ESC f (n) — Sets Line Height

Format

ASCII value: ESC f n ;

HEX value: 1B 66 n 3B

Description

Sets the line height—applies to entire line.

Where *n* (0x30 – 0x33) determines the Height:

0 = Normal Height
1 = Double Height
2 = *reserved*
3 = *reserved*

ESC h (n) — Select Character Set

Format

ASCII value: ESC h n ;

HEX value: 1B 68 n 3B

Description

Selects the country character set.

Where *n* (0x30–0x39) determines the character set:

0 = U.S.A. (default)
1 = France
2 = Germany
3 = United Kingdom
4 = Denmark I
5 = Sweden
6 = Italy
7 = Spain
8 = Japan
9 = Norway

ESC i — Return Printer Indentification

Format

ASCII value: ESC i

HEX value: 1B 69

Description

Returns the printer identification number. The OMNI 460 returns an uppercase "A".

A P P E N D I X

E

VeriFone Fonts

OMNI 39x Series terminals feature a pixel-type display that provides the ability to display 1 to 4 lines of ASCII, international or graphical characters. OMNI 39x terminals are shipped with an assortment of character and graphic sets, called fonts. Each font comprises up to 128 characters or graphics of one size and language. Fonts are stored in in the terminal's ROM as font *pages*. Each terminal may hold up to 16 font pages (numbered 0–15).

Character Size

Characters are measured by pixel width and height. For example, an 8 x 16 character is 8 pixels wide and 16 pixels high; this is the character size used for the default ASCII two-line by 18-character display. Table E-1 lists all the character sizes available for OMNI 39x fonts.

Table E-1. Font Character and Display Sizes

Character Size Width x Ht	Display Size Line x Chars	Grid ID
8 x 16	2 x 18	GRID_2x18 (0)
8 x 10	3 x 18	GRID_3x18 (1)
6 x 8	4 x 25	GRID_4x25 (2)
16 x 16	2 x 9 (double-wide)	GRID_2x18 (0)

All characters within a font are of one size only. For example, font 11A3112E is the ASCII set of 8 x 16 size characters, while font 11A1112E is the ASCII set of 6 x 8 size characters.

Font Organization

VeriFone font ROMs contain an assortment of up to 16 fonts. Each font ROM is tailored to meet the language requirements of a specific country. For example, font ROM 1162A11E includes three ASCII and two Chinese character fonts, and the assortment of Asia-Pacific graphic characters.

Extended Fonts

Some fonts, such as the Chinese character set (font ROM 1162A11E), may include more than 128 characters, and therefore require more than one font page for storage. For example, the first 128 Chinese characters are stored in font 15N4111E, the next 128 characters are stored in font 15N4122E, and the remaining characters are stored in font 15N4131E.

Displaying Characters

To display a character in a specific font, first identify the font ROM page (0–15) containing the desired character, then find the character's offset number. See *Chapter 8, Pixel-type Display, Raw Mode Character Selection.* Also see the example file *RAW.C.*

Retrieving Font Information

To determine the fonts available in your font ROM, use `getfontinfo()`; this function returns the name of your font ROM and other font information. Once you have the name of your font ROM, refer to its Font Set table (provided later in this appendix). The Font Set table will identify all the fonts contained in your font ROM, listing the page number (0–15), name and character size of each font.

Once you have the font number, you can refer to its Font Character Table; this table lists each character and its offset for your programming convenience and quick reference.

Font Set Tables

The following tables are organized numerically by font ROM number. Each table lists the page number, name and character size of each font in the font ROM, as well as each character in the font. *Note: See Chapter 8, System Devices, Pixel-type Display, for more information on fonts and font ROMs.*

Table E-2. 1112A11E—USA Font Set

Font Page	Font Name	Font Type	Size line x character	Font Definition File
Page 0	11A3112E	ASCII	2 x 18	ASCIIS0.FDF
Page 1	11A2112E	ASCII	3 x 18	ASCIIS1.FDF
Page 2	11A1112E	ASCII	4 x 25	ASCIIS2.FDF
Page 3	17N4112E	Logo	2 x 9	VFIRAW0.FDF
Page 4	14N4112E	Chinese Refined	2 x 9	VFIRAW0.FDF
Page 5	14N4122E	Chinese Refined Extended	2 x 9	VFIRAW0.FDF
Page 6	14N4132E	Chinese Refined Extended	2 x 9	VFIRAW0.FDF

Table E-3. 1123A11E—Thailand Font Set "A"

Font Page	Font Name	Font Type	Size line x character	Font Definition File
Page 0	11A3112E	ASCII	2 x 18	ASCIIS0.FDF
Page 1	11A2112E	ASCII	3 x 18	ASCIIS1.FDF
Page 2	11A1112E	ASCII	4 x 25	ASCIIS2.FDF
Page 3	17N4112E	Logo	2 x 18	VFIRAW0.FDF
Page 4	16N3111E	Thailand A	2 x 18	VFIRAW0.FDF
Page 5	16N3121E	Thailand A Extended	2 x 18	VFIRAW0.FDF
Page 6	18N4112E	ASPAC Logo	2 x 9	VFIRAW0.FDF

Table E-4. 1124A11E—Thailand Font Set "B"				
Font Page	**Font Name**	**Font Type**	**Size line x character**	**Font Definition File**
Page 0	11A3112E	ASCII	2 x 18	ASCIIS0.FDF
Page 1	11A2112E	ASCII	3 x 18	ASCIIS1.FDF
Page 2	11A1112E	ASCII	4 x 25	ASCIIS2.FDF
Page 3	17N4112E	Logo	2 x 9	VFIRAW0.FDF
Page 4	16N3112E	Thailand B	2 x 18	VFIRAW0.FDF
Page 5	16N3122E	Thailand B Extended	3 x 18	VFIRAW0.FDF
Page 6	18N4113E	ASPAC Logo	2 x 9	VFIRAW0.FDF

Table E-5. 1134A11E—Saudi Arabia Font Set				
Font Page	**Font Name**	**Font Type**	**Size line x character**	**Font Definition File**
Page 0	11A3112E	ASCII	2 x 18	ASCIIS0.FDF
Page 1	11A2112E	ASCII	3 x 18	ASCIIS1.FDF
Page 2	11A1112E	ASCII	4 x 25	ASCIIS2.FDF
Page 3	17N4112E	Logo	2 x 9	VFIRAW0.FDF
Page 4	12N2112E	Arabic	3 x 18	VFIRAW1.FDF
Page 5	12N3112E	Arabic	2 x 18	VFIRAW0.FDF

Table E-6. 1143A11E—Taiwan Font Set				
Font Page	**Font Name**	**Font Type**	**Size line x character**	**Font Definition File**
Page 0	11A3112E	ASCII	2 x 18	ASCIIS0.FDF
Page 1	11A2112E	ASCII	3 x 18	ASCIIS1.FDF
Page 2	11A1112E	ASCII	4 x 25	ASCIIS2.FDF
Page 3	17N4112E	Logo	2 x 9	VFIRAW0.FDF
Page 4	14N4112E	Chinese Refined	2 x 9	VFIRAW0.FDF
Page 5	14N4122E	Chinese Refined Extended	2 x 9	VFIRAW0.FDF
Page 6	14N4132E	Chinese Refined Extended	2 x 9	VFIRAW0.FDF
Page 7	18N4112E	ASPAC Logo	2 x 9	VFIRAW0.FDF

Table E-7. 1153A11E—Hong Kong Font Set				
Font Page	**Font Name**	**Font Type**	**Size line x character**	**Font Definition File**
Page 0	11A3112E	ASCII	2 x 18	ASCIIS0.FDF
Page 1	11A2112E	ASCII	3 x 18	ASCIIS1.FDF
Page 2	11A1112E	ASCII	4 x 25	ASCIIS2.FDF
Page 3	17N4112E	Logo	2 x 9	VFIRAW0.FDF
Page 4	13N4111E	Cantonese	2 x 9	VFIRAW0.FDF
Page 5	13N4121E	Cantonese Extended	2 x 9	VFIRAW0.FDF
Page 6	18N4112E	ASPAC Logo	2 x 9	VFIRAW0.FDF

Table E-8. 1162A11E—People's Republic of China Font Set				
Font Page	**Font Name**	**Font Type**	**Size line x character**	**Font Definition File**
Page 0	11A3112E	ASCII	2 x 18	ASCIIS0.FDF
Page 1	11A2112E	ASCII	3 x 18	ASCIIS1.FDF
Page 2	11A1112E	ASCII	4 x 25	ASCIIS2.FDF
Page 3	17N4112E	Logo	2 x 9	VFIRAW0.FDF
Page 4	15N4111E	Chinese Simplified	2 x 9	VFIRAW0.FDF
Page 5	15N4122E	Chinese Simplified Extended	2 x 9	VFIRAW0.FDF
Page 6	15N4131E	Chinese Simplified Extended	2 x 9	VFIRAW0.FDF
Page 7	18N4112E	ASPAC Logo	2 x 9	VFIRAW0.FDF

Table E-9. 117xA11E—Korean (Hangul) Font Set				
Font Page	**Font Name**	**Font Type**	**Size line x character**	**Font Definition File**
Page 0	11A3112E	ASCII	2 x 18	ASCIIS0.FDF
Page 1	11A2112E	ASCII	3 x 18	ASCIIS1.FDF
Page 2	11A1112E	ASCII	4 x 25	ASCIIS2.FDF
Page 3	17N4112E	Logo	2 x 9	VFIRAW0.FDF
Page 4	19N4111E	Hangul	2 x 9	VFIRAW0.FDF
Page 5	19N4121E	Hangul Extended	2 x 9	VFIRAW0.FDF
Page 6	19N4131E	Hangul Extended	2 x 9	VFIRAW0.FDF

Font Character Tables

The following Font Character Tables are organized numerically by font number. Each font may have up to 128 characters or graphics, numbered 0–127. The Font Character Tables provide the decimal and hexadecimal values for each character in the font. To select a character for display, simply write the character's decimal or hexadecimal offset value. For example, write a 0 to display the first character in the font, or write a 4 to display the fifth character.

❖ *See Chapter 8, System Devices, for more details on font page and character selection.*

Table E-10. 11A1112E—ASCII Font for 4-line Display

Decimal	Hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0																
16	1																
32	2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
48	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
96	6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	

Table E-11. 11A2112E—ASCII Font for 3-line Display

Decimal	Hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0																
16	1																
32	2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
48	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
96	6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	

Table E-12. 11A3112E–ASCII Font for 2-line Display

Decimal	Hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0																
16	1																
32	2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
48	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
96	6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	

Table E-13. 12N2112E—Saudi Font for 3-line Display

Decimal	Hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0																
16	1																
32	2																
48	3																
64	4																
80	5																
96	6																
112	7																

This font supersedes the 12N2111E Saudi font.

Table E-14. 12N3112E—Saudi Font for 2-line Display

Decimal	Hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0																
16	1																
32	2																
48	3																
64	4																
80	5																
96	6																
112	7																

This font supersedes the 12N3111E Saudi font.

Table E-15. 13N4111E—Cantonese/Hong Kong Font

Decimal	Hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0																
16	1																心
32	2	入	小	已	不	中	戶	日	月	代	功	卡	可	失	必	本	未
48	3	正	用	示	交	再	列	印	回	在	存	年	成	收	有	此	行
64	4	別	即	否	完	序	改	更	束	刷	取	始	放	易	法	表	金
80	5	倌	品	度	按	是	查	重	限	候	原	息	效	時	消	納	能
96	6	記	訊	起	送	退	除	商	售	執	密	專	將	接	掛	授	敗
112	7	啟	器	清	理	處	首	通	單	報	復	提	期	款	測	無	發

Table E-16. 13N4121E—Cantonese/Hong Kong Extended Font

Decimal	Hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0																
16	1																
32	2	稍	等	筆	簡	結	絡	開	傳	新	滑	照	當	腦	號	試	話
48	3	詢	路	過	途	電	對	滿	與	認	誤	銀	需	撥	數	槽	確
64	4	碼	編	線	請	賬	機	機	輸	辨	錯	錄	聽	檔	檢	總	聯
80	5	謝	鍵	斷	攝	其	覆	銷	額	識	權	聽	讀	顯	唔	店	告
96	6	脫	打	那	明	細	閱	保	個								
112	7																

Table E-17. 14N4112E—Chinese Refined Font

Decimal	Hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	至	集	續	連	特	約	沒	參	考	資	心	腦	主	要	動	任
16	1	意	核	指	令	後	美	止	別	只	刪	增	受	國	做	停	佔
32	2	入	小	已	不	中	戶	日	月	代	功	卡	可	失	必	本	未
48	3	正	用	示	交	再	列	印	回	在	存	年	成	收	有	此	行
64	4	別	即	否	完	序	改	更	束	刷	取	始	放	易	法	表	金
80	5	倌	品	度	按	是	查	重	限	候	原	息	效	時	消	納	能
96	6	記	訊	起	送	退	除	商	售	執	密	專	將	接	掛	授	敗
112	7	啟	器	清	理	處	首	通	單	報	復	提	期	款	測	無	發

Table E-18. 14N4122E—Chinese Refined Extended Font 1

Decimal	Hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	毀	簽	資	料	主	定	丟	第	般	預	先	作	某	管	整	補
16	1	登	欲	系	統	盤	式	設	拒	共	二	三	之	畢	故	障	絕
32	2	稍	等	筆	簡	結	絡	開	傳	新	滑	照	當	腦	號	試	話
48	3	詢	路	週	逾	電	對	滿	與	認	誤	銀	需	撥	數	標	確
64	4	碼	編	線	請	服	機	告	輸	辨	錯	錄	雙	檔	檢	總	聯
80	5	謝	鍵	斷	框	舊	覆	銷	額	識	槽	聽	讀	顯	喑	店	告
96	6	肌	打	那	明	細	開	保	個	或	選	擇	音	忙	軌	據	進
112	7	一	常	案	端	末	批	次	費	僅	離	調	前	保	留	妥	損

Table E-19. 14N4132E—Chinese Refined Extended Font 2

Decimal	Hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	上	匯	銀	亩	花	嘉	實	壹	貳	肆	伍	陸	柒	捌	玖	拾
16	1	海	豐	行	合	旗	利	慶	佰	仟	萬	元	作	現	欠	缺	加
32	2	百	屋	公	司	太	平	洋	崇	光	環	弧	力	霸	大	歡	立
48	3	迎	曜	臨	越	興	遠	東	領	友	高	島	下	班	加	假	工
64	4	推	門	進	進	外	出	返	井	次	群	否	繼	續	失	時	間
80	5	終	查	米	錄	字	星	西	份	分	秒	午	關	禁	紅	黃	綠
96	6	藍	白	灰	紙	張	帳	欠	產								
112	7																

Table E-20. 15N4111E—Chinese Simplfied Font

Decimal	Hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	至	集	续	连	特	约	没	参	考	资	心	脑	主	要	劲	任
16	1	意	核	指	令	后	美	止	别	只	删	增	受	国	做	停	佔
32	2	入	小	已	不	中	户	日	月	代	功	卡	可	失	必	本	未
48	3	正	用	示	交	再	列	印	回	在	存	年	成	收	有	此	行
64	4	别	即	否	完	序	改	更	束	刷	取	始	放	易	法	表	金
80	5	信	品	度	按	是	查	重	限	候	原	息	效	时	消	纳	能
96	6	记	讯	超	送	退	除	商	鲁	执	密	专	将	授	挂	授	败
112	7	启	器	清	理	处	货	通	单	报	复	提	期	款	测	无	发

Table E-21. 15N4122E—Chinese Simplfied Extended Font 1

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 0	毁	签	测	料	主	定	义	第	般	预	先	作	业	管	型	补
16 1	登	欲	系	统	盘	式	设	拒	共	二	三	之	毕	故	障	绝
32 2	稍	等	笔	简	结	络	开	传	新	滑	照	当	脑	号	试	话
48 3	词	路	过	逾	电	对	满	与	认	误	银	需	拨	数	帽	确
64 4	码	编	线	读	账	搬	告	输	辨	错	录	应	档	检	总	联
80 5	谢	键	断	桓	旧	覆	销	额	识	权	听	读	显	咭	店	告
96 6	脱	打	那	明	细	闻	保	个	或	选	择	音	忙	轨	摘	进
112 7	一	常	案	端	末	批	次	费	仅	离	调	前	保	留	妥	损

Table E-22. 15N4131E—Chinese Simplfied Extended Font 2

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 0	上	汇	银	童	花	惠	宾	登	贰	肆	伍	陆	柒	捌	玖	拾
16 1	海	丰	行	合	旗	利	庆	佰	仟	万	元	作	现	欠	缺	加
32 2	百	屋	公	司	太	平	洋	常	光	环	亚	力	霸	大	欢	立
48 3	迎	眼	临	越	兴	远	东	领	友	高	岛	下	班	加	假	工
64 4	推	门	进	准	外	出	返	井	次	群	香	继	续	失	时	间
80 5	终	查	米	录	字	星	西	份	分	秒	午	关	禁	红	黄	绿
96 6	蓝	白	灰	纸	张	帐	欠	产								
112 7																

Table E-23. 16N3112E—Thailand "B" Font

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 0	ก	ข	ฃ	ค	ฅ	ฆ	ง	จ	ฉ	ช	ซ	ฌ	ญ	ฎ	ฏ	ฐ
16 1	ฑ	ฒ	ณ	ด	ต	ถ	ท	ธ	น	บ	ป	ผ	ฝ	พ	ฟ	ภ
32 2	ม	ย	ร	ฤ	ล	ฦ	ว	ศ	ษ	ส	ห	ฬ	อ	ฮ	ฯ	ะ
48 3	ั	า	ำ	ิ	ี	ึ	ื	ุ	ู	ฺ	฿	เ	แ	โ	ใ	ไ
64 4	ๅ	ๆ	็	่	้	๊	๋	์	ํ	๎	๏	๐	๑	๒	๓	๔
80 5	๕	๖	๗	๘	๙	๚	๛	๜	๝	๞	๟	๠	๡	๢	๣	๤
96 6	๥	๦	๧	๨	๩	๪	๫	๬	๭	๮	๯	๰	๱	๲	๳	๴
112 7	๵	๶	๷	๸	๹	๺	๻	๼	๽	๾	๿	฀	ก	ข	ฃ	ค

This font supersedes the 16N3111E Thailand "A" font.

Table E-24. 16N3122E—Thailand "B" Extended Font

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	๑	๑	๒	๓	๔	๕	๖	๗	๘	๙	๐	๑	๒	๓	๔	๕
16	1	๘	๙	-	๒	๓	๔	๕	๖	๗	๘	๙	๐	๑	๒	๓	๔
32	2		๑	๒	๓	๔	๕	๖	๗	๘	๙	๐	๑	๒	๓	๔	๕
48	3	๑	๒	๓	๔	๕	๖	๗	๘	๙	๐	๑	๒	๓	๔	๕	๖
64	4	๒	๓	๔	๕	๖	๗	๘	๙	๐	๑	๒	๓	๔	๕	๖	๗
80	5	๓	๔	๕	๖	๗	๘	๙	๐	๑	๒	๓	๔	๕	๖	๗	๘
96	6	๔	๕	๖	๗	๘	๙	๐	๑	๒	๓	๔	๕	๖	๗	๘	๙
112	7	๕	๖	๗	๘	๙	๐	๑	๒	๓	๔	๕	๖	๗	๘	๙	๐

This font supersedes the 16N3121E Thailand "A" Extended font.

Table E-25. 17N4112E—Standard Logo & Character Set

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	◢	◣	_						◪	◩	◪	◣	V◂	ər	iFc
16	1	ɔn	e													
32	2															
48	3															
64	4															
80	5															
96	6															
112	7															

Table E-26. 18N4112E—Asian-Pacific Logo & Character Set

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	◢	◣	_						◪	◩	◪	◣	V◂	ər	iFc
16	1	ɔn	e		◣	⌉	◢	◤	◪	◩	◪	◩	◣	⌉	◢	◤	◪
32	2	◪	◪	◪	◢	◢	◢	◢	◢	◢	◢	◢	◢	◢	◢	◢	◢
48	3	⊔	⊓	⊔													
64	4																
80	5																
96	6																
112	7																

Table E-27. 19N4111E—Hangul Font

Decimal	Hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	가	각	간	감	갑	값	갔	강	같	개	객	거	건	검	것	겉
16	1	겠	격	걸	경	계	고	곧	공	과	관	광	교	구	국	군	권
32	2	귀	그	글	금	급	기	길	김	까	꼭	끝	나	난	낭	내	너
48	3	널	년	노	논	농	누	느	는	능	니	다	달	담	답	당	대
64	4	더	데	도	독	등	되	된	됨	두	들	드	든	듣	등	디	때
80	5	또	라	락	란	람	래	램	략	량	려	력	련	로	록	료	류
96	6	른	름	름	리	림	마	막	만	많	말	망	매	맹	머	멋	메
112	7	면	명	모	목	무	미	민	밀	바	박	반	받	발	방	버	번

Table E-28. 19N4121E—Hangul Extended Font 1

Decimal	Hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	법	벌	보	복	봄	부	분	불	비	빠	사	산	산	삼	삽	상
16	1	새	색	생	서	석	선	설	성	세	소	속	손	송	쇄	수	숙
32	2	순	숫	스	승	시	식	신	실	십	싼	아	안	않	알	암	았
48	3	앞	애	액	야	약	어	억	얻	업	없	에	여	역	연	였	영
64	4	에	오	온	옷	와	완	외	요	용	우	운	원	월	유	육	으
80	5	은	을	음	의	이	인	일	읽	입	있	자	작	잠	장	재	저
96	6	적	전	절	점	접	정	제	조	종	좌	죄	주	준	중	즘	지
112	7	진	집	째	차	착	찬	처	첫	청	체	초	추	출	취	치	카

Table E-29. 19N4131E—Hangul Extended Font 2

Decimal	Hex	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	커	컴	코	콤	크	키	킨	타	태	택	터	테	통	투	트	특
16	1	튼	파	판	패	표	품	프	피	필	하	한	할	합	해	행	허
32	2	험	현	형	호	화	확	환	회	효	후	흐	히				
48	3																
64	4																
80	5																
96	6																
112	7																

APPENDIX

F

Portability Guidelines

The VeriFone OMNI product family of transaction systems includes 8-bit microprocessor devices, such as the OMNI 300 Series terminals and OMNI 460, and 16-bit microprocessor devices, such as the OMNI 480 and 490 Series terminals. Where possible (i.e., where features and operations are common to all terminal platforms), VeriFone has maintained programming compatibility to enable you to port application source code from one terminal platform to another.

This appendix discusses three general categories of the differences between terminals requiring special handling:

- OMNI Display Types
- Pointer Size
- Byte Order and Word Order

Also included in this appendix are several specific differences between terminals that should be taken into consideration when porting an application.

Table F-1 below identifies the main differences between OMNI family terminals.OMNI 400 series terminals

Table F-1. OMNI Terminal Differences

300 Series Terminals

Model	Display Size/Type	Pointer Size	Byte Order	Keyboard	Features
OMNI 380	1x16†/Segment	2 bytes	Z180	4x4	1200 baud modem
OMNI 385L	1x16†/Segment	2 bytes	Z180	4x4	COM3/LAN
OMNI 385	1x16†/Segment	2 bytes	Z180	4x4	COM3/ 2400 baud modem
OMNI 390	Varies‡/Pixel	2 bytes	Z180	4x4+8	1200 baud modem
OMNI 395L	Varies‡/Pixel	2 bytes	Z180	4x4+8	COM3/LAN
OMNI 395	Varies‡/Pixel	2 bytes	Z180	4x4+8	COM3/ 2400 baud modem
OMNI 460	1x16†/Segment	2 bytes	Z180	4x4	Internal printer/ 2400 baud modem

† *1x16: One-line display with 16 characters*

‡ *Pixel Display Sizes: 2x18 (basic), 3x18, 4x25*

400 Series Terminals

OMNI 480	2x20†/Segment	4 bytes	68000	4x4+4	Multi-tasking/ Memory card/LAN/ 1200 baud modem
OMNI 490	2x20†/ Dot matrix	4 bytes	68000	4x4+8	Multi-tasking/ Internal PINpad/ IBM 4683/ Memory card/LAN/ 2400 baud modem

† *2x20: Two line display with 20 characters per line*

Tips for Handling Display Source Code

Display dimensions and the ability to display graphic characters or specialized graphics vary between OMNI terminals. These differences must be taken into consideration when porting an application.

To port an application to a terminal using a different display dimension:

☞ *grep the source code for* `printf()` *and* `write()` *calls, and also for display functions implemented in a support library such as the Application Construction Library's* `display()` *and* `display_at()` *routines. Check for* `#includes` *to the header file <aclconio.h> to determine if ACL display functions are implemented.*

☞ *When examining display calls, make sure that the string to be displayed will fit the new platform's display dimensions. Even if the string fits in the display, display scrolling may alter the message.*

☞ *Send a form feed (*`\f`*) character to line up the display at the upper-left character position. The* `clrscr()` *command will also do this.*

❖ *Note: If a reverse video font is used on a pixel-display,* `clrscr()` *will create a darkened space in reverse video rather than simply blank out the screen.*

Character Sets

The segment displays implement a subset of the ASCII character set including the uppercase alphabet (A-Z), numeric characters and a space. Periods and commas are included in with their preceding character unless they are the leftmost character in the string.

Dot matrix cell and pixel displays support uppercase and lower-case characters. Unlike segment-type displays, on these displays each period and comma takes up one character position.

When porting an application from a dot matrix or pixel display to a segment display, you will lose the ability to print lower-case letters. When porting an application from a segment display to a pixel or dot matrix display, any strings using periods or commas may need to be adjusted to fit the display space.

Table F-2 identifies the differences between OMNI terminal displays.

Table F-2. OMNI Terminal Display Differences

	Model	Characters per Line	Lines per Display	Display Type
300 Series	OMNI 380	16	1	Segment
	OMNI 385	16	1	Segment
	OMNI 390	18	2	Pixel
		18	3	
		25	4	
	OMNI 395	18	2	Pixel
		18	3	
		25	4	
	OMNI 460	16	1	Segment
400 Series	OMNI 480	20	2	Segment
	OMNI 490	20	2	5x7† dot matrix cell

† *5x7 dot matrix cell: 5 columns, 7 rows*

Display Functions

While most display functions are available to all terminals, pixel and dot matrix displays have additional functions not supported on segment displays. Table F-3 lists display-related functions available only to the OMNI 39X pixel display and the OMNI 49X dot matrix display:

Table F-3: Terminal-specific Display Functions

OMNI 39X	`getcontrast()`, `getgridid()`, `putpixelcol()`, `setcontrast()`, `setfont()`, `getfont()`, `getfontinfo()`, `resetdisplay()`
OMNI 49X	`blink()`, `delete_font()`, `load_font()`, `getfont()`, `getfontinfo()`, `resetdisplay()`

☞ *Both OMNI 39X and OMNI 49X display functions are defined in header files named <dsp_90.h>. Although they share the same name, these files are different for 300 Series and 400 Series platforms and are placed in separate library subdirectories during TXO Workbench installation. Checking the source code for an #include to <dsp_90.h> will indicate the probability that these display functions have been implemented.*

Pointer Size

Because OMNI 400 Series terminals support a larger memory space and application size than OMNI 300 Series terminals, the EM interpreter on OMNI 400 Series terminals has been implemented with a larger pointer size. On OMNI 300 Series terminals, a pointer is the same size as an `int` or `unsigned` (two bytes); on OMNI 400 Series terminals, a pointer is the same size as a `long` (four bytes).

When working on code for an OMNI 400 Series terminal, avoid copying pointers into integer variables, as you will lose part of the value of the pointer and probably place a bug in your program. Should this error condition occur,

the TXO compiler will identify the error with either or both of the following warning messages:

[file],[line] warning C8012:
illegal conversion of pointer to int

[file],[line] warning C8014:
conversion of pointer to int loses accuracy

Do not ignore these warning messages; locate the code causing the message and correct it by casting pointers to type `long`.

Other potential problems associated with pointers are:

- Using 0 for a null pointer value, instead of using the `#define` for the NULL pointer value. `NULL` is defined in `<stdlib.h>`, which should be included in most programs. (`NULL` is also defined in `<stdio.h>`)
- Forgetting to declare the return type of a function which returns a pointer. (The default in C is for functions to return integers, so the function will convert the pointer to an integer, losing part of its value.)
- Forgetting to declare a prototype for a function which returns a pointer, in the scope in which the function is called. (Again, the default is for the calling routine to assume the function returns an integer, and convert it to a pointer, losing part of its value.)
- Performing logical/bit operations on pointers using integer constants, instead of long or unsigned long constants. (Addition and subtraction on pointers will always work properly, as the integer will be converted to the right offset for the pointer type.)

Generally speaking, the best rule is to include the standard header files such as `<stdlib.h>` to get a good definition of the NULL pointer and of system functions you will be using, to declare the correct return type for application-defined functions, and to create and include header files with proper prototypes for all of your application-defined functions. Avoid assigning pointers to integer variables or performing bit-oriented operations on them; if you must manipulate them as integers, cast them to type `long` on OMNI 400 Series terminals. Do not ignore

compiler warnings unless you fully understand the reasons for them. Following these rules will lead to code which is portable between the OMNI 300 Series terminals and OMNI 400 Series terminals, or to virtually any other C implementation.

Byte Order and Word Order

On OMNI 300 Series terminals, the bytes in a word or integer are stored in the native order for the Z80 processor: low-order byte followed by high-order byte. On the OMNI 480 and OMNI 490 terminals, the bytes in a word or integer are stored in the native order for the 68000 processor: high-order byte followed by low-order byte. Likewise, the order of the two words within a `long` variable will be different.

In general, the byte order should be transparent to most applications, but there are a few contexts in which the difference may appear:

- In a union where a character array is overlaid on an integer type; for instance in the following declaration:

  ```
  union confused { char c[2]; int i } x;
  ```

  `c[0]` will correspond to a different byte of the integer `i`, depending on whether you execute the code on an OMNI 300 Series terminal or an OMNI 400 Series terminal.

- When a pointer to an array of some integer type is converted to a character pointer, or vice versa. For instance, in the following declaration and assignment:

  ```
  char *cp = "Hello, world.";
  int *i = cp;
  ```

  the expression `*i` (the value pointed to by `i`) will have a different value on OMNI 300 Series terminals and OMNI 400 Series terminals. This different value will be relevant if the application performs CRC, LRC, or other checksum calculations.

☞ *The best rule here is to access characters as characters, integers as integers, and perform explicit conversions when trying to combine multiple characters into a single integer value or split an integer into several characters. For instance, convert two characters to an integer using the expression:*

```
i = ( c[0] << 8 ) + c[1];
```

This will have the same result for given values of `c[0]` *and* `c[1]` *on OMNI 300 Series terminals or OMNI 400 Series terminals, or virtually any C implementation.*

Additional Portability Considerations

This section identifies additional OMNI terminal differences that should be checked and adjusted accordingly. These topics include:

- Multi-tasking
- Memory Utilization
- Implicit and Explicit Opens
- Clock Ticks
- Exception Handling And Trap Numbers
- Clear Key Trapping during SVC_KEY Functions
- LAN Differences
- Modem ioctl() Function Difference
- SVC_VERSION()
- SVC_WAIT()
- Keyboard Functionality
- Barcode Reader
- Beeper

Multi-tasking

OMNI 300 Series terminal cannot support the multi-tasking features found on OMNI 400 Series terminals. Multi-tasking functions and conventions related to OMNI 400 Series terminals, such as `lock()` and `unlock()`, piping and expanded clock functionality cannot be implemented on OMNI 300 Series terminals.

Memory Utilization

OMNI 400 Series terminals allow application code, data, stack and heap to occupy as much RAM as needed. The only limitation is the amount of available RAM space. The application programmer can set the size of the heap and stack using the TXO download converter EM_CNV24; the default is 1000 bytes for the heap (set by EM_CNV24) and 5000 bytes for the stack (set by the interpreter in the terminal if no value is selected).

The OMNI 300 Series terminals, however, have the following memory limitations:

Application Data/Downloadable drivers..... 2-14k

Buffer Pool..... 1-8k

Implicit and Explicit Opens

Some OMNI 400 Series terminal devices, such as the clock device, require an explicit open (using `open()`) before the device can be accessed. OMNI 300 Series terminals may regard these devices, for example the clock, as always open (implicit open). TXO portability guidelines recommend you explicitly open all devices.

Clock Ticks and OMNI 400 Series Terminals

On OMNI 400 Series terminals, the system tick value is incremented 100 times per second. The tick value is incremented approximately 64 times per second on OMNI 300 Series terminals.

The value `TICKS_PER_SEC` defined in the `<io.h>` file contains the appropriate value for the given system; when it is necessary to convert ticks to seconds, the number of ticks can be divided by the `TICKS_PER_SEC` value.

On OMNI 400 Series terminals, the clock tick trap is signaled 50 times a second, rather than approximately 64 times a second as on OMNI 300 Series terminals.

The value `TRAPS_PER_SEC` defined in the `<io.h>` file contains the appropriate value for the given system.

Exception Handling and Trap Numbers

On OMNI 400 Series terminals, the clock device (`"/dev/clock"`) is not "pre-opened" as it is on OMNI 300 Series terminals. In order to activate the one-second clock tick and the 10 millisecond system tick, the clock device must be specifically opened, and must remain open.

If the clock is not currently open, no traps will be signalled for the clock-tick events, even if guards are armed and activated for them.

In addition, the system tick on OMNI 400 Series terminals occurs 100 times per second, and the clock tick trap occurs 50 times per second, rather than 64 times per second as on OMNI 300 Series terminals. Use `TICKS_PER_SEC` and `TRAPS_PER_SEC` defined in <io.h> to help you write code that is not platform dependent.

☞ *For portability, when setting a guard for a device event or exception, use the* `trap_of()` *function to obtain the trap number for the device.*

Clear Key Trapping during SVC_KEY Functions

If [CLEAR] key presses are being trapped (the function `arm_guard()` has been called to monitor [CLEAR] key presses) and the [CLEAR] key is pressed during `SVC_KEY_TXT()` or `SVC_KEY_NUM()` operation, the OMNI 300 Series terminal will:

- trap the [CLEAR] key press,
- call an interrupt handling function from within the `SVC` function,
- return with a -1 return code.

Programmers must mask out the [CLEAR] key trap prior to calling `SVC_KEY_TXT()` and `SVC_KEY_NUM()` if the interrupt handling from within these functions is not desired. This is normally the case, because the `SVC` function itself checks for [CLEAR] key entries and returns a -1 if pressed.

OMNI 400 Series terminals mask out the [CLEAR] key trap for you. Therefore, neither of the `SVC` functions will call an external routine to handle a [CLEAR] key press. The [CLEAR] key press goes into the keyboard buffer, and -1 is returned by the `SVC` function.

LAN Differences

OMNI 400 Series LAN terminals implement several LAN-related features, library routines and protocols not available to an OMNI 300 Series LAN terminal. These include:

- 400 Series 'Auto Install' feature
- 400 Series 'LOOK' support
- LAN `close()` routine
- 400 Series LAN System Mode
- 400 Series LAN buffers
- Additional LAN protocols supported by 400 Series LAN terminals
- 400 Series LAN Internal modem

400 Series 'Auto Install' Support

The 3X5 LAN terminal does not support 'Auto Install' functionality as provided in the 480 LAN terminal. Through 'Auto Install', a fresh terminal (address 0) can be assigned a vacant LAN address if instructed to do so by the user from system mode.

On 3X5 terminals, the terminal address must always be either specified by the application (using the `put_env()` library call) or already present in the CONFIG.SYS file.

☞ *To support Auto Install, the OMNI 480 always keeps its LAN active, even during system mode operations. The OMNI 3X5 terminal closes the LAN when in system mode.*

400 Series 'LOOK' Support

The 400 Series LAN 'LOOK' facility provides the user with a list of active terminals on the LAN while in system mode. The 3X5 LAN terminal does not support this feature.

LAN close() call

3X5 LAN firmware will physically disable the LAN port hardware on a `close()` command. An application on the OMNI 480 LAN terminal will not have access to the LAN after a `close()`, although the firmware will still keep the LAN open.

LAN System Mode Menuing System

The menuing system implemented on the 480 LAN terminal to support LAN features in system mode has not been implemented on 3X5 LAN terminals.

400 Series LAN Buffers

The 3X5 terminal architecture uses fixed length buffers; the number of buffers is settable only before application startup. Furthermore, the 3X5 buffer pool is drawn on for all I/O related functions, not just those of the LAN. The OMNI 480, on the other hand, can configure the size and count of buffers dynamically with the `Opn_Blk` structure and all such buffers are exclusively for LAN related use.

☞ *Because of the limited buffer pool available to the 300 Series LAN terminal, we do not recommended using the 300 Series LAN terminal as a LAN gateway.*

In the LAN_TERM_PARMS structure, the OMNI 480 terminal uses two additional parameters:

buffer_size and
buffer_number.

400 Series LAN protocols

In addition to support for the OMNI 300 Series Peer-to-Peer LAN protocol, the OMNI 400 Series LAN terminals support the VFI LAN (OMNI 480 LAN) and the PC/Pinstripe LAN protocols.

400 Series Internal Modem

OMNI 400 Series terminals include an internal (COM1) modem not available on OMNI 300 Series terminals.

Modem ioctl() Function Differences

The Opn_Blk structure definition for the OMNI 480/490 terminal can be found in the <device.h> include file.

The modem device (/dev/com4) does not support the GetCtrl|3 and SetCtrl|3 function types, and these sub-functions are not intended to be accessed through ioctl() on the OMNI 400 Series terminal. The full functionality of these commands is available through the read_cmd() and write_cmd() functions.

SVC_VERSION() Return Value

The SVC_VERSION() function returns the firmware version as a counted string in the following formats:

- V8 – OMNI 300 Series terminals
- VC – OMNI 400 Series terminals

SVC_WAIT()

On 400 Series terminals (multi-tasking), this function forces a task switch when its parameter is zero.

Keyboard Functionality

Although the OMNI family of terminals share a common core of 4x4 keys, several terminals include more keys. When porting an application, be sure that all keys in the application are supported by the target terminal. Modification may be necessary if the target terminal cannot support the key presses in the source application. Prompts may also require modification when screen addressable keys are used in the source application.

Table F-4. OMNI Terminal Keyboards

Model	Keys supported
OMNI 380	4x4
OMNI 385	4x4
OMNI 390	4x4 + 8 programmable keys
OMNI 395	4x4 + 8 programmable keys
OMNI 460	4x4
OMNI 480	4x4 + 4 programmable keys
OMNI 490	4x4 + 8 programmable keys

Barcode Reader

The barcode reader (`/dev/bar`) is currently not supported on the OMNI 400 Series terminals.

Beeper

The `ioctl()` commands required for variable tones and durations on the beeper device are not supported on 400 Series terminals. Use the `sound()` function to create portable beeper tones and durations.

I N D E X

A

B

C

D