

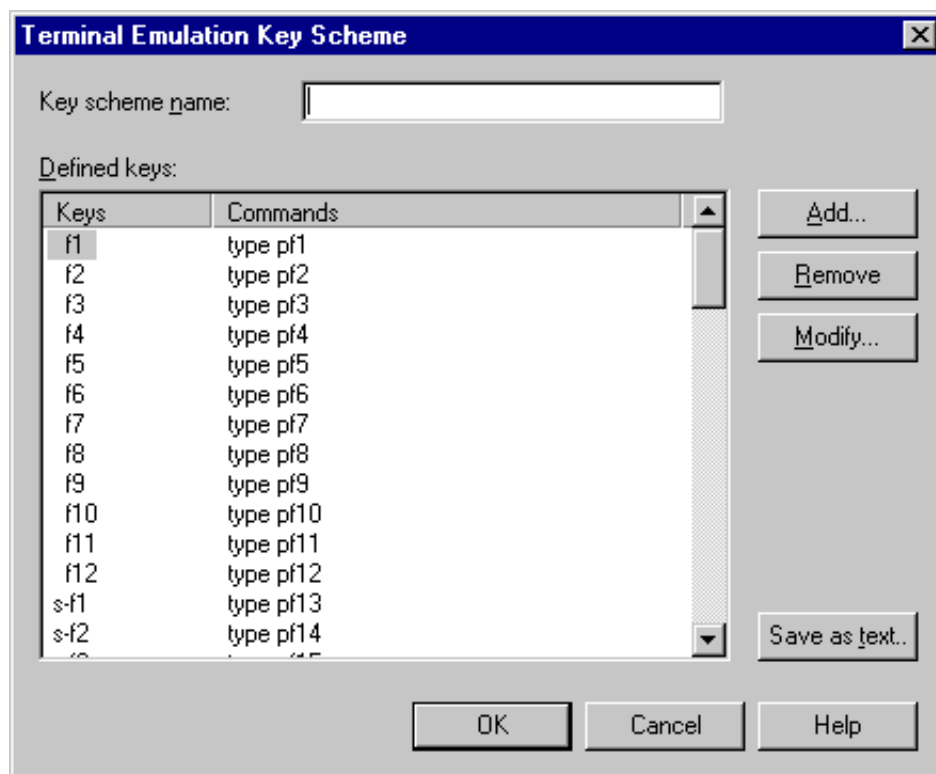
# Key Schemes

This chapter covers the following topics:

- Defining a Key Scheme
- Writing the Key Definitions to a File
- Terminal Emulation Keys
- Processing of Keyboard Input

## Defining a Key Scheme

The **Terminal Emulation Key Scheme** dialog box lets you define the keys to be used for a specific key scheme. The key scheme to be used in a session is defined in the session properties.



The following table explains the abbreviations that are used in this dialog box. For example, the abbreviation "a-b" stands for the key combination ALT+B.

Abbreviation	represents the following key
a-	ALT
c-	CTRL
f-	Function key
gray-	Gray key
n-	Key on numeric keypad
s-	SHIFT

**Notes:**

1. The shortcut keys that are displayed next to a menu command in the terminal application (e.g. CTRL+V for pasting text) are not available when the active key scheme uses them for different purposes.
2. The key combinations CTRL+RIGHT-ARROW, CTRL+LEFT-ARROW and CTRL+END have been hardcoded. They are used for jumping to the next word, the previous word, and to the end of the field. These functions are only available in the terminal application if the corresponding key combinations are not defined otherwise in the active key scheme.

The **Terminal Emulation Key Scheme** dialog box provides the following command buttons (in addition to the standard command buttons):

<b>Add</b>	Add a key to the key scheme.
<b>Remove</b>	Remove the selected key.
<b>Modify</b>	Modify the key definition for the selected key.
<b>Save as text</b>	Write the current key definitions to a file.

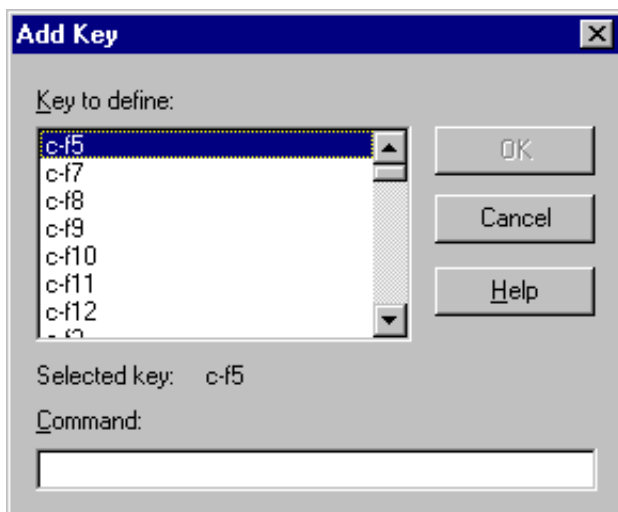
 **To define a key scheme**

1. When you create a new key scheme, specify a name in the **Key scheme name** text box.

The name can be up to 32 characters long and may contain blanks. When you modify a key scheme, this text box cannot be modified.

2. Choose the **Add** button.

The **Add Key** dialog box appears.



3. In the **Key to define** list box, select the key to be added.
4. In the **Command** text box, enter the command you want to assign to the selected key. For example:

Command	Description
EXECTASK DSYSTRANS	Executes the task DSYSTRANS.
EXECUTE LOGOFF	Executes the procedure file <i>Logoff.ncp</i> .
TYPE 'SAAAA' CR	Sends the characters SAAAA and CR (ENTER) to the host.

5. Choose the **OK** button.

The key is now added to the **Terminal Emulation Key Scheme** dialog box.

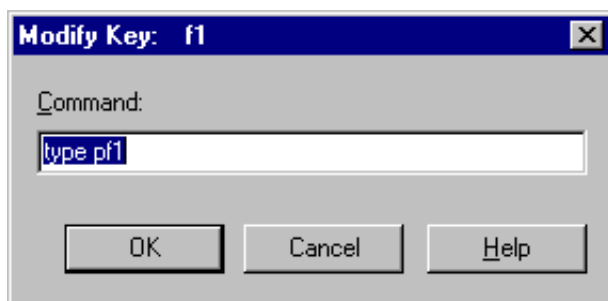
6. Optionally: choose the **Remove** button to remove the selected key from the key scheme.

A dialog box appears, asking whether you want to remove the key.

7. Optionally: double-click a key to modify the command in a dialog box.

Or:

Select the key and choose the **Modify** button.



8. In the **Terminal Emulation Key Scheme** dialog box, choose the **OK** button to save the key scheme.

## Writing the Key Definitions to a File

You can write the key definitions for the currently shown key scheme to a file from which you can print them.

### To write the key definitions to a file

1. In the **Terminal Emulation Key Scheme** dialog box, choose the **Save as text** button.

The **Save As** dialog box appears.

2. Specify a file name.

The default extension is *txt*.

3. Choose the **Save** button.

The commands assigned to each key are written to the specified file.

## Terminal Emulation Keys

A distinction is made between terminal function keys and physical function keys:

- Terminal Function Keys

This terms refers to all keys that start a terminal function. Different host systems have different terminal functions and associated keys. Examples for IBM 3270 hosts: PF1, PF2, ATTN, PA1. Examples for Siemens hosts: F1, K1, FKT1, DUE1.

The code sequence eventually sent to the host for a given terminal function key is taken from the terminal function code table. The share file contains terminal function code tables for all communication methods. The administrator can modify the terminal function code tables.

- Physical Function Keys

This terms refers to all keys on the physical keyboard that can be used as function keys. Examples: F1, F2, CTRL+F3, CTRL+A, ALT+B. Entire Connection has a unique name for each function key.

## Terminal Function Keys

Some terminal function key names are found on all host systems, others are specific to a given host system. Entire Connection supports the following:

- 3270 Function Key Names
- 9750 Function Key Names
- VT Function Key Names

### 3270 Function Key Names

The following key names can be used with the TYPE command to transmit the corresponding 3270 key to the host:

ATTN	DOWN	LEFT	RIGHT
BACKSPACE	EEOF	NEWLINE	SYSREQ
BACKTAB	ERASEINP	PA1 to PA3	TAB
CLEAR	HOME	PF1 to PF24	TEST
CR	IDENT	POR	UP
DELETE	INSERT	PRINT	
DEVCNCL	JUMP	RESET	

If any of the following key names is used with the TYPE command, Entire Connection will automatically wait for a response from the host before continuing:

ATTN  
 CLEAR  
 CR  
 POR  
 PA1 to PA3  
 PF1 to PF24

### 9750 Function Key Names

The following key names can be used with the TYPE command to transmit the corresponding 9750 key to the Siemens BS2000 host:

AFG	FKT1 to FKT24	P1 to P20	SMU
AFZ	K1 to K14	RU	SNZ
DUE1	LSP	SBA	SZA
DUE2	LVD	SDZ	TABL
EFG	LZE	SML	TABR
EFZ	LZF	SMO	
ENDM	MAR	SMR	

### VT Function Key Names

The following key names can be used with the TYPE command to transmit the corresponding VT-type key to the host:

AKEY1 to AKEY15
^A to ^Z (where ^ stands for CTRL)

## Physical Function Keys

Physical function keys are all keys and key combinations that can be used as function keys by Entire Connection. In the **Terminal Emulation Key Scheme** dialog box, you can assign the following to a function key:

- a command
- a command string
- an emulation key
- a combination of the above possibilities.

When the function key is pressed, the assigned action is executed.

All physical function keys have symbolic names in Entire Connection. These symbolic names are displayed in the **Terminal Emulation Key Scheme** dialog box and can be used with the commands `TYPE` and `SET TEKEY` in a procedure file.

Note that the **Key to define** list box in the **Add Key** dialog box contains more symbolic names than illustrated in the tables below (for example, "xf1"). These symbols can be used with keyboard types that have more keys. See the file *Keyboard\_us.kyb* in the *tables* folder of Entire Connection for details.

The following table lists the Entire Connection physical function keys and indicates the combinations in which these keys can be used.

Key	single key	SHIFT	CTRL	ALT
A through Z			X	X
BACKTAB		X		
BKSPC			X	X
CENTER	X		X	
CR	X		X	X
DELBACK	X			
DELETE	X		X	
DOWN	X		X	
END	X		X	
ESC	X			X
F1 to F12	X	X	X	X
HOME	X		X	
INSERT	X		X	
LEFT	X		X	
PGDN	X		X	
PGUP	X		X	

Key	single key	SHIFT	CTRL	ALT
PRTSC			X	
RIGHT	X		X	
TAB	X		X	
UP	X		X	
0 to 9				X
2 and 6			X	
n-0 to n-.	X			
GRAY *	X		X	X
GRAY /	X		X	X
\			X	X
[			X	X
]			X	X
-			X	X
GRAY -	X		X	X
=				X
GRAY +	X			X
;				X
,				X
.				X
QUOTE				X

**Note:**

The names n-0 to n-. represent the keys on the numeric keypad. You can also transmit these keys by pressing NUM.

Additional key support for IBM 101 enhanced keyboards:

Key	single key	SHIFT	CTRL	ALT
DELETE (gray)	X			
DOWN (gray)	X			
END (gray)	X			
HOME (gray)	X			
INSERT (gray)	X			
LEFT (gray)	X			
PGDN (gray)	X			
PGUP (gray)	X			
RIGHT (gray)	X			
UP (gray)	X			

## Processing of Keyboard Input

When you press a function key, the following occurs:

1. Entire Connection takes the keyboard code received from the Windows operating system and scans the physical keyboard table for the name of the physical function key.
2. If the name of the physical function key is found, Entire Connection checks the keys that have been defined for terminal emulation. If the function key is not assigned, the key is passed to Windows for further processing. If the function key is assigned, the key definition is passed to the communication modules.
3. The communication modules parse the key definition. The key definition may consist of several elements, including the name of a terminal function. For example:

```
TYPE 'username' CR
```

4. If the key definition contains the name of a terminal function (CR in the above example), Entire Connection scans the current terminal function code table for the code sequence of this terminal function, and passes the code to the host.