FlagShip

Object Oriented Database Development System

Cross-Compatible to Unix, Linux and MS-Windows

MULTISOFT

Release 8.1

Section APP

The whole FlagShip 8 manual consist of following sections:

Section	Content
GEN	General information: License agreement & warranty, installation and de-installation, registration and support
LNG	FlagShip language: Specification, database, files, language elements, multiuser, multitasking, FlagShip extensions and differences
FSC	Compiler & Tools: Compiling, linking, libraries, make, run-time requirements, debugging, tools and utilities
CMD	Commands and statements: Alphabetical reference of FlagShip commands, declarators and statements
FUN	Standard functions: Alphabetical reference of FlagShip functions
OBJ	Objects and classes: Standard classes for Get, Tbrowse, Error, Application, GUI, as well as other standard classes
RDD	Replaceable Database Drivers
EXT	C-API: FlagShip connection to the C language, Extend C System, Inline C programs, Open C API, Modifying the intermediate C code
FS2	Alphabetical reference of FS2 Toolbox functions
QRF	Quick reference: Overview of commands, functions and environment
PRE	Preprocessor, includes, directives
SYS	System info, porting: System differences to DOS, porting hints, data transfer, terminals and mapping, distributable files
REL	Release notes: Operating system dependent information, predefined terminals
APP	Appendix: Inkey values, control keys, ASCII-ISO table, error codes, dBase and FoxPro notes, forms
IDX	Index of all sections
fsman	The on-line manual “f**sman**” contains all above sections, search function, and additionally last changes and extensions

multisoft Datentechnik, Germany

Object Oriented Database Development System,
Cross-Compatible to Unix, Linux and MS-Windows

Section APP

Manual release: 8.1

For the current program release see your Activation Card,
or check on-line by issuing *FlagShip -version*

Note: the on-line manual is updated more frequently.

Copyright

Made in Germany. Printed in Germany.

Trademarks

Headquarter Address

multisoft Datentechnik
Schönaustr. 7
84036 Landshut
Germany

Phone: *(*+49) 0871-3300237

E-mail: support@flagship.de
support@multisoft.de
sales@multisoft.de

Web: http://www.fship.com

APP: Appendix

INKEY and LASTKEY Return Codes

Function Key	Numer.Code	terminfo	Compatibily
F1	28	kf1	Clipper F1
F2	-1	kf2	Clipper F2
F3	-2	kf3	Clipper F3
F4	-3	kf4	Clipper F4
F5	-4	kf5	Clipper F5
F6	-5	kf6	Clipper F6
F7	-6	kf7	Clipper F7
F8	-7	kf8	Clipper F8
F9	-8	kf9	Clipper F9
F10	-9	kf10	Clipper F10
F11	-40	kf11	FlagShip only
F12	-41	kf12	FlagShip only
shift-F1	-10	kf13	Clipper F11
shift-F2	-11	kf14	Clipper F12
shift-F3 to F10	-12 to -19	kf15-22	Clipper F13..20
shift-F11	-42	kf23	FlagShip only
shift-F12	-43	kf24	FlagShip only
ctrl-F1	-20	kf25	Clipper F21
ctrl-F2 to ctrl-F10	-21 to -29	kf26-34	Clipper F22..30
ctrl-F11	-44	kf35	FlagShip only
ctrl-F12	-45	kf36	FlagShip only
alt-F1 (or shift-ctrl-Fx)	-30	kf37	Clipper F31
alt-F2 to alt-F10	-31 to -39	kf38-46	Clipper F32..40
alt-F11	-46	kf47	FlagShip only
alt-F12	-47	kf48	FlagShip only
ctrl-D Cursor right	4	kcuf1	same as Clipper
ctrl-S Cursor left	19	kcub1	same as Clipper
ctrl-E Cursor up	5	kcuu1	same as Clipper
ctrl-X Cursor down	24	kcud1	same as Clipper
ctrl-B ctrl-Cursor right	2	kctab	same as Clipper
ctrl-Z ctrl-Cursor left	26	khts	same as Clipper
ctrl-V Insert (ins)	22	kich1	same as Clipper
ctrl-G Delete (del)	7	kdch1	same as Clipper
ctrl-A Home (home)	1	khome	same as Clipper
ctrl-F End (end)	6	kend	same as Clipper
ctrl-R Page-up (PgUp)	18	kpp	same as Clipper
ctrl-C Page-dwn (PgDn)	3	knp	same as Clipper
ctrl-] ctrl-Home (^home)	29	ked	same as Clipper
ctrl-W ctrl-End (^end)	23	kel	same as Clipper
ctrl-- ctrl-Page-up (^PgUp)	31	kri	same as Clipper
ctrl-^ ctrl-Page-dwn (^PgDn	) 30	kind	same as Clipper
ctrl-A...ctrl-^	1...30		see ASCII

Control Keys for READ

Key (note)		lastkey()	terminfo	Action
ctrl-S**	Cursor left	19	kcub1	character left
ctrl-D	Cursor right	4	kcuf1	character right
ctrl-Z	ctrl-Cursor left	26	khts	word left
ctrl-B	ctrl-Cursor right	2	kctab	word right
ctrl-E	Cursor up	5	kcuu1	previous GET
ctrl-X	Cursor down	24	kcud1	next GET
ctrl-M	Return, Enter	13	kent	next GET
ctrl-A	Home (home)	1	khome	first charact. in GET
ctrl-F	End (end)	6	kend	last character in GET
ctrl-]	ctrl-Home (^home)	29	ked	first GET / READ
ctrl-W*	ctrl-End (^end)	23	kel	last GET / READ
ctrl-V	Insert (ins)	22	kich1	insert on/off
ctrl-G	Delete (del)	7	kdch1	delete act.character
ctrl-H	Backspace (<-)	8	kbs	delete previous char
ctrl-T		20		delete word right
ctrl-Y		25		delete GET from cursor
ctrl-U		21		restore orig. value
ctrl-R	Page-up (PgUp)	18	kpp	end of READ
ctrl-C	Page-dwn (PgDn)	3	knp	end of READ
	Escape (ESC)	27	kext	terminates READ

Notes:

* The marked ctrl-[key] differs from Clipper. The ctrl-keys are in FlagShip always consistently assigned to the appropriate function key, see also the INKEY() table.

** The key is often redefined by stty, see chapter 9.2

Control Keys for MEMOEDIT

Key (note)		lastkey()	terminfo	Action
ctrl-S**	Cursor left	19	kcub1	character left
ctrl-D	Cursor right	4	kcuf1	character right
ctrl-E	Cursor up	5	kcuu1	previous line
ctrl-X	Cursor down	24	kcud1	next line
ctrl-Z	ctrl-Cursor left	26	khts	word left
ctrl-B*	ctrl-Cursor right	2 *	kctab	word right
ctrl-A	Home (home)	1	khome	line begin
ctrl-F	End (end)	6	kend	line end
ctrl-]	ctrl-Home (^home)	29	ked	top of page
ctrl-L*	(instead of ctrl-End)	12*		end of page
ctrl-R	Page-up (PgUp)	18	kpp	previous page
ctrl-C	Page-dwn (PgDn)	3	knp	next page
ctrl--	ctrl-Page-up (^PgUp)	31	kri	begin of text
ctrl-^	ctrl-Page-dwn (^PgDn)	30	kind	end of text
ctrl-T		20		delete word right
ctrl-Y		25		delete line at cursor
ctrl-V	Insert (ins)	22	kich1	insert on/off
ctrl-U*		21*		reformat text
ctrl-W	ctrl-End (^end) *	23	kel	end of memoedit
	Escape (ESC)	27	kext	terminate memoedit

UDF-status	Description
0	idle, all keys are processed
1	invalid key depressed, text unchanged
2	invalid key depressed, text altered
3	init status

UDF-return	Description
0	continue Memoedit, process actual key
1..31	process key 1 to 31. The key (2) is in FlagShip interpreted as 21 (reformat), for compatibility to Clipper
32	ignore this key
33 *	ignore this key (Clipper: save control key as character value in text)
34	word wrap on/off
35	scroll on/off
100	perform function key (2) word right (ctrl-Cursor right, ^—>)
101	perform function key "text end", same as (12).

Notes: * and ** Differences to Clipper, see note in READ table

Control Keys for DBEDIT

Key (note)		lastkey()	terminfo	Action
ctrl-S**	Cursor left	19	kcub1	column left
ctrl-D*	Cursor right	4	kcuf1	column right
ctrl-E*	Cursor up	5	kcuu1	previous row
ctrl-X*	Cursor down	24	kcud1	next row
ctrl-Z*	ctrl-Cursor left	26	khts	scroll left
ctrl-B*	ctrl-Cursor right	2	kctab	scroll right
ctrl-A*	Home (home)	1	khome	first field
ctrl-F*	End (end)	6	kend	last field
ctrl-]*	ctrl-Home (^home)	29	ked	previous field
ctrl-W*	ctrl-End (^end)	23	kel	next field
ctrl-R*	Page-up (PgUp)	18	kpp	previous page
ctrl-C*	Page-dwn (PgDn)	3	knp	next page
ctrl--*	ctrl-Page-up (^PgUp)	31	kri	first record
ctrl-^*	ctrl-Page-dwn (^PgDn)	30	kind	last record
ctrl-M*	Return, Enter	13	kent	end of Dbedit
	Escape (ESC)	27	kext	abort Dbedit

UDF-status	Description
0	idle, all keys are processed
1	cursor up depressed, first record reached (bof)
2	cursor down depressed, last record reached (eof)
3	database is empty
4	undefined key or Esc, Enter

UDF-return	Description
0	exit from DBEDIT
1	continue DBEDIT
2	continue DBEDIT, re-read data
3	append a record

Notes:

* The marked ctrl-[key] differs from Clipper. The ctrl-keys are in FlagShip always consistently assigned to the appropriate function key, see also the INKEY() table.

** The key is often redefined by stty, see chapter 9.2

Control Keys for ACHOICE

Key	(note)		lastkey()	terminfo	Action
ctrl-E		Cursor up	5	kcuu1	one item up
ctrl-X		Cursor down	24	kcud1	one item down
ctrl-R		Page-up (PgUp)	18	kpp	previous page
ctrl-C		Page-dwn (PgDn)	3	knp	next page
ctrl-A	*	Home (home)	1	khome	first item
ctrl-F	*	End (end)	6	kend	last item
ctrl-]		ctrl-Home (^home)	29	ked	first item in window
ctrl-W		ctrl-End (^end)	23	kel	last item in window
ctrl--		ctrl-Page-up (^PgUp)	31	kri	first choice
ctrl-^		ctrl-Page-dwn (^PgDn	) 30	kind	last choice
!..A..Z		first character	asc(..)		go to item ...
ctrl-M	*	Return, Enter	13	kent	choice, returns item no
ctrl-S	*	Cursor left	19	kcub1	abort, returns 0
ctrl-D	* *	Cursor right Escape (ESC)	4 27	kcuf1 kext	abort, returns 0 abort, returns 0

UDF-status	Description
0	iddle, all keys are processed
1	cursor up pressed, first item reached
2	cursor down pressed, last item reached
3	undefined key pressed
4	display mode only, no choice

UDF- return Description 0 abort selection, return 0 1 terminate selection, return the actual item number 2 continue selection

Notes:

* the marked keys are not available to the ACHOICE() when UDF is included. Status 3 is passed to the UDF instead.

Supported Character Sets

See <FlagShip_dir>/docu/charset.pdf for the IBM-PC8/ASCII/CP437 <-> ISO-8859-1/ANSI conversion table.

For detailed description of different character sets, refer to e.g. http://en.wikipedia.org/wiki/Codepage and http://en.wikipedia.org/wiki/iso-8859

Codepage	Description	Windows GUI	Windows -io=t	Linux GUI	Linux -io=t	Example
ISO-8859-1	Western Europe	yes	-	yes	-	western.prg
ISO-8859-2	Central Europe	yes	-	yes	-	slavic.prg
ISO-8859-3	South Europe	yes	-	yes	-	
ISO-8859-4	North Europe	yes	-	yes	-	
ISO-8859-5	Cyrillic	yes	-	yes	-	
ISO-8859-6	Arabic	yes	-	yes	-	arabic.prg
ISO-8859-7	Greek	yes	-	yes	-	greek.prg
ISO-8859-8	Hebrew	yes	-	yes	-	
ISO-8859-9	Turkish	yes	-	yes	-	
ISO-8859-10	Nordic	yes	-	yes	-	
ISO-8859-11	Thai	yes	-	yes	-	
ISO-8859-13	Baltic	yes	-	yes	-	
ISO-8859-14	Celtic	yes	-	yes	-	
ISO-8859-15	8859-1 with Euro	yes	-	yes	-	western.prg
ISO-8859-16	South/EastEurope	yes	-	yes	-	
CP437	IBM-PC/West.Eur.	-	yes	-	yes	western.prg
CP438*	IBM-PC with Euro	-	-	-	yes	western.prg
CP720	Arabic	-	yes	-	yes	arabic.prg
CP737	Greek	-	yes	-	yes	greek.prg
CP775	Baltic	-	yes	-	yes	
CP850	Western Europe	-	yes	-	yes	western.prg
CP852	Slavic/Centr.Eur	. -	yes	-	yes	slavic.prg
CP855	Cyrilic	-	yes	-	yes	
CP857	Turkish	-	yes	-	yes	
CP858	CP850 with Euro	-	yes	-	yes	western.prg
CP860	Poruguese	-	yes	-	yes	
CP861	Icelandic	-	yes	-	yes	
CP862	Hebrew	-	yes	-	yes	
CP863	French Canada	-	yes	-	yes	
CP864	Arabic	-	yes	-	yes	arabic.prg
CP865	Danish/Norwey	-	yes	-	yes	
CP866	Cyrillic	-	yes	-	yes	
CP869	Greek	-	yes	-	yes	
CP874	Thai	-	yes	-	yes	

PC8 (ASCII) ↔ ISO-8859-1 (ANSI) Table

deci	hex	ASCI	ISO	ctrl	deci	hex	ASCII	ISO	deci	hex	ASCII	ISO	deci	hex	ASCII	ISO
0	0x00		NUL	^@	32	0x20		space	64	0x40	@	@	96	0x60	`	`
1	0x01	☺	SOH	^A	33	0x21	!	!	65	0x41	A	A	97	0x61	a	a
2	0x02	☻	STX	^B	34	0x22	"	"	66	0x42	B	B	98	0x62	b	b
3	0x03	♥	ETX	^C	35	0x23	#	#	67	0x43	C	C	99	0x63	c	c
4	0x04	♦	EOT	^D	36	0x24	$	$	68	0x44	D	D	100	0x64	d	d
5	0x05	♣	ENQ	^E	37	0x25	%	%	69	0x45	E	E	101	0x65	e	e
6	0x06	♠	ACK	^F	38	0x26	&	&	70	0x46	F	F	102	0x66	f	f
7	0x07	•	BEL	^G	39	0x27	'	'	71	0x47	G	G	103	0x67	g	g
8	0x08	◘	BS	^H	40	0x28	(	(	72	0x48	H	H	104	0x68	h	h
9	0x09	○	HT	^I	41	0x29	)	)	73	0x49	I	I	105	0x69	i	i
10	0x0A	■	LF	^J	42	0x2A	*	*	74	0x4A	J	J	106	0x6A	j	j
11	0x0B	♂	VT	^K	43	0x2B	+	+	75	0x4B	K	K	107	0x6B	k	k
12	0x0C	♀	FF	^L	44	0x2C	,	,	76	0x4C	L	L	108	0x6C	l	l
13	0x0D	♪	CR	^M	45	0x2D	-	-	77	0x4D	M	M	109	0x6D	m	m
14	0x0E	♫	SO	^N	46	0x2E	.	.	78	0x4E	N	N	110	0x6E	n	n
15	0x0F	☼	SI	^O	47	0x2F	/	/	79	0x4F	O	O	111	0x6F	o	o
16	0x10	►	DLE	^P	48	0x30	0	0	80	0x50	P	P	112	0x70	p	p
17	0x11	◄	XON	^Q	49	0x31	1	1	81	0x51	Q	Q	113	0x71	q	q
18	0x12	↕	DC2	^R	50	0x32	2	2	82	0x52	R	R	114	0x72	r	r
19	0x13	‼	XOF	^S	51	0x33	3	3	83	0x53	S	S	115	0x73	s	s
20	0x14	¶	DC4	^T	52	0x34	4	4	84	0x54	T	T	116	0x74	t	t
21	0x15	§	NAK	^U	53	0x35	5	5	85	0x55	U	U	117	0x75	u	u
22	0x16	▬	SYN	^V	54	0x36	6	6	86	0x56	V	V	118	0x76	v	v
23	0x17	↨	ETB	^W	55	0x37	7	7	87	0x57	W	W	119	0x77	w	w
24	0x18	↑	CAN	^X	56	0x38	8	8	88	0x58	X	X	120	0x78	x	x
25	0x19	↓	EM	^Y	57	0x39	9	9	89	0x59	Y	Y	121	0x79	y	y
26	0x1A	→	SUB	^Z	58	0x3A	:	:	90	0x5A	Z	Z	122	0x7A	z	z
27	0x1B	←	ESC	^[	59	0x3B	;	;	91	0x5B	[	[	123	0x7B	{	{
28	0x1C	⌐	FS	^\	60	0x3C	<	<	92	0x5C	\	\	124	0x7C	|	|
29	0x1D	↔	GS	^]	61	0x3D	=	=	93	0x5D	]	]	125	0x7D	}	}
30	0x1E	▲	RS	^^	62	0x3E	>	>	94	0x5E	^	^	126	0x7E	~	~
31	0x1F	▼	US	^_	63	0x3F	?	?	95	0x5F	_	_	127	0x7F	⌂	

PC8 ASCII ↔ ISO 8859-1 Table (continued)

deci	hex	ASCII	ISO	deci	hex	ASCII	ISO	deci	hex	ASCII	ISO	deci	hex	ASCII	ISO
128	0x80	Ç	€	160	0xA0	á		192	0xC0	└	À	224	0xE0	α	à
129	0x81	ü		161	0xA1	í	¡	193	0xC1	┴	Á	225	0xE1	ß	á
130	0x82	é		162	0xA2	ó	¢	194	0xC2	┬	Â	226	0xE2	Γ	â
131	0x83	â		163	0xA3	ú	£	195	0xC3	├	Ã	227	0xE3	π	ã
132	0x84	ä		164	0xA4	ñ	¤ €	196	0xC4	─	Ä	228	0xE4	Σ	ä
133	0x85	à		165	0xA5	Ñ	¥	197	0xC5	┼	Å	229	0xE5	σ	å
134	0x86	å		166	0xA6	ª	¦	198	0xC6	╞	Æ	230	0xE6	µ	æ
135	0x87	ç		167	0xA7	º	§	199	0xC7	╟	Ç	231	0xE7	τ	ç
136	0x88	ê		168	0xA8	¿	¨	200	0xC8	╚	È	232	0xE8	Φ	è
137	0x89	ë		169	0xA9	⌐	©	201	0xC9	╔	È	233	0xE9	Θ	é
138	0x8A	è		170	0xAA	¬	ª	202	0xCA	╩	Ê	234	0xEA	Ω	ê
139	0x8B	ï		171	0xAB	½	«	203	0xCB	╦	Ë	235	0xEB	δ	ë
140	0x8C	î		172	0xAC	¼	¬	204	0xCC	╠	Ì	236	0xEC	∞	ì
141	0x8D	ì		173	0xAD	¡	-	205	0xCD	═	Í	237	0xED	φ	í
142	0x8E	Ä		174	0xAE	«	®	206	0xCE	╬	Î	238	0xEE	ε	î
143	0x8F	Å		175	0xAF	»	¯	207	0xCF	╧	Ï	239	0xEF	∩	ï
144	0x90	É		176	0xB0	░	º	208	0xD0	╨	Ð	240	0xF0	≡	ð
145	0x91	æ		177	0xB1	▒	±	209	0xD1	╤	Ñ	241	0xF1	±	ñ
146	0x92	Æ		178	0xB2	▓	²	210	0xD2	╥	Ò	242	0xF2	≥	ò
147	0x93	ô		179	0xB3	│	³	211	0xD3	╙	Ó	243	0xF3	≤	ó
148	0x94	ö		180	0xB4	┤	´	212	0xD4	╘	Ô	244	0xF4	⌠	ô
149	0x95	ò		181	0xB5	╡	µ	213	0xD5	╒	Õ	245	0xF5	⌡	õ
150	0x96	û		182	0xB6	╢	¶	214	0xD6	╓	Ö	246	0xF6	÷	ö
151	0x97	ù		183	0xB7	╖	·	215	0xD7	╫	×	247	0xF7	≈	÷
152	0x98	ÿ		184	0xB8	╕	¸	216	0xD8	╪	Ø	248	0xF8	°	ø
153	0x99	Ö		185	0xB9	╣	¹	217	0xD9	┘	Ù	249	0xF9	●	ù
154	0x9A	Ü		186	0xBA	║	º	218	0xDA	┌	Ú	250	0xFA	•	ú
155	0x9B	¢		187	0xBB	╗	»	219	0xDB	█	Û	251	0xFB	√	û
156	0x9C	£		188	0xBC	╝	¼	220	0xDC	▄	Ü	252	0xFC	ⁿ	ü
157	0x9D	¥		189	0xBD	╜	½	221	0xDD	▌	Ý	253	0xFD	²	ý
158	0x9E	₧		190	0xBE	╛	¾	222	0xDE	▐	Þ	254	0xFE	■	þ
159	0x9F	ƒ		191	0xBF	┐	¿	223	0xDF	▀	ß	255	0xFF		ÿ

Error Codes, Run-Time-Errors

For compiler and linker messages, refer to section FSC.1.8

Error codes displayed by the Run-Time-Errorsystem (RTE) where additional description appears. See also the FlagShip Error-System LNG.2.11.6, LNG.9.4.7, FUN.ERROR*() and OBJ.Error-Class, as well as files FSerrmsg.c, FSerrmsg_ger.c and FSerrors.h

Definition	Number	**M**essage / **C**ause / **F**ix
GENRAL	0	**M:**general (unspecified) error **C/F:**see the displayed error message for details
RECNOOR	1	**M:**record number out of range **C:**Access/replace on EOF(), BOF() or with record# < 0 or record# > RecCount() **F:**check RECNO() or the passed record number, debugger
IOOPEN	101	**M:**file open error **C:**General acces/open orror. File not available (lower/uppercase?), or insufficient access rights to file or directory, or exceeded quotas or system/user max files number **F:**see LNG.3.2-3, LNG.9, FILE(), #include "fspreset.fh", system docu
IOCLOS	102	**M:**file close error **C:**Insufficient disk space, exceeded quotas, network failure **F:**see system "ls", "df"
IOREAD	103	**M:**file read error **C:**File deleted by other process, disk or network failure **F:**use and check database or file locking
IOWRIT	104	**M:**file write error **C/F:**same as IOREAD = 103
IOCREAT	105	**M:**create file error **C:**Insufficient access rights to directory **F:**see LNG.3.3, system "ls -ld ."
IOERASE	106	**M:**erase file error **C/F:**same as IOCREAT = 105
IOSTAT	107	**M:**file status error **C:**access to directory denied, invalid dir name (lower/upper?), insufficient access rights to directory **F:**see LNG.3.3, DIRECTORY(), system "ls -ld ."

DTMMACH	201	**M:**data type mismatch **C:**operation on or comparison of incompatible operators **F:**check VALTYPE() for both operators, debugger FSC.5
DTILOPR	202	**M:**illegal operation on data type **C/F:**similar to DTMMACH = 201
DTILEGL	203	**M:**illegal data type **C/F:**similar to DTMMACH = 201
DTILLFL	204	**M:**illegal field type **C/F:**similar to DTMMACH = 201
DTILVAL	205	**M:**illegal field value **C/F:**similar to DTMMACH = 201
RTNOMEM	301	**M:**no memory available **C:**physical and virtual (RAM + swap) memory exhausted **F:**add RAM or at least swap space, avoid extremely large strings and arrays, use LOCAL variables if possible to automatically free unused space for variables.
RTNOVAR	302	**M:**no memory variables available **C:**access to undefined variable **F:**check for typos and var scope/visibility LNG.2.6.3, VALTYPE(), TYPE(), run with debugger FSC.5
RTNLMVP	303	**M:**null pointer for memory variable **C:**internal error or incorrect use of Open-C API **F:**if in your C program: check there. Otherwise report it to multisoft
RTXPMPT	304	**M:**too many prompts for menu **C:**should not happen in VFS, since unlimited. Clipper and previous FS versions supported 32 PROMPTs per MENU TO
RTMXGET	305	**M:**too many gets **C:**should not happen in VFS, since unlimited
RTNOFLD	306	**M:**cannot find that field **C:**invalid field name **F:**typo, wrong working area (SELECT), closed database. Check USED(), SELECT() or use debugger FSC.5
RTUNVAR	307	**M:**undefined variable **C/F:** similar to RTNOVAR = 302
RTUNFUN	308	**M:**undefined function **C:**accessing unknown or not linked-in UDF via macro or index key **F:**see CMD.EXTERNAL, check by ISFUNCTION(), debugger

RTSYNTX	309	**M:**Syntax error in expression **C:**syntax error which could not be detected earlier by compiler (e.g. in macro) **F:**check the statement for correct syntax
RTEEVAL	310	**M:**NULL pointer returned from run-time system **C/F:** similar to RTNLMVP = 303
RTARGCNT	311	**M:**wrong argument count **C:**insufficient number of mandatory parameters **F:**check the manual page for correct invocation
RTARGTYP	312	**M:**wrong argument type **C:**incorrect/unsupported type of passed parameter **F:**check the manual page for correct invocation
RTARGVAL	313	**M:**wrong argument value **C:**incorrect value of passed parameter **F:**check the manual page for correct invocation
RTENDLMEXP	314	endless loop in macro expansion **C:**the macro calls itself many times directly or indirectly **F:**check the macro, debugger FSC.5
RTTLMEXP	315	**M:**too long macro expansion **C:**exceeded buffer size for macro evaluation **F:**check the macro, result may not exceed 4 Kbytes
RTMISARG	316	**M:**missing argument **C:**insufficient number of mandatory parameters **F:**check the manual page for correct invocation
RTILLTYP	317	illegal type **C:**incorrect parameter or operator type **F:**check the statement for correct syntax
RTIVSKEL	318	**M:**illegal skeleton **C:**incorrect/unsupported NEXT/FOR/WHILE clause for this operation **F:**check the description for correct invocation
RTSTRTL	319	**M:**string too long for runtime system **C:**the string length exceeds its limit (2GB for variables, 250 bytes for file name incl. path, 4096 bytes for macro expansion). Can also happen in Extend-C API when passing strings not terminated by zero byte. **F:**check the correct string length by LEN(), debugger

RTILLVAL	320	**M:**illegal value **C:**incorrect parameter or operator value **F:**check the statement for correct syntax
RTOSERR	321	**M:**operating system command returned err **C:**invalid name by system call via RUN, or failure of system function in e.g. COPY/DELETE/RENAME FILE, directory access etc. **F:**the 'errno' system code is returned via FERROR() or DOSERROR() function
RTNODBF	322	**M:**no database opened **C:**database operation w/o open database, wrong work area **F:**check USED(), SELECT(), NETERR(), debugger FSC.5
RTFILNAM	323	**M:**wrong file name **C:**file name incorrect, e.g. empty or with illegal characters **F:**check the passed file name, debugger FSC.5
RTARGNLS	324	**M:**argument is null string **C/F:** similar to RTARGVAL = 313
RTNOLOCK	325	**M:**DBF file or record not locked **C:**auto-lock disabled and no own RLOCK()/FLOCK() **F:**avoid SET AUTOLOCK OFF for automatic locking, check ISDBFLOCK(), ISDBRLOCK(), debugger FSC.5
RTDBFVER	326	**M:**wrong DBF version **C:**unsupported non-standard xBase database version, e.g. from FoxPro, dBaseIV etc. **F:**convert the database to dBaseIII+ format by the associated RDBS or export it to text file and create the database by FlagShip using APPEND...SDF or APPEND..DELIMITED
RTNOMORLCKS	327	**M:**number of locks for the DBF exceeded **C:**unsufficient system or user environment setting **F:**see SYS.3
RTWRNGLIN	328	**M:**incorrect line in file **C:**invalid value in the key translation table **F:**check the displayed line#
RTILLCOL	329	**M:**illegal color specification **C:**invalid value in color specification **F:**see SET COLOR
RTNODBF1	330	**M:**no database opened **C/F:** similar to RTNODBF = 322

RTIDXCRPT	331	**M:**index file corrupted **C:**index key count does not match database records, or index key was not found in the database. Most possibly, the database was REPLACEd, APPENDed, ZAPed or PACKed without associated index **F:**see LNG.4.5 index integrity checking, INDEXCHECK()
RTCONVAR	332	**M:**wrong type for typed var/par **C:**assigning invalid value to typed variable, or passing parameter to an UDF not matching typed arguments **F:**check the assignment type, debugger FSC.5
RTILLPICT	333	**M:**wrong picture in say or get **C:**incorrect character in PICTURE's type or function **F:**check the PICTURE clause, see @..SAY, @..GET
RTILLEOF	334	**M:**DBF on eof or bof but record needed **C/F:** similar to RECNOOR = 1
RTILLBRK	335	**M:**Illegal break **C:**the BREAK statement was invoked but there is no corresponding BEGIN..END SEQUENCE active **F:**check by ISBEGSEQ(), debugger FSC.5
RTILLDATE	336	**M:**wrong date format in set() **C:**incorrect date format **F:**see SET DATE or SET()
RTILLNAM	337	**M:**undefined identifier **C:**undefined variable accessed by macro **F:**check the macro string and VALTYPE() of variables used there, debugger FSC.5
RTILLCLS	338	**M:**Illegal object class **C:**invalid class or object type used **F:**see the error message for details
RTUSERS	339	**M:**unable to open file **C/F:** similar to IOOPEN = 101
RTIDXVER	340	**M:**wrong IDX version **C:**index was not created by Visual FlagShip **F:**reindex via INDEX ON..TO..
RTNOFLD1	341	**M:**cannot find that field **C/F:** similar to RTNOFLD = 306
RTILLALIAS	342	**M:**Illegal Alias **C:**trying to SELECT work area by invalid ALIAS name **F:**check available aliases by DBFUSED(), debugger FSC.5

RTILLMET	343	**M:**Illegal Method **C:**the used class method is not available in the class **F:**check for correct class name by ISOBJCLASS() and if the used property is not defined as Access/Assign or Extern instance instead using ISOBJPROPERTY() or debugger
RTILLINS	344	**M:**Illegal Instance **C:**trying to access or assign an instance not available in the class, or not declared Extern instance nor as Access/Assign method **F:**check for correct class name by ISOBJCLASS() and if the used property is not defined as method or as protected instance by ISOBJPROPERTY() or debugger
RTDECLARED	345	**M:**Previously declared identifier **C:**either trying to declare CONSTANT anew, or illegal use of objects **F:**see the displayed error message
RTILLPAR	346	**M:**wrong # of parameters or their types **C:**incorrect parameter type or insufficient parameters **F:**check the function call for correct syntax
RTLOCKFAIL	347	**M:**Lock failed **C:**insufficient system resources or locking via NFS without NFS locking enabled **F:**see SYS.3, or "mount"
RTNOAPPOBJ	351	**M:**Constant oApplic not instantiated **C:**incorrect modification of initio.prg **F:**check your changes there
RTNOSTDOBJ	352	**M:**Constant _oStdIO not instantiated **C/F:** similar to RTNOAPPOBJ = 351
RTNOMEMOBJ	353	**M:**Constant _oMemoBrow not instantiated **C/F:** similar to RTNOAPPOBJ = 351
RTWIDCREA	361	**M:**Could not create GUI widget **C:**either trying to run an application in GUI mode without X11/Windows active, or incorrect modification of initio.prg, or insufficient memory **F:**see the error message for details
RTWIDAVAIL	362	**M:**GUI widget not available **C:**the accessed widget/control was already deleted or is out of the visibility scope **F:**see the error message for details

RTCLASSTYPE	371	**M:**Illegal class type **C:**trying to assign an object of incompatible class **F:**check for correct class name by ISOBJCLASS() or debugger FSC.5
RTCLASSPROP	372	**M:**Invalid class property **C:**trying to access or assign an instance or method not available in the class, or not declared Extern **F:**check for correct class name by ISOBJCLASS() and if the used property is not defined as method or as protected instance by ISOBJPROPERTY() or debugger
RTOBJNOPROP	373	**M:**Missing object property **C:**the initialization of the object is insufficient for the required operation **F:**see the class description, displayed message, debugger
INTARGCN	401	**M:**arguments count negative **C:**internal error or incorrect use of Open-C API **F:**if in your C program: check there. Otherwise report it to multisoft
INTARGCB	402	**M:**too many arguments **C:**passed more than the supported number of parameters, reported usually for code blocks **F:**check function description for details
INTARGCL	403	**M:**too little arguments **C/F:** similar to RTARGCNT = 311
INTARGTYP	404	**M:**wrong argument type **C/F:** similar to RTARGTYP = 312
INTARGNEG	405	**M:**negative argument **C/F:** similar to RTARGVAL = 313
INTARGNUL	406	**M:**argument is zero **C/F:** similar to RTARGVAL = 313
INTARGNLP	407	**M:**argument is null pointer **C:**internal error or incorrect use of Open-C API **F:**if in your C program: check there. Otherwise report it to multisoft
INTARGNLS	408	**M:**argument points to null string **C/F:** similar to RTARGVAL = 313
INTARGOR	409	**M:**argument out of range **C/F:** similar to RTARGVAL = 313
INTCMDFLG	410	**M:**flag has impossible value **C/F:** similar to RTARGVAL = 313

INTMEMOUT	411	**M:**out of memory **C/F:** similar to RTNOMEM = 301
INTNDXSEC	412	**M:**non-existing index data sector access **C:**index not created by FlagShip, or internal error **F:**report it to multisoft
INTDBFNO	413	**M:**database file (DBF) not opened **C/F:** similar to RTNODBF = 322
INTDBTNE	414	**M:**database file has no memo file (DBT) **C:**there is a "M" field in the database but the corresponding .dbt file does not exist or is not accessible **F:**check files by "ls -la"
INTNDXNO	415	**M:**index file (IDX) not opened **C:**index operation (e.g. SEEK) w/o open or with disabled controlling index **F:**check INDEXCOUNT(), INDEXORD(), INDEXDBF(), debugger
INTILLVAL	416	**M:**illegal internal value **C:**index not created by FlagShip, or internal error **F:**report it to multisoft
FTNOTOPN	501	**M:**unable to open file **C/F:** similar to IOOPEN = 101
FTFUNARG	502	**M:**too many function arguments **C:**passed more than the supported number of parameters, reported for few standard functions which do not accept/do not ignore additional parameters **F:**check the function description for details
LOCKNDX	701	**M:**index file already locked **C:**index not created by FlagShip, or internal error **F:**report it to multisoft

Differences to dBASE III+

The compatibility of FlagShip to dBASE III plus is similar to their compatibility to Clipper:

dBASE, Fox commands	FlagShip command or function
APPEND, INSERT, EDIT	APPEND BLANK
ASSIST	-
BROWSE	DBEDIT(), BROWSE()
CHANGE	REPLACE
CLEAR FIELDS	-
CREATE LABEL	LABEL EDIT
CREATE REPORT	REPORT EDIT
CREATE QUERRY, SCREEN	-
DISPLAY FILES	DIRECTORY(), FILE()
DISPLAY MEMORY	MEMORY()
DISPLAY STATUS	debugger, SET(), FS_SET()
DISPLAY	display <fields>
STRUCTURE	CREATE STRUCTURE, DBSTRUCT(), DBCREATE()
DISPLAY USERS	USERSACTIVE()
ERROR()	FERROR(), Error handle, Error object
EXPORT TO	-
HELP	Procedure HELP
IMPORT FROM	APPEND FROM
LIST FILES	FILE(), DIRECTORY()
LIST HISTORY, MEMORY	debugger
LIST STATUS, STRUCTURE	debugger, DBSTRUCT()
LOAD	CALL, #Cinline, Open C API
LOGOUT	QUIT
MESSAGE	-
MODIFY COMMAND	Standard text editor
MODIFY QUERRY, SCREEN	-
MODIFY LABEL	LABEL EDIT
MODIFY REPORT	REPORT EDIT
ON ERROR	ERRORBLOCK(), BEGIN SEQUENCE..BREAK / RECOVER
ON ESCAPE, KEY	SET KEY TO
RESUME, RETURN TO	BREAK, BEGIN SEQUENCE...END
RETRY	DO WHILE !lock() ... ENDDO
RETURN TO MASTER	BEGIN SEQUENCE ... BREAK ... END
SET	debugger
SET ...	see SET commands and SET() or FS_SET() function

Macro evaluation: In FlagShip (same as in Clipper), the macro cannot contain command, or part of commands, or command clauses. A single command value is accepted as macro. See further details in section LNG.2.10

Database and index: standard dBaseIII+ database (.dbf) and memo fields (.dbt) are fully supported and are backward compatible. Indices (.ndx and .mdx) are incompatible, you need to re-index them by INDEX ON..TO..

Source files: dBaseIII .PRG files are supported almost unchanged 1:1

Supported FoxBase and FoxPro commands and functions

FoxPro (2.5, 2.6) and Foxbase is a superset of dBase containing many xBase incompatible structures. The most of them are supported in FlagShip by using the -fox compiler switch (e.g. "FlagShip -fox myapp.prg"). This automatically includes the preprocessor file <FlagShip_dir>/include/ stdfoxpro.fh, which extends (or overrides) standard command translation by std.fh preprocessor file. Here summary of main features:

= command	supported by DO ...
and / or / not	translated to .AND. / .OR. / .NOT.
m. (prefix)	translated to m->
@...EDIT...	supported by MemoEdit()
@...MENU...	supported by Achoice()
@...MENU TITLE,SHADOW	supported by fox_AtMenu() in foxpro_api.prg
@...SAY	supported including FONT, FUNCTION, PICTURE clauses, but COLOR SCHEME, SIZE, STYLE are ignored
@...GET	supported by standard @..GET, but clauses SCHEME, FONT, STYLE, SIZE, OPEN, WINDOW, FROM are ignored
? COLOR,FONT	supported by fox_Qout() in foxpro_api.prg
?? COLOR,FONT	supported by fox_Qqout() in foxpro_api.prg
ACTIVATE MENU [PAD]	supported by TopBar class
ACTIVATE POPUP	supported by fox_PopupExec() in foxpro_api.prg
ACTIVATE WINDOW	supported by Wselect() or fox_MDIselect(), but IN WINDOW | SCREEN, BOTTOM, TOP, SAME, NOSHOW ignored
AELEMENT(..)	supported by direct array access, 3rd param unsupp
ALEN(..)	supported, 2nd parameter if given, only as 0
APPEND FROM ARRAY	supported except FIELDS clause
APPEND MEMO	supported by REPLACE
BAR()	supported by PopupBar() and PopUp class
BROWSE	supported by fox_Browse() in foxpro_browse.prg
BUILD APP	message, simply compile by FlagShip -fox
BUILD EXE	message, simply compile by FlagShip -fox

CHANGE, EDIT	partially supported via Browse(), message
CLEAR ALL	supported
CLEAR ...	clauses FIELDS, MACROS, MENUS, POPUPS, PROGRAM, WINDOWS are ignored
CLEAR PROMPT	ignored, use @..CLEAR TO...
CLEAR READ	supported by CLEAR GETS
CLOSE ALL	supported by DbCloseAll()
CLOSE PROCEDURE	ignored, since not required
COMPILE	ignored, compile by FlagShip -fox
COPY MEMO TO	supported including ADDITIVE clause
COPY TO ARRAY	supported incl. NOOPTIMIZE, except FIELDS etc. clause
COUNT	supported by DbEval() incl. TO, FOR, WHILE, NEXT, RECORD, REST, ALL, NOOPTIMIZE clauses
CREATE LABEL	supported by LABEL EDIT
CREATE REPORT	supported by REPORT EDIT
CRATE TABLE	message, use DbCreate() instead
DEACTIVATE MENU	supported by TopBar class or fox_DelMenu()
DEACTIVATE POPUP	supported
DEACTIVATE WINDOW	supported by Wclose() or fox_MDIclose()
DEFINE BAR	supported by PopUp and MenuItem class. but clauses KEY, BEFORE, AFTER, MESSAGE, MARK, COLOR ignored
DEFINE MENU	supported by TopBar and MenuItem class, all clauses except AT LINE are ignored
DEFINE PAD	supported by TopBar and MenuItem class, but clauses KEY, COLOR are ignored
DEFINE POPUP PROMPT	supported by PopUp and MenuItem class, but clauses RELATIVE, SHADOW are ignored
DEFINE POPUP FROM	supported by PopUp and MenuItem class, but only clauses TO, PROMPT FIELD, TITLE, FOOTER considered
DEFINE POPUP ...	supported by PopUp and MenuItem class, but clauses except above listed are ignored
DEFINE WINDOW	supported by Wopen() or fox_MdiOpen() but only clauses AT, SIZE, FROM, TO, TITLE are considered
DELETE	supported by DbDelete(), NOOPTIMIZE clause ignored

DELETE ...	supported by DbEval() including FOR, WHILE, NEXT, RECORD, REST, ALL, NOOPTIMIZE clauses
DIMENSION	supported by DECLARE, see below
DIR	supported, clause ON ignored
DISPLAY FILES	supported incl. LIKE clause, same as DIR
DISPLAY STRUCTURE	supported by DbStruct() and Aeval()
DO ...	supported, .PRG extension and IN clause ignored
DTOC(date,1)	supported, translated to DTOC(date)
EJECT PAGE	supported by fox_EjectPage() in foxpro_api.prg
ENDFOR	supported, same as NEXT
ENDPRINTJOB	supported, see PRINTJOB
ENDSCAN	supported, see SCAN
ERROR()	supported by fox_Error() in foxpro_api.prg
EVALUATE()	supported by &(<var>)
EXTERNAL ...	clauses LABEL, LIBRARY, MENU, PROCEDURE, REPORT, SCREEN ignored
FFLUSH()	supported by DbCommitAll()
FGETS()	supported by FreadTxt()
FILTER()	supported by DbFilter()
FLUSH	supported by DbCommitAll()
FPUTS()	supported by FWrite()
FULLPATH()	supported by TruePath()
GATHER FROM	supported by FieldPutArr() incl. MEMO clause
GATHER MEMVAR	unsupported
GO|GOTO IN	supported by dbSelectArea() and dbGoto()
GO|GOTO BOTTOM IN	supported by dbSelectArea() and dbGoBottom()
GO|GOTO TOP IN	supported by dbSelectArea() and dbGoTop()
INSMODE()	supported by Set(_SET_INSERT)
KEY()	supported by IndexKey()
KEYBOARD ...	behaves same as KEYBOARD...ADDITIVE, key labels (according to ON KEY LABEL) unsupported, PLAIN clause ignored - use INKEY() or INKEYTRAP()

LINENO()	supported by ProcLine()
LOAD ...	ignored
MEMLINES()	supported by MlCount()
MRSSAGE()	supported by fox_Message() in foxpro_api.prg
MLINE()	supported by MemoLine()
MODIFY MEMO	supported by MemoEdit(), clauses except NOEDIT ignored
MODIFY LABEL	supported by LABEL EDIT, clauses ignored
MODIFY REPORT	supported by REPORT EDIT, clauses ignored
MODIFY STRUCTURE	message, use dbu utility instead
MOVE WINDOW	supported by Wcenter() with FS2, message otherwise
NDX()	supported by IndexNames()
OCCURS()	supported by StrOccurs()
ON BAR	supported by Popup class, ACTIVATE MENU ignored
ON ERROR DO	supported by ErrorBlock()
ON ERROR ...	LABEL and other clauses raises message
ON KEY = DO	supported by OnKey()
ON KEY LABEL	unsupported, message
ON PAD	supported by PopUp and MenuItem class, but clause ACTIVATE MENU is ignored
ON PAGE	supported by fox_OnPage() in foxpro_api.prg
ON SELECTION BAR OF	supported by PopUp and MenuItem class, only clauses DO, WITH, IN, DEACTIVATE MENU are supported
ON SELECTION PAD	supported by PopUp and MenuItem class, only clauses DO, WITH, IN, DEACTIVATE MENU are supported
ON SELECTION POPUP	supported by PopUp and MenuItem class, only clauses DO, WITH are supported
PARAMETERS()	supported by Pcount()
PRINTJOB	supported by fox_Printjob() in foxpro_api.prg
PRINTSTATUS()	supported by IsPrinter()
PRIVATE	supported except ALL clause
PROGRAM()	supported by ProcName() or fox_Sys(16)
PUBLIC	supported by default

PUBLIC ARRAY	message, see below how to avoid it
READ	supported by READ and fox_Readmenu(), all clases except SAVE and MENU are ignored
RECNO(x)	supported, including optional parameter
RECCOUNT(x)	supported, including optional parameter
RECSIZE(x)	supported, including optional parameter
#REGION	ignored, see below
REGIONAL	supported by LOCAL, see below
RELATION(n,x)	supported by DbRelation(), incl. optional parameter
RELEASE LIBRARY	ignored, not required
RELEASE MODULE	ignored, not required
RELEASE WINDOW	supported by Wclose() or fox_MDIclose()
REPLACE	supported by fox_FieldPut() with WITH, NOOPTIMIZE or by DbEval() for WITH, NOOPTIMIZE, FOR, WHILE, NEXT, RECORD, REST, ALL clauses
RESTORE MACROS	ignored
RESUME	message, use Altd() or debugger
RETRY	message, use loop and NetErr()
RETURN TO	message, use BEGIN SEQUENCE ... BREAK ... END
RLOCK(x)	supported, but only one parameter
SAVE MACROS	ignored
SCAN..ENDSCAN	supported by a loop containing SKIP and filter, but better is to use DbEval() instead
SCATTER	supported by FieldGetArr() incl. MEMO clause
SCATTER MEMVAR	unsupported, use SCARER to array or FieldGetArr()
SCOLS()	supported by MaxCol()
SEEK()	supported by DbSeek()
SELECT	supported by DbSelectArea()
SELECT FROM	unsupported, message, use SQL 3rd party tools
SELECT(0), SELECT()	supported by DbSelectArea()
SELECT(1)	supported by fox_Selemax() in foxpro_api.prg

SET ...	supported by standard FlagShip SET command, but clauses AUTOSAVE, BLINK, FIELDS, NEAR, LIBRARY, STATUS BAR are ignored
SET DEBUG	ignored, use -d compiler switch
SET FOXYEAR	supported by SET EPOCH TO
SET LOCK	supported by fox_Set(_SET_AUTOLOCK)
SET PROCEDURE	supported at compile-time, see below
SET REPROCESS	unsupported, message
SET SKIP TO	supported fox_setSkip() in foxpro_api.prg
SET STEP	supported, invokes debugger when compiled with -d
SET UDFPARMS	message, use parameter calling convention
SHOW GET	supported by fox_Showget() in foxpro_api.prg
SHOW GETS	message, use @..GET instead
SHOW MENU	supported by TopBar class
SHOW OBJECT	message, use object:show() instead
SHOW POPUP	message, use ACTIVATE POPUP or InfoBox() instead
SHOW WINDOW	message, use ACTIVATE WINDOW instead
SKIP ... IN ...	supported by DbSkip()
SROWS()	supported by MaxRow()
SUSPEND	supported by Altd() invokes debugger (compile with -d)
SYS()	several SYS() functions are supported by fox_Sys() available in foxpro_api.prg, e.g.: SYS(0, 1, 2, 3, 5, 6, 10, 11, 12, 13, 14, 16, 19, 21, 22, 23, 30, 100, 102, 103, 1001, 2001, 2002, 2003, 2004, 2010, 2011)
UNLOCK IN ...	supported
USE	supported by DbUseArea() incl. IN, VIA, ALIAS, NEW, EXCLUSIVE, SHARED, NOUPDATE, READONLY, INDEX, ORDER clauses but AGAIN, ASCENDING, DESCENDING ignored
USE ?	usupported, specify file name
USE TAG	message, use USE...INDEX...
VARREAD()	supported by ReadVar()

Additional notes:

- "alias.var" syntax is converted to "alias->var" by -fox compiler switch
- the "=" command (but not assignment) is translated to "DO"
- the "and", "or" and "not" operators are translated to ".and.", ".or." and ".not." when at least one leading and trailing space is included
- .dbf and .dbt databases and memo fields are backward compatible
- .fpt memo files are supported by default with USE, and for creating new databases by SET MEMOFILE TO FPT
- .cdx (and Foxbase .idx) indices are not supported by default, you need to re-index them by FlagShip by the common INDEX ON..TO.. command
- SET PROCEDURE is in FlagShip compile-time instruction, it is not resolved at run-time as in Fox. Therefore it accepts only file name, not variables nor macros, since vars are undefined at compile time. See details in the corresponding manual page, section CMD.
- RESTORE FROM... from FoxPro is supported for C, N, D, L var types when FS_SET("memcompat", .T.) is set. Arrays and objects are not supported.
- REGIONAL variables (and #REGION nn) are not supported, you may replace them by FlagShip's LOCAL or STATIC variables (in addition to PUBLIC and PRIVATE) with a clear visibility and lifetime scope. Note: as opposite to FoxPro, FlagShip supports unlimited number of variables.
- Macro evaluation: In FlagShip (same as in Clipper), the macro cannot contain command, or part of commands, or command clauses. A single command value is accepted as macro. See further details in LNG.2.10
- Windowing like DEFINE or ACTIVATE WINDOW or MENU or POPUP etc. are emulated by FlagShip's windowing, available fully in the FS2 Tollbox. Without licensed FS2, it is emulated directly in GUI's MDI mode, i.e. the -mdi and -fox compiler switches are required.
- DEFINE POPUP...PROMPT FIELD... should be used for small databases only, it reads the current database starting at current record into memory for the Popup. For large database, use DbEdit() or TbrowseDB() instead.
- See also <FlagShip_dir>/include/stdfoxpro.fh for a list of differences, supported FoxPro commands not listed in this manual and it clauses.
- There are several functions/wrappers available in <FlagShip_dir>/system/ foxpro_api.prg. These functions are available in the FlagShip library and triggered by the -fox compiler switch. You may freely modify its behavior by re-compiling the source and adding the object in the link list.
- Since Visual FoxPro is incompatible to itself (i.e. backward to FoxPro), only the FoxBase and FoxPro 2.5, 2.6 syntax is supported by FlagShip.

Array handling: Arrays are supported by the same way as in FoxBase and FoxPro. The syntax with square brackets xyz[...] is supported by default. The alternative Foxbase and Foxpro syntax using parentheses like xyz(2,1) is not supported directly, since ambiguous (the

compiler may assume this is a call to function xyz with two arguments). However, this parenthised syntax may be translated by the FlagShip pre-processor to proper syntax:

- Array declaration DIMENSION xyz(2,3) is translated automatically to FlagShip's DECLARE xyz[2,3] by the -fox switch.
- The -fox switch will also translate xyz(2,1) to xyz[2,1] automatically, but only if the array was previously declared by DIMENSION in the same .prg source. If you use the array in other .prg(s) or UDFs, add

  ```
  #xtranslate xyz( <elem,...> ) => xyz\[ <elem> \]
  ```

  at begin of each .prg source where used, whereby the "xyz" is your array name. Declare this #xtranslate... statement for each array name used.
- You may alternatively define all array translations by the same way in one common file named e.g. "myapplic.fh", then add the statement #include "myapplic.fh" at the begin of your .prg files instead.
- Hint: to create such include file with all used arrays of your application, invoke

  ```
  grep -i declare *.prg > myapplic.fh
  grep -i dimension *.prg >> myapplic.fh
  grep -i "public array" *.prg >> myapplic.fh
  ```

  and change there array names to #xtranslate... according to above.
- In the myapplic.fh you may then also define

  ```
  #command PUBLIC ARRAY <*list*> => PUBLIC <list>
  ```

  to support any public array declared there by #xtranslate. This will also override the default warning message from stdfoxpro.fh

Source Files: Foxbase and FoxPro .PRG files are supported by the -fox compiler switch, see above. You will need to export the FoxPro project to .PRG sources and databases to dBaseIII+ compatibility. To transfer .FRX and .FRT report files to .PRG source, you may use the frx2prg utility available at ftp://fship.com/pub/multisoft/flagship/tools/frx2prg2.zip

Databases and Indices: Foxbase and FoxPro .DBF databases are used 1:1, memo .FPT files are supported as well (when the 1st byte of .dbf is 0xF5), alternatively can be exported by FoxPro to .dbt = dBaseIII+ compatibility. Foxbase and FoxPro indices are incompatible (since FlagShip supports files >> 2GB and has automatic integrity checking), you simply need to reindex them for the first time by INDEX ON..TO.. Alternatively, you may use 3rd party commercial RDD (replaceable database driver), see details in http://www.fship.com/rdds.html

Hint: compile your FoxPro sources with

```
FlagShip -fox myapp.prg other*.prg
```

or

```
FlagShip -fox -mdi -m -Mappstart *.prg
```

or

```
FlagShip -DFS2 -fox -m -Mappstart *.prg
```

and watch possible compiler messages, see also FSC.1.8. You may redirect the output to a file (e.g. err.log) by

```
FlagShip ... 2>>err.log
```

or in Linux even better by using

```
script err.log
FlagShip -q ...
FlagShip -q ...
exit  ##the script, creates err.log file
```

where all the stdin, stdout and stderr are then listed in that log file. Then consult this FlagShip manual for supported commands, it clauses and functions, as well as the stdfoxpro.fh for undocummented but supported FoxPro commands. In the most cases, the displayed errors are solved fast by modifying the corresponding command (at the displayed line number).

Selected examples and hints

Changes of .prg sources for the DOS/Unix compatibility	QRF, LNG.9.5
FlagShip compiler invocation	FSC.1.2-6
Debugging session	FSC.5
Debugger and break key, user-defined sorting, DOS to Unix	FUN.FS_SET
Customizing the application for a distribution	SYS.1.2
Determining optimal/max no of users using the same applic	SYS.3
Determining if the current execution is on Unix or DOS	LNG.9.4-5
Porting hints from/to DOS and another Unix systems	QRF, SYS.1
System tuning, number of files, resources	SYS.3 [, REL]
Language	
Argument passing from the Unix shell or Windows CMD	LNG.2.3.1
Argument passing to UDFs, usage of optional arguments	FUN.Notation
Usage of optional parameters within a UDF	FUN.PCOUNT
Ambiguous variable vs. dbf field	CMD.FIELD
The usage of aliases with a UDF	CMD.FUNCTION
Speed advantage using typed variables	CMD.LOCAL.AS
Code blocks vs. Macros	LNG.2.3.3
Exceptions and Recovery, e.g. to find a file name	CMD.BEGIN
Scanning an one or a multidimensional array	FUN.ASCAN
Sorting an one or a multidimensional array	FUN.ASORT
Database, Indices, Files	
Creating and modifying a database structure	FUN.DBCREATE
Traversing a database	CMD.DO WHILE
Textual, multiple searching in the index key	CMD.SEEK.EVAL
Report a percentage of the index processed	CMD.INDEX
SHOWINDEX() graphically displays the indexing process	FUN.DBCREATEIN
ADD_NEW_REC() reusing the deleted records instead of PACK	CMD.PACK
Usage of "filtered" index instead of SET FILTER	CMD.INDEX
Usage of SEEK instead of FILTER	CMD.SET.FILTER
Browsing a database using a "filtered" index	FUN.DBEDIT
Soft vs. hard-coded relations	CMD.SET.RELAT
MY_USE() general dbf open UDF with user prompt on error	CMD.USE
MYUSE() opens a .dbf with a user defined error handler	OBJ.ERRORNEW
MYHANDLER() error handler used for denied file access	OBJ.5
Multiuser/multitasking requirements	LNG.4.8
Multiuser/multitasking usage	FUN.RLOCK
Redisplay changed data by other user	CMD.COMMIT
Determine a Unix or DOS-alike directory	FUN.CURDIR

Determine/display files in any directory	FUN.DIRECTORY
Access denied checking	FUN.FCREATE
Check the user permission before USEing of database	FUN.FOPEN
Determining the index/dbf integrity	FUN.INDEXCHECK
Determining the status of dbf and locking	FUN.ISDBxxx

Input/Optput

Determining the size of the terminal used	FUN.MAXCOL
Nesting of @..GET/READ	CMD.READ
User modifiable READ	<FlagShip_dir>/system/getsys.prg
SAA look-like menus	CMD.@.PROMPT
The usage of vi or other editor alternatively to MEMOEDIT()	CMD.!.RUN
Usage of a Unix pipe or device for printing	CMD.SET.PRINT
Read lines from a text file	FUN.FREADTXT
General purpose functions for text editing	FUN.MEMOEDIT
File & path conversion, terminal determining and setting, zero-byte support	FUN.FS_SET
PRINTSCREEN() prints the current screen contents	EXT.4.8

Numeric, Strings, Screens

Random number generator	EXT.3.3
Sinus, cosinus and other numeric calculation	LNG.8
SINUS() demonstrates the usage of Unix default libraries	EXT.2.5
ROTATESTR() converts string left to right	LNG.8
Expanded STRPOKE() or STUFF() function	EXT._parc
Storing/restoring of screen contents in memo fields	FUN.SCREEN2CHR
DOSSCR2UNIX() converts screen contents from DOS to Unix	FUN.RESTSCREEN
Manipulation of the screen variable (contents, colors)	EXT._retscw
Determine coordinates of the SAVESCREEN() variable	EXT.VAR_SCR

Miscellaneous

SAVERESET() saves and restores all settings	FUN.SET
Measuring the overall execution time	CMD.PROCEDURE
Measuring of the elapsed time, performances	FUN.SECONDSCPU
Redisplay screen after RUN	CMD.REFRESH
Executing a Unix or Windows command in background	CMD.RUN
Start-up and program switching script similar to SWITCH.EXE	FUN.ERRORLEVEL
Determining the user name and workstation name	FUN.NETNAME
Determining the current process id number	t"), FUN.FOPEN
FUN.FS_SET("prin	
Determining the required environment setting	FUN.GETENV
Check the number of users and the FlagShip license	FUN.USERSDBF
The usage of TBROWSE	OBJ.3,4
A simple program using TBROWSE and TBCOLUMN	LNG.7
Inter-process communication, using named pipes	EXT.5

Registration Card

Please print this page using the [R] key, fill out and return to:

---

multisoft Datentechnik
Schoenaustr. 7
D-84036 Landshut

Or by Email: support@flagship.de

REGISTRATION CARD

This card entitles you to a free initial period support and updates of FlagShip. For DEMO users of FlagShip, this card entitles you for a free installation support and a free information about future enhancements and add-ons.

FlagShip
Serial Number _____(check by FlagShip -v, see FSC.1.8)

Operat.System _____

Purchased by _____

_____

Purchase Date _____

LICENSEE:

Company Name _____

Last Name _____ First Name _____

Title _____

Department _____

Address _____

Street, POB _____

City _____ State _____

Zip,PostCode _____ Country _____

Telphone No. _____ Fax _____

Email _____ Internet _____

I agree with the FlagShip license conditions and warranty as stated in the manual, section GEN.2.

Date _____ Sign _____

Support Request Form

Please use this form when contacting the tech support department by e-mail or fax. Note, that only registered users can be supported.

```
------------------------------------------------------------------------------
FlagShip SUPPORT REQUEST (and SUGGESTIONs)        My ref.no.:
------------------------------------------------------------------------------

Licensee         (Note 1, name, company) ...:
                          address .........:
e-mail address   ........................:
Purchase Date    (Note 1) ................:

FlagShip ser.No. (Note 2, e.g. abcd...98765):
FlagShip Release (Note 2, e.g. 4.32.1234)...:
License-Users    (Note 2, e.g. Demo|unlim|4):

Oper.System used (Note 3, e.g. SCO 3.2.42)..:
cc compiler rel. (Note 4, e.g. mcc 3.2.0)...:
Hardware used    (Note 3, e.g. 486DX50,16MB):
Environment used (GUI,terminal,basic) ) ....:
TERM environment (Note 5, e.g. FSansi) .....:

------------------------------------------------------------------------------
Note 1: Data according to the  Registration Card.  If not registered  yet,
        fill in the registration and attach it. The registr. form  is ava-
        ilable a) on your computer in  <FlagShip_dir>/docu/REGCARD, or  b)
        in the manual, or  c) via http://www.fship.com/support.html
Note 2: If possible, execute 'a.out -FSversion [your-params]' and press ^S
        to watch  the output  or redirect stdout + stderr to file.  If the
        executable is  not available,  forward data  from the distribution
        media (e.g. floppies) and the activation card.
Note 3: forward the full OS release, check e.g. with uname -a, and details
        about your Hardware (computer, model, RAM, terminal type and model
        emulator etc.)
Note 4: check it with e.g. cc -v or forward the date of /bin/cc
Note 5: check it with e.g. #env or #echo $TERM $TERMINFO

Our support is FREE for registered users within the initial period of SIX
months (thereafter, an inexpensive support contract is available). The
support department will QUICKLY respond, usually within 24 hours.
------------------------------------------------------------------------------

 Check your request(s):
   ( ) FlagShip installation
   ( ) FS precompiler (phase 1)
   ( ) FS compiler    (phase 2)
   ( ) cc/bcc compiler error
   ( ) linker error
   ( ) execution of the applic, input/outp, terminfo (specify TERM & HW)
   ( ) Unix system, tuning etc.
   ( ) FS language, OOP, APIs, RDDs
```

```
   ( ) FS library, std.function
   ( ) FS compatibility to ( )Clipper__, ( ) VO__, ( )Fox___, ( )dBase__
   ( ) FS manual, docu
   ( ) other

 Please describe your problem, question or suggestion:



 Did you check the FlagShip manual? ( )yes, sections (e.g.REL.2,LNG.2.3):


 Include a small, COMPILABLE and TESTED program example (up to approx.
 100 lines), if applicable:



 - Compile the example with options:


 - Execute the example with options:


 - To re-produce the problem the following command or key sequence
   is required:


 - If required, additional data (.dbf, .dbt, .mem etc.) are included in
   the file (Note 5) :
   ( ) put into the multisoft incoming server (see below) - named:
   ( ) e-mailed (binary) named:

-------------------------------------------------------------------------
Note 5: For binary files, use tar + gzip (.tgz) or tar + compress (.taz)
  or zip (.zip) format. Specify the file name and forward it (using bin
  protocol) to:

  FTP, PUT into dir: ftp:/ftp.flagship.de/3/382125_24023/incoming
  or by e-mail: binary attachment
-------------------------------------------------------------------------
Forward this by e-mail or fax to your distributor or directly to

  multisoft Datentechnik
  Schoenaustr. 7
  D-84036 Landshut

Or by Email:
  support@flagship.de
```

Index

Notes

multisoft Datentechnik
Schönaustr. 7
D-84036 Landshut

http://www.fship.com
sales@multisoft.de
support@flagship.de