Modifying the WinIATE™ Terminal Keyboard Map

Contents

3095 Richmond Parkway, Suite 207
Richmond, CA 94806-1900 USA
(510) 222-7717

Introduction

This technical note explains how to change key assignments for WinIATE. The **termkeys** text file describes the key assignments. You can edit this file to change the assignments. Before modifying the file, be sure to make a backup of the original.

The key map file contains two sections, separated by a line that reads as follows:

```
*CHARS
```

The first section of the file, above the *CHARS line, contains key assignment specifications in the following format:

```
Command Keyword      Command Code   ,   Key Code part 1   Key Code part 2
```

The *Command Keyword* is text. The command and key codes are hexadecimal values. For example, one typical line in the first section might look like this:

```
CLEAR                0x80           ,              0xa1   0x00
```

The second section of the file, below the *CHARS line, contains key assignment specifications in a different format:

```
Command Keyword      Command Code    Key Code
```

For example, one typical line in the second section might look like this:

```
CLEAR                0x80           0x03
```

The two sections of the file contain two different kinds of *key codes* — dual values in the first section, and single values in the second section.

Edit **termkeys** using a full-screen text editor, such as the "Edit" program that comes included with MS-DOS® versions 5 and later, or word processing software that you may already own. If you are using word processing software such as Word Perfect®, be sure to edit the key map file as an "ASCII text file" or "DOS text file" — plain, unformatted text — and, when finished editing, save it back to disk in that form. For example, in Word Perfect, you'll need to use a special command to save a "DOS text file".

The spaces between columns in the file consist of "tabs", which you can generate by pressing the Tab key on your keyboard. The space-bar may also be used. When you're editing the file, the columns of command codes, code values, and commas should appear "lined up" vertically, with perhaps just a few "misaligned" entries here and there. If you're using word processing software, and the columns do not appear to be "lined up", try resetting your word processor's tabstop width to 8.

The *Command Code* and *Key Code* values appear in "hexadecimal" notation. You may not be familiar with the hexadecimal (base 16) number system. Understanding hexadecimal notation is *not* necessary in order to edit the key map file. You can edit the file even if you've never seen hexadecimal before. But in case you would nonetheless prefer to understand this notation, we've included a brief explanation in the Appendix the end of this document.

Command Keywords and Command Codes

You may have noticed that the *Command Code* values are listed in ascending numerical order, with a few duplicate values grouped together on consecutive lines. For every command, there is a unique *Command Keyword* and a unique *Command Code* number. In the few cases of duplicates, you'll notice that the keywords as well as the codes are identical, because the duplicated lines refer to the same command. The keywords are important for readability, but they are there mainly for our convenience, so that we can read the file and find the commands we're looking for. IATE ignores the names, and just reads the command code numbers to identify the different commands. So the format or spelling of a command keyword is not significant to the software (except that a command name never contains any blank space).

In the second section of the file, you'll notice that some "command keywords" consist of just one character (a letter, a number, or a punctuation symbol). These characters generally signify the corresponding keys on the keyboard. Some symbols are mapped to different keys to more closely approximate certain airline keyboard layouts.

It is possible to define more than one assignment for a given command, giving the user a choice of different keys that will invoke the same command. In that case you will see the same command code on more than one line in the file.

Key Codes

The Key Code value(s) on each line identify which key will invoke the command whose name appears on that line. In the first section of the file, the two Key Code values on the right-hand side of each line represent certain keys defined in Table 2. In the second section of the file, the Key Code values represent keys defined in Table 3.

(*Note:* In this discussion, when we refer to values of "zero" in the key map file, we mean the notation 0x0 or 0x00 in hexadecimal. All other hexadecimal values are "nonzero".)

Many of the lines in Section 1 contain a nonzero value in Key Code part 1, and a zero in Key Code part 2. These code pairs signify single keys. For example, consider the key code pair `0xa1 0x00`. If you look up the value `0xa1` in Table 2, you will find that it signifies the **PageUp** key on the cursor keypad. The second value of zero (`0x00` or `0x0`) is necessary to maintain the consistent format of the lines in Section 1; but because the second value is zero, it does not actually modify the meaning of the key assignment.

Several lines in Section 1 contain a nonzero value in both Key Code part 1 and Key Code part 2. On these lines, Key Code part 1 is either `0x10` or `0x11`. Table 2 shows that `0x10` signifies **Shift**, and `0x11` signifies **Ctrl**. This means that these code pairs signify **Shift** and **Ctrl** key combinations. For example, consider the key code pair `0x10 0x08`. Consulting Table 2 again, we find that that `0x08` signifies Backspace. So the code pair `0x10 0x08` signifies **Shift-Backspace**.

The remaining lines in Section 1 contain zero values in both Key Code part 1 and Key Code part 2. These lines do not assign a command to any key. The lines labeled with command name "NULL" are included merely to fill in gaps in the continuous ascending sequence of Command Code values. The "NULL" lines could be removed without affecting the operation of the terminal. (The "MOREACTION..." command lines also have key codes of zero, because the current version of WinIATE does not support the special action-key commands.)

Changing Key Assignments

The first table that we'll present below lists IATE commands and their abbreviated names in the **termkeys** file. The second table lists the key code pairs for Section 1, and the third table lists the key codes for Section 2 of the file.

To change a key assignment for a given command, look up the command in the first table, find it in the file, and check the key code(s) to which it's currently assigned. (Keep in mind that a given command be listed in more than one key assignment — possibly one in Section 1 and another in Section 2.) Then, in the second and / or third tables, look up the key code(s) that you want to assign to the command. Make sure that your chosen key code(s) are not already in use for any other command.

If you want to reassign key code(s) that are already in use for another command, it will be necessary also to revise the assignment for the other command. One possible solution is to "trade" key assignments between the two commands. Or you might reassign the other command to some other key that's not currently in use. If you find yourself working towards changing several assignments in order to get the one key assignment that you want, stop. We recommend that you choose some other key, because extensive changes can quickly lead to confusion; and also because it may not be desirable to diverge too much from standard airline keyboard layouts.

Please make a note of any changes you make to the key map file, and keep your notes filed for reference. It can be difficult to correct a modified key map without some record of the intended assignments. Keep a backup copy of the original file.

If possible, change no more than two or three assignments at a time, and then test IATE to find out whether your changes work.

For a description of what the commands do, please refer to your WinIATE User's Guide.

If you have any questions, please call InnoSys Technical Support.

Please Note :

> **Shift** and **Ctrl** key combinations can be assigned by using the codes for **Shift** and **Ctrl** in Key Code part 1, as explained in the **Key Codes** section. However, it is not currently possible to assign **Alt** key combinations. Furthermore, it is not possible to distinguish between the left-hand and right-hand **Shift** keys, or the dual **Ctrl** keys available on some keyboards.
>
> The **Caps Lock** key shifts the alphabetic keys. The key map file swaps upper and lower case letter codes so that the unshifted keys appear as upper case, and the shifted keys appear as lower case. If your host type cannot support lower case, you may wish to change the mapping so that the shifted keys generate upper case.

Table 1. Command Table

Following is a list of command keywords and codes that may be used in the **termkeys** file for the current release of WinIATE.

(*Note:* In earlier versions of the file, released before January 1995, some names will appear slightly different — for example, "ENTERCHAR" instead of "ENTER", and "RETURNCHAR" instead of "NEW_LINE". Also, many of the older names end with the suffix "CMD", which no longer appears in the newer command names. These differences are not significant to the software. The names were changed for consistency with recently released key map files for other versions of IATE. The numeric codes were not changed. If any confusion arises, or if you would like a copy of the latest **termkeys** file, please contact InnoSys.)

Not all commands are available on every type of host. For instance, only SABRE hosts support the PROTECT command, which turns on the terminal's Protected mode and moves the cursor to a protected field. Other hosts do not support lower case letters, etc.

Command Keyword in the TERMKEYS file	**Command Code** in TERMKEYS file	**Name of Command**	**Key Function Description**
ENTER	0x03	Enter	Send current entry to host
BACK_SPACE	0x08	Backspace	Delete one character and move left
GO_TO_TAB	0x09	Tab	Advance to next tab
NEW_LINE	0x0d	Carriage Return	Advance to beginning of next line
LEFT_ARROW	0x1c	Left Arrow	Move left 1 space
RIGHT_ARROW	0x1d	Right Arrow	Move right 1 space
UP_ARROW	0x1e	Up Arrow	Move up 1 line
DOWN_ARROW	0x1f	Up Arrow	Move down 1 line
SPACE	0x20	Blank Space	Insert 1 blank space
"!" through ")"	0x21 through 0x29	Punctuation	Several of the punctuation symbols
DISPLAY(*)	0x2a	SABRE Display	SABRE Display character
"+" through "/"	0x2b through 0x2f	Punctuation	More punctuation symbols
"0" through "9"	0x30 through 0x39	Numeric digits	The number keys (digits, not arrows)
":" through "?"	0x3a through 0x3f	Punctuation	More punctuation symbols
PARS_CHANGE(@)	0x40	PARS Change	PARS Change character
"A" through "Z"	0x41 through 0x5a	Upper-case letters	
"\"	0x5c	Punctuation	
FIELD_MARK	0x5e	Field Mark	Field Mark character
SABRE_CHANGE(_)	0x5f	SABRE Change	SABRE Change character
"a" through "z"	0x61 through 0x7a	Lower-case letters	
"{"	0x7b	Punctuation	

Command Keyword in the TERMKEYS file	**Command Code** in TERMKEYS file	**Name of Command**	**Key Function Description**
CROSS_OF_LORRAINE	0x7c	Cross of Lorraine	SABRE Cross of Lorraine
DELETE_TO_EOL	0x7d	Delete to End of Line	Erase from cursor to end of line
SET_OPT_FIELD	0x7e	SABRE Set Opt. Field	(Also known as "Lightning Bolt")
CLEAR	0x80	Clear	Erase all text in the current window
DELETE_TO_EOP	0x81	Clear to End of Page	Erase from cursor to end of window
CLEAR_SERVICE	0x82	Clear Service	Clear a displayed service message
PRINT_WINDOW	0x83	Print Window	Print contents of current window
HOME	0x84	Home	Move to upper left corner
SOM	0x85	Start of Message	Insert SOM character
RESET	0x86	Reset	Reset terminal (unlock keyboard)
REENTER	0x87	Re-enter	Send Re-Enter to host

Command Keyword in the TERMKEYS file	**Command Code** in TERMKEYS file	**Name of Command**	**Key Function Description**
MORE_ACTION_1	0xcc	Special Action #1	*Currently unsupported in WinIATE*
MORE_ACTION_2	0xcd	Special Action #2	*Currently unsupported in WinIATE*
MORE_ACTION_3	0xce	Special Action #3	*Currently unsupported in WinIATE*
MORE_ACTION_4	0xcf	Special Action #4	*Currently unsupported in WinIATE*
MORE_ACTION_5	0xd0	Special Action #5	*Currently unsupported in WinIATE*
MORE_ACTION_6	0xd1	Special Action #6	*Currently unsupported in WinIATE*
FUNCTION_1	0xe2	Function key #1	Programmable function key
FUNCTION_2	0xe3	Function key #2	Programmable function key
FUNCTION_3	0xe4	Function key #3	Programmable function key
FUNCTION_4	0xe5	Function key #4	Programmable function key
FUNCTION_5	0xe6	Function key #5	Programmable function key
FUNCTION_6	0xe7	Function key #6	Programmable function key
FUNCTION_7	0xe8	Function key #7	Programmable function key
FUNCTION_8	0xe9	Function key #8	Programmable function key
FUNCTION_9	0xea	Function key #9	Programmable function key
FUNCTION_10	0xeb	Function key #10	Programmable function key
FUNCTION_11	0xec	Function key #11	Programmable function key
FUNCTION_12	0xed	Function key #12	Programmable function key
FUNCTION_13	0xee	Function key #13	Programmable function key
FUNCTION_14	0xef	Function key #14	Programmable function key
FUNCTION_15	0xf0	Function key #15	Programmable function key
FUNCTION_16	0xf1	Function key #16	Programmable function key
FUNCTION_17	0xf2	Function key #17	Programmable function key
FUNCTION_18	0xf3	Function key #18	Programmable function key
FUNCTION_19	0xf4	Function key #19	Programmable function key
FUNCTION_20	0xf5	Function key #20	Programmable function key
FUNCTION_21	0xf6	Function key #21	Programmable function key
FUNCTION_22	0xf7	Function key #22	Programmable function key
FUNCTION_23	0xf8	Function key #23	Programmable function key
FUNCTION_24	0xf9	Function key #24	Programmable function key
FUNCTION_25	0xfa	Function key #25	Programmable function key
FUNCTION_26	0xfb	Function key #26	Programmable function key
FUNCTION_27	0xfc	Function key #27	Programmable function key
FUNCTION_28	0xfd	Function key #28	Programmable function key
FUNCTION_29	0xfe	Function key #29	Programmable function key
FUNCTION_30	0xff	Function key #30	Programmable function key

Table 2. Key Code Table for Section 1

The table below shows the key codes that WinIATE can recognize in the first section of the key map file. Refer to the earlier discussion of **Key Codes** for information on how these codes must be combined into pairs (Key Code part 1 and Key Code part 2).

Note :

For commands not assigned to keys, or for assignments where Key Code part 2 is zero, the key map file will contain zero values as placeholders. In hexadecimal notation, either `0x0` or `0x00` can be used to indicate a value of zero.

Key	Code	
Backspace	0x08	same as Ctrl-H
Tab	0x09	same as Ctrl-I
Enter	0x0d	The Enter (or Return) key
Shift	0x10	Key Code part 1 for shifted keys
Ctrl	0x11	Key Code part 1 for Ctrl key combinations
PageUp	0x21	PageUp key on numeric keypad
PageUp	0xa1	PageUp key on separate cursor keypad
PageDown	0x22	PageDown key on numeric keypad
PageDown	0xa2	PageDown key on separate cursor keypad
End	0x23	End key on numeric keypad
End	0xa3	End key on separate cursor keypad
Home	0x24	Home key on numeric keypad
Home	0xa4	Home key on separate cursor keypad
Left-Arrow	0x25	Left-Arrow key on numeric keypad
Left-Arrow	0xa5	Left-Arrow key on separate cursor keypad
Up-Arrow	0x26	Up-Arrow key on numeric keypad
Up-Arrow	0xa6	Up-Arrow key on separate cursor keypad
Right-Arrow	0x27	Right-Arrow key on numeric keypad
Right-Arrow	0xa7	Right-Arrow key on separate cursor keypad
Down-Arrow	0x28	Down-Arrow key on numeric keypad
Down-Arrow	0xa8	Down-Arrow key on separate cursor keypad
Insert	0x2d	Insert key on numeric keypad
Insert	0xad	Insert key on separate cursor keypad
Delete	0x2e	Delete key on numeric keypad
Delete	0xae	Delete key on separate cursor keypad
Keypad 0	0x60	0 key on numeric keypad, with NumLock *on*
Keypad 1	0x61	1 key on numeric keypad, with NumLock *on*
Keypad 2	0x62	2 key on numeric keypad, with NumLock *on*
Keypad 3	0x63	3 key on numeric keypad, with NumLock *on*
Keypad 4	0x64	4 key on numeric keypad, with NumLock *on*
Keypad 5	0x65	5 key on numeric keypad, with NumLock *on*
Keypad 5	0x0c	5 key on numeric keypad, with NumLock ***off***
Keypad 6	0x66	6 key on numeric keypad, with NumLock *on*
Keypad 7	0x67	7 key on numeric keypad, with NumLock *on*
Keypad 8	0x68	8 key on numeric keypad, with NumLock *on*

Key	Code	
Keypad 9	0x69	9 key on numeric keypad, with NumLock *on*
Keypad *	0x6a	* key on numeric keypad
Keypad +	0x6b	+ key on numeric keypad
Keypad -	0x6d	- key on numeric keypad
Keypad .	0x6e	. key on numeric keypad
Keypad /	0x6f	/ key on numeric keypad
F1	0x70	
F2	0x71	
F3	0x72	
F4	0x73	
F5	0x74	
F6	0x75	
F7	0x76	
F8	0x77	
F9	0x78	
F10	0x79	
F11	0x7a	
F12	0x7b	

Table 3. Key Code Table for Section 2

The table below shows the key codes that WinIATE can recognize in the second section of the key map file. The second section begins immediately after the line in the file that reads:

```
*CHARS
```

Note :

For commands not assigned to keys, or for assignments where Key Code part 2 is zero, the key map file will contain zero values as placeholders. In hexadecimal notation, either `0x0` or `0x00` can be used to indicate a value of zero.

Key	Code	
Ctrl-K	0x0b	
Ctrl-L	0x0c	
Ctrl-N	0x0e	
Ctrl-O	0x0f	
Ctrl-P	0x10	
Ctrl-Q	0x11	
Ctrl-R	0x12	
Ctrl-S	0x13	
Ctrl-T	0x14	
Ctrl-U	0x15	
Ctrl-V	0x16	
Ctrl-W	0x17	
Ctrl-X	0x18	
Ctrl-Y	0x19	
Ctrl-Z	0x1a	
Ctrl-[	0x1b	
Escape	0x1b	same as Ctrl-[
Ctrl-\	0x1c	
Ctrl-]	0x1d	
Ctrl-^	0x1e	
Ctrl-_	0x1f	
Space	0x20	
!	0x21	
"	0x22	
#	0x23	
$	0x24	
%	0x25	
&	0x26	
'	0x27	
(	0x28	
)	0x29	
*	0x2a	* key on main keyboard, not numeric keypad
+	0x2b	+ key on main keyboard, not numeric keypad
,	0x2c	

Key	Code	
-	0x2d	- key on main keyboard, not numeric keypad
.	0x2e	. key on main keyboard, not numeric keypad
/	0x2f	/ key on main keyboard, not numeric keypad
0	0x30	0 key on main keyboard, not numeric keypad
1	0x31	1 key on main keyboard, not numeric keypad
2	0x32	2 key on main keyboard, not numeric keypad
3	0x33	3 key on main keyboard, not numeric keypad
4	0x34	4 key on main keyboard, not numeric keypad
5	0x35	5 key on main keyboard, not numeric keypad
6	0x36	6 key on main keyboard, not numeric keypad
7	0x37	7 key on main keyboard, not numeric keypad
8	0x38	8 key on main keyboard, not numeric keypad
9	0x39	9 key on main keyboard, not numeric keypad
:	0x3a	
;	0x3b	
<	0x3c	
=	0x3d	
>	0x3e	
?	0x3f	
@	0x40	
A	0x41	(*Note:* Codes for Lower-Case letters are
B	0x42	shown later in this table.)
C	0x43	
D	0x44	
E	0x45	
F	0x46	
G	0x47	
H	0x48	
I	0x49	
J	0x4a	
K	0x4b	
L	0x4c	
M	0x4d	
N	0x4e	
O	0x4f	
P	0x50	
Q	0x51	
R	0x52	
S	0x53	
T	0x54	
U	0x55	
V	0x56	
W	0x57	
X	0x58	
Y	0x59	
Z	0x5a	
[	0x5b	
\	0x5c	

Key	Code
]	0x5d
^	0x5e
_	0x5f
`	0x60
a	0x61
b	0x62
c	0x63
d	0x64
e	0x65
f	0x66
g	0x67
h	0x68
i	0x69
j	0x6a
k	0x6b
l	0x6c
m	0x6d
n	0x6e
o	0x6f
p	0x70
q	0x71
r	0x72
s	0x73
t	0x74
u	0x75
v	0x76
w	0x77
x	0x78
y	0x79
z	0x7a
{	0x7b
|	0x7c
}	0x7d
~	0x7e

(*Note:* Codes for Upper-Case letters are shown earlier in this table.)

Appendix: Hexadecimal Notation

It is not necessary to understand hexadecimal notation in order to edit a key map file. The hexadecimal values in the file are simply numbers written in this special notation. You do not need to know how to decipher these values; you need only use the values as given in the tables shown above. We present the following explanation for those of you who would nonetheless prefer to have a basic understanding of the notation.

InnoSys chose to use hexadecimal instead of decimal notation in key map files because hexadecimal was more convenient to use when programming and debugging the IATE software.

In hexadecimal notation, each digit counts from zero to fifteen. Numbers zero through nine can be written just the same as for a decimal number; but numbers ten through fifteen are represented by letters "a" through "f":

```
0 1 2 3 4 5 6 7 8 9 a b c d e f
```

The letters may be upper or lower case. In a key map file, we prefix these values with "`0x`", a common computer notation to indicate that we're using hexadecimal. So, for instance, "`0x1`" and "`0x9`" represent values one and nine respectively. "`0xa`" means ten, "`0xb`" means eleven, and so on through "`0xf`" for fifteen.

After the value "`f`" for fifteen, the next value in hexadecimal requires two digits. Just as 9 + 1 = 10 in decimal, similarly `0xf + 0x1 = 0x10` in hexadecimal. And the values continue from there. Hopefully you're beginning to get the idea of counting in hexadecimal:

```
0 1 2 3 4 5 6 7 8 9 a b c d e f 10 11 12 13 14 15 16 17 18 19 1a 1b 1d 1d 1e 1f 20 21…
```

and so on.

To figure out what a two-digit hexadecimal number means, you can multiply the left-hand digit by sixteen, and add the right-hand digit. For example, the hexadecimal value `0x14` is equal to twenty, because it is sixteen plus four: `0x10 + 0x4 = 16 + 4 = 20`. In a key map file, there may also be a few three-digit values; in these cases the leftmost digit is a multiple of 256. (256 is sixteen times sixteen). So the value `0x124` is equal to `(1 x 256) + (2 x 16) + (4 x 1) = 256 + 32 + 4 = 292`.

Extra zeroes immediately following the "`0x`" are not significant; for example, `0x0c` is just the same as `0xc` — both indicate a value of twelve. And `0x012` or `0x12` would both indicate eighteen (sixteen plus two). The extra zeroes are insignificant. (These zeroes can be included merely to align values vertically with adjacent multi-digit values above and below, for ease of reading.)

Again, we want to remind you that you can edit a key map file using the tables and explanations given earlier, with no need to decipher the hexadecimal values provided.