first one contains all definitions except accents, which you find in the second table.

The *User* is normally empty and meant for you to add definitions.

The definitions in the *User* are interpreted first so you can also override definitions defined in one of the other tables. For example if you define **cr**® in the *User*, then you will get the ®-sign instead of the ©-sign when typing **Ctrl c r**.

This way you never have to change the *Standard* and *Accent*.

After changing the tables you always have to stop *AllChars* and start it again to activate the changes.

## Upgrading from version v1.2x

The format of AllChars.Ini is changed in version 2. Do *not* overwrite the new AllChars.Ini with AllChars.Ini from version 1.2x ! AllChars will not work with the old AllChars.Ini.

Now there are now more sections in AllChars.Ini.

The '[tables]' section contains the translating tables. There names are changed from *StandardTable* to *Standard*, *AccentTable* to *Accent* and *UserTable* to *User*.

The configuration settings are now in the section '[Configuration]'.

So if you want to update and reuse your *UserTable*, copy your '*UserTable=...*' line from the old file to the new file and change it in '*User=...*'.

Do not modify the other sections.

## Appendix B - Technical data

- Operatingsystem - Windows NT 4.0 or Windows95
- Harddisk usage - ca. 1 Mb (350 Kb without Manual).
- Memory usage - ca. 350 Kb plus some Kb per running application.
- Principle of operation - AllChars uses the Windows keyboard-hook and calls, if not translating a key-combination, other keyboard-hooks if any. Characters and keys are send to the target window using WM\_CHAR or WM\_KEYDOWN + WM\_KEYUP Windows messages.
- Configuration - In ALLCHARS.INI (standard Windows profile format).
- Minimal configuration - *AllChars* needs the following files to function:  
ALLCHARS.EXE  
ALLCHR32.DLL  
ALLCHARS.INI
- Complete file list - *AllChars* v2.00 should be distributed with the files:  
ALLCHARS.EXE - the program itself  
ALLCHR.DLL - *AllChars* dynamic link library  
ALLCHARS.INI - *AllChars* configuration  
ALLCHARS.DOC - *AllChars* manual (in Word format)  
FILE\_ID.DIZ - short information about *AllChars*  
HISTORY.TXT - Version history information  
README.TXT - a bit about *AllChars*

## Appendix C - ANSI-table and AllChars key-combinations

ANSI	chars	ANSI	chars	ANSI	chars	ANSI	chars
0128	€	0160	s p	0192	À ` A	0224	à ` a
0129		0161	¡ ^ !	0193	Á ' A	0225	á ' a
0130	, . ,	0162	¢ / c	0194	Â ^ A	0226	â ^ a
0131	f - f	0163	£ - L	0195	Ã ~ A	0227	ã ~ a
0132	„ ^ "	0164	₪ x o	0196	Ä " A	0228	ä " a
0133	… 3 .	0165	¥ = Y	0197	Å o A	0229	å o a
0134	†   +	0166	¦	0198	Æ A E	0230	æ a e
0135	‡ + +	0167	§ * f	0199	Ç , C	0231	ç , c
0136	^ \ ^	0168	¨ \ "	0200	È ` E	0232	è ` e
0137	‰ % %	0169	© O C	0201	É ' E	0233	é ' e
0138	Š ^ S	0170	ª . a	0202	Ê ^ E	0234	ê ^ e
0139	< . <	0171	« > >	0203	Ë " E	0235	ë " e
0140	Œ O E	0172	¬ - ,	0204	Ì ` I	0236	ì ` i
0141		0173	¸ s -	0205	Í ' I	0237	í ' i
0142	Ž	0174	® O R	0206	Î ^ I	0238	î ^ i
0143		0175	¯ ^ _	0207	Ï " I	0239	ï " i
0144		0176	° . o	0208	Ð - D	0240	ð
0145	' . '	0177	± + -	0209	Ñ ~ N	0241	ñ ~ n
0146	' . '	0178	² . 2	0210	Ò ` O	0242	ò ` o
0147	" \ "	0179	³ . 3	0211	Ó ' O	0243	ó ' o
0148	" ' "	0180	´ \ '	0212	Ô ^ O	0244	ô ^ o
0149	• * .	0181	µ , u	0213	Õ ~ O	0245	õ ~ o
0150	- n -	0182	¶ q	0214	Ö " O	0246	ö " o
0151	— m -	0183	· . .	0215	× x x	0247	÷ : -
0152	~ \ ~	0184	, \ ,	0216	Ø / O	0248	ø / o
0153	™ T M	0185	¹ . 1	0217	Ù ` U	0249	ù ` u
0154	š ^ s	0186	º . 0	0218	Ú ' U	0250	ú ' u
0155	> . >	0187	» > >	0219	Û ^ U	0251	û ^ u
0156	œ o e	0188	¼ 1 4	0220	Ü " U	0252	ü " u
0157		0189	½ 1 2	0221	Ý ' Y	0253	ý ' y
0158	ž	0190	¾ 3 4	0222	Þ b p	0254	þ   o
0159	ÿ Y	0191	¿ ^ ?	0223	ß s s	0255	ÿ " y

Alternative combinations:

ANSI	chars	ANSI	chars	ANSI	chars	ANSI	chars
0131	f f f	0163	£ L L	0178	² ^ 2	0186	º ^ 0
0132	„ ^ "	0167	§ p p	0179	³ ^ 3	0188	¼ / 4
0150	- 2 -	0169	© c r	0182	µ m u	0189	½ / 2
0151	— 3 -	0174	® r m	0185	¶ a a	0190	¾ / 3
0161	j ! !	0176	° d g	0185	¹ ^ 1	0191	¿ ? ?