FlagShip

Object Oriented Database Development System

Cross-Compatible to Unix, Linux and MS-Windows

MULTISOFT

Release 8.1

Section REL

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

multisoft Datentechnik, Germany

Copyright (c) 1992..2017
All rights reserved

FlagShip

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

Section REL

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

Copyright © 1992..2017 by multisoft Datentechnik, D-84036 Landshut, Germany. All rights reserved worldwide. Manual authors: Jan V. Balek, Ibrahim Tannir, Sven Koester

No part of this publication may be copied or distributed, transmitted, transcripted, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, manual, or otherwise; or disclosed to third parties without the express written permission of multisoft Datentechnik. Please see also "License Agreement", section GEN.2

Made in Germany. Printed in Germany.

Trademarks

FlagShip™ is trademark of multisoft Datentechnik. Other trademarks: dBASE is trademark of Borland/Ashton-Tate, Clipper of CA/Nantucket, FoxBase of Microsoft, Unix of AT&T/USL/SCO, AIX of IBM, MS-DOS and MS-Windows of Microsoft. Other products named herein may be trademarks of their respective manufacturers.

Headquarter Address

multisoft Datentechnik
Schönaustr. 7
84036 Landshut
Germany

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

Phone: (+49) 0871-3300237

Web: http://www.fship.com

REL: Release Notes

System Specific Release Notes

Please refer to the on-line manual "fsman" section REL, or the ASCII file <FlagShip_dir>/manual/relnotes.asc for specific notes regarding FlagShip for your currently used operating system.

Note: the <FlagShip_dir> is your installation directory, e.g. C:\FlagShip8 in MS-Windows or /usr/local/FlagShip8 in Linux.

FlagShip 8 release (Visual FlagShip)

General Compatibility Issues

This FlagShip 8 (Visual FlagShip VFS8) is backward **compatible** to:

- all source files (*.prg and *.c written in Extend-C and Open-C API) of all previous FlagShip releases
- all #include *.fh files except std.fh - many *fh files of FS8 includes additional options, so it is highly recommended to use the new ones
- all used binary files (.dbf, .dbt, .idx, .mem, .lbl, .frm, .txt etc)
- databases (.dbf, .dbt) are backward compatible from FS4 to VFS7
- database variable fields (.dbv) are backward compatible to VFS6 and VFS7
- indices (.idx) are backward compatible to VFS5, VFS6 and VFS7

Incompatible to FlagShip 8 (VFS8) are:

- the by the former FlagShip release **produced** .c and .o or .obj files
- the former FlagShip library (and the most of 3rd party or user libs)
- the former FlagShip patches (addendum's)
- the former FS2 Toolbox library
- indices created by FlagShip 4.x, Clipper, Foxbase, Foxpro or dBase
- 16bit object files and libraries from DOS

New in VFS8 = FlagShip rel. 8 (Visual FlagShip)

Visual FlagShip 8 allows you to create native objects and executables for the **32bit or 64bit** architecture, see details in section FSC. The main highlights in addition to VFS7 are:

- creates native 32bit or **64bit** freely* distributable .EXE file for MS-Windows and 32bit or 64bit Linux objects/executables from common .prg sources (* only executables from the Pro version are freely distributable, see the License in section GEN)
- fully backward compatible sources to previous FlagShip versions (4 to 7)
- supports fully Clipper ('87 to 5.3) source and databases as well as the most of dBaseIII+, Foxbase and FoxPro2.x
- uses dBaseIII compatible databases with automatic integrity check
- supports large files >> 2GB (up to tera or penta-bytes of size)
- featured source-code debugger for comfortable development
- hybrid mode for GUI and/or text oriented look-and-feel application, created from the same source code
- all your sources and data remains cross-compatible between Linux and MS-Windows
- integrated Web and network support
- many additional functions and features

For additional details and improvements, see the CHANGES.LOG and CHANGES.TXT files in the <FlagShip_dir> directory.

New in VFS7 = FlagShip rel. 7 (Visual FlagShip)

The main highlight in Visual FlagShip 7 is a true graphical printing and graphical output to screen and printer. All sources and data are backward compatible to VFS5 and VFS6, see above. T he manual and documentation was extended correspondingly. You may differentiate the VFS7 features and extensions by using the #ifdef FlagShip8 ... #endif preprocessor directives declared automatically. Highlights of FlagShip release 7:

- creates native 32bit, freely* distributable .EXE file for Windows and 32bit Linux objects / executables from common .prg sources (* only executables from the Pro version are freely distributable, see the License in section GEN)
- fully backward source compatible to previous FlagShip versions (4 to 6) - supports Clipper ('87 to 5.3) source and the most of dBaseIII+, Foxbase and FoxPro
- uses dBaseIII+, Clipper and FoxPro 2.x compatible databases
- has automatic integrity check - supports large files >> 2GB (up to tera or penta-bytes of size) - featured source-code debugger for comfortable development
- hybrid mode for GUI and/or text oriented look-and-feel application, created from the same source code
- all your sources and data remains cross-compatible between Linux and MS-Windews
- integrated Web and network support - many additional features
- new: images, lines, circles, ellipses, pies and polygons are drawn on desktop and/or printer by simple @..DRAW command
- new: redirection of the desktop output (preview) to selected printer
- new: supports all Windows printers, also USB and network based
- new: coordinates in row/col (default), pixel, cm, mm, inch
- new: multi-column printout - new: automatic resolution of 1:n:n relations
- new: tool for semi-automatic creation of make-files

For additional details and improvements, see the CHANGES.LOG and CHANGES.TXT files in the <FlagShip_dir> directory.

New in VFS6 = FlagShip rel. 6 (Visual FlagShip)

Compared to FlagShip 5, many commands and functions have extended functionality - but are of course fully backwards compatible. There are also many new commands, functions and classes.

The .dbf structure was extended (but backwards compatible), containing also variable and auto-compressed character and binary/blob fields. See the DbCreate() function for details.

Ports and compatibility for Linux and MS-Windows were added. You may differentiate the FS6.x features and extensions by using the #ifdef FlagShip6 ... #endif preprocessor directives, where the #define FlagShip6 is declared automatically. The #define FlagShip is set in FS6config, so with #ifdef FlagShip ... #endif you may decide between code from other xBase systems like Clipper, and FlagShip.

The manual and additional documentation was partially reworked and includes also description or hints notable for using it in MS-Windows environment.

For additional details and improvements, see the CHANGES.LOG and CHANGES.TXT files in the <FlagShip_dir> directory.

New in VFS5 = FlagShip rel 5 (Visual FlagShip)

Compared to previous release 4.48, FlagShip was heavy extended. However, the FS4.4x sources and databases remains backward compatible, so you simply re-compile previous sources with VFS5 (or later) preferably using the -io=t switch, to get look-and-feel of FS4. You only will need to re-index your databases, since the index structure changes.

The main highlights of VFS5 are:

The compiler and library handles now THREE different i/o modes:

GUI : graphical oriented i/o, requires X11 or Windows32
Terminal: text/curses oriented i/o e.g. for console or remote terminals, same behavior as FlagShip 4.48. Available for Unix & Linux only.
Basic : basic/stream i/o e.g. for Web, CGI, background processing etc. The screen oriented i/o is roughly simulated for source compatibility purposes.

The i/o mode is either set at compile-time, or determined at run-time from the currently used environment. The compile-time solution is recommended when the target environment is known, it produces faster and smaller executables.

See more in section LNG.1.2, LNG.1.3.3, LNG.5.3 and FSC. The extension of commands and functions are described in the reference part (sections CMD, FUN, OBJ and FS2) in each "Compatibility" part.

For additional details and improvements, see the CHANGES.LOG and CHANGES.TXT files in the <FlagShip_dir> directory.

Predefined Terminals

This section applies for Terminal i/o mode only

Included with the FlagShip package are also many extended terminal descriptions which support the extensions of your UNIX hardware, such as colors, the proper input/output with (emulated) IBM-PC8 character set, the usage of cursor and function keys, etc. For more information, please refer to the description in the section SYS and the machine dependent notes in section REL.

These predefined terminals are included in the files FStinfo.src, FSchrmap.def (output mapping) and partially in FSkeymap.*, where input mapping is necessary. You may modify or/and distribute these files with your application.

Your FlagShip application will automatically use the curses library (and the terminfo definition) to support virtually **any** terminal, so the usage of predefined terminals are only for your convenience to get fast and successful results.

Terminal/ terminfo entry	Description		input/output mapping	
	Ansi Terminals w. IBM-PC Character Set, System Console of SCO UNIX:			
fsansi	ANSI color	25x80	-	x
fsansi-43	ANSI color	43x80	-	x
fsansi-c	ANSI color, alternate char. set	25x80	-	x
fsansi-m	ANSI monochrome, alt. char. set	25x80	-	x
fsansi-a	no terminal			
	PC AT386 Systems, System Console of AT&T, NCR, GENERICS and some others:			
fsAT386	at386 color	25x80	-	x
fsAT386-M	at386 monochrome	25x80	-	x
fs_def_ansi	no terminal, function key defs			
fs_def_inter	no terminal, function key defs			
fs_def_color	no terminal, color definitions			
	DEC vt100/200/300 terminals with with 7-Bit control sequences, Wyse GT99 terminal: see note **n2**			
fsVT100	DEC vt100/200 monochrome	24x80	x	x
fsVT100-25	DEC vt100/200 monochrome	25x80	x	x
fsVT100-80	DEC vt100/200 monochrome	24x80	x	x
fsVT200	DEC vt100/200 monochrome	24x80	x	x
fsVT200-80	DEC vt100/200 monochrome	24x80	x	x

fsVT100-132	DEC vt100/200 monochrome	24x132	x	x
fsVT200-132	DEC vt100/200 monochrome	24x132	x	x
fsVT100-a	no terminal			
	PC-Emulations of vt100 and vt102, supporting also ANSI style color and function keys:			
fsvt100	vt100 color, IBM-PC char set	25x80	-	x
fsvt100-50	vt100 color, IBM-PC char set	50x80	-	x
fsvt100-m	vt100 mono, IBM-PC char set	25x80	-	x
fsvt100-a	no terminal			
	SUN sparc station:			
fssun	Sun console with scrollable history	x	x	
	Siemens 97801/97808 terminals: see note **n1**			
fs97801-833	Siemens 97801 8-bit monochrome	33x80	x	x
fs97801-825	Siemens 97801 8-bit monochrome	25x80	x	x
fs97801-833c	Siemens 97801 8-bit color	33x80	x	x
fs97801-825c	Siemens 97801 8-bit color	25x80	x	x
fs97801-733	Siemens 97801 7-bit monochrome	33x80	x	x
fs97801-725	Siemens 97801 7-bit monochrome	25x80	x	x
fs97801-724	Siemens 97801 7-bit monochrome	24x80	x	x
fs_def_sie???	no terminal			
	NCR 4970 terminal in vt100/vt200 mode with 7-Bit control sequences:			
fsncr4970-80	NCR 4970 monochrome	24x80	x	x
fsncr4970-132	NCR 4970 monochrome	24x132	x	x
	Bull Questar 303 or 310 terminal in vt220/vt320 with 7-Bit control sequences:			
fsbullq303-80	BULL Questar 303/310 mono	24x80	x	x
fsbullq303-132	BULL Questar 303/310 mono	24x132	x	x
	HP System Console with X Windows:			
FShpterm	HP console as hpterm	25x80	x	x
	Wyse 60 terminal with ANSI Cursor keys, see note **n3** HP 700/60 in Wyse mode or Link MC5 A/S in Wyse mode: see note **n7**			
fswy60-24	Wyse 60 monochrome	24x80	-	x
fswy60-25	Wyse 60 monochrome	25x80	-	x

fswy60-43	Wyse 60 monochrome	43x80	-	x
fswy60-a	no terminal			
	IBM terminals System Console of IBM RS6000:			
fsaixterm	IBM X terminal emulator monochr.	25x80	-	x
fsaixterm-M	IBM X terminal emulator monochr.	25x80	-	x
fsibm6153	IBM6153 terminal monochrome	25x80	-	x
fsibm5154	IBM5154 terminal monochrome	25x80	-	x
fsibm5151	IBM5151 terminal monochrome	25x80	-	x
	RELtech ANSI Terminals: see note **n5**			
fsIBMansi	TR170 color term. with automargin	25x80	x	x
General AViiON Sy	stem Console:			
FSDGxterm	Data General X terminal emulator	65x80	x	x
	C.Itoh 324 Terminal in VT300 mode: see note **n4**			
FScitoh324	vt300 monochrome	24x80	x	x
	ICL M5/M15 terminals in vt200 7-Bit mode, with PC AT Keyboard:			
fsiclM5-24	ICL M5/M15 monochrome	24x80	x	x
fsiclM5-25	ICL M5/M15 monochrome	25x80	x	x
fsicl-a	no terminal			
	DDE420 terminals in vt200 7-Bit mode:			
fsDDE420-24	fsDDE 420 monochrome	24x80	x	x
fsDDE420-25	fsDDE 420 monochrome	25x80	x	x
fsDDE420-48	fsDDE 420 monochrome	48x80	x	x
fsicl-a	no terminal			
	ICL Alfaskop P420E terminals: see note **n6**			
fsVT400-25	ICL P420E monochrome	25x80	x	x
fsVT400-48	ICL P420E monochrome	48x80	x	x
fsVT400-p25	ICL P420E mono (PC char set)	25x80	-	x
fsVT400-p48	ICL P420E mono (PC char set)	48x80	-	x
fsicl-a	no terminal			
	Univel/UnixWare xterm terminal emulator:			
fsunivel	Univel X terminal emulator	25x80	x	x

	Silicon Graphics Indigo xwsh terminal emulator:			
fsIndiXwsh	Indigo xwsh color	25x80	x	x
	DEC VT300/VT400/VT420 terminal: see note **n8**			
fsVT300	DEC VT 300/400 Terminal	24x80	x	x

You can set one of the above terminal description according for your physical (or virtual) hardware terminal (or emulation) using e.g.

```
$  TERM=fsansi;export TERM                                  for sh, ksh
% setenv TERM fsansi                                            for csh
```

Your shell and vi editor will also work properly.

NOTES

n1 Some Siemens 97801 terminals, especially Series 5xx, do not properly support the switching into 25 or 33 line mode. If any problems occurs with the terminal, use the general purpose terminal description fs97801-724.

n2 For the Wyse 99GT Terminal use the fsVT100-25 terminal definition and set up the terminal in the following way:

```
DISP->COLOUMS             80
DISP->LINES               25
DISP->CURSOR              BLINK LINE
DISP->CELLSIZE            10x10
GENERL->PERSONALITY       VT220 7 BIT
KEYBOARD->LANGUAGE        GERMAN
MISC1->KEYS               TYPEWRITER
MISC1->DEL                BS/DEL
MISC1->CHARMODE           MULTINATIONAL
MISC1->CURSORKEYS         NORMAL
```

The Wyse 99GT terminal does not support control cursorkey combinations. The keyboard does not have an escape key, please use control-# instead.

n3 The standard Wyse 60 terminal sends backspace, return and other conflicting keys when a cursor key is pressed. Therefore, ANSI escape sequences are used in the fswy60-xx definitions and must be assigned on the terminal. Please set up the following keys on your Wyse terminal (esc represents the escape character):

```
(Cursor up key)    to esc[A
(Cursor down key)  to esc[B
(Cursor left key)  to esc[C
(Cursor right key) to esc[D
```

n4 The C.Itoh Terminal Description was specially designed for a HP 9000 running under HPUX 8.00. For different machines, it may make more sense to use the VT200-80 terminal description instead. Set up the terminal in the following way:

```
Terminal Setup->
  Emulation Mode          VT300 7bit
  Character Set           Multinational
Display Setup->
  Text Display            Light on Dark
  Cursor                  On
  Cursor Style            Underline
  Cursor State            Blink
  Control Codes           Interprete
  Status Line             Off
Keyboard Setup->
  Cursor Keys             Normal
  Keyboard                Typewriter
```

n5 The fsIBMansi Terminal description was specially designed for the IBM6000 extended curses library to support ANSI color terminals with AIX. On all other machines, use fsansi instead and change it according to the automargin ability of the terminal.

n6 When using the FS400-xx terminal description with the ICL Alfaskop P420E terminal, enter the terminal setup by pressing <ALT>-<PRTSCR> and set up the terminal in the following way:

```
General->
  Terminal Mode           VT400-7bit
  PC-TERM                 Off
  Character Set Mode      8 bit Multinat.
  Pref. 8-Bit Char.set    (see below)
User->
  Normal Screen Backgnd.  Black
  Cursor Type             Blink Underl.
Keyboard->
  Keyboard Emul. Mode     ICL DRS VT Plus
  Keyboard Type           Typewriter
  Numeric Keypad          Normal
  Cursor Key Mode         Normal
  Break Function          Enabled
DefKeys->
  Backarrow Key           Backspace
  String Set              A
Screen->
  Columns per line        80
  Lines per Screen        25/48
  Lines per Page          25/48
  Display Controls        Interpret
  Auto Wrap               On
  Background              Normal
  Cursor                  Visible
  Status Display          Local
Actions->
  Save settings    will save the setup
```

You may use the 8 Bit Character Set: PC437 with the fsVT400-p25 and fsVT400-p48 Terminal description, with fsVT400-25 and fsVT400-48 choose the ISO Latin 1 character set instead. Note that it may be necessary to reset the terminal when changing the screen resolution.

n7 When using the Link Technologies MC5 A/S terminal in Wyse 60 mode with the fswy60-xx terminal description, enter the terminal setup by pressing <Shift>-<Select> and set up the terminal in the following way:

```
F1 Gen->
  Emulation               Wyse 60
  Enhancements            On
  Auto Scroll             On
  Auto Wrap               On
F3 Disp->
  Pages                   1 x Lines
  Cursor Style            Blink Line
  Cursor                  On
  Background              Dark
  Attributes              Char
  Pound Char              US
  Auto Font Load          On
F4 Kbd->
  Return Key              CR
  Enter Key               CR
  Back Space Key          BS/DEL
  Delete Key              Normal
  Key Code                ASCII
  WP Kbd Mode             Off
```

n8 When using the DEC VT300, VT320, VT400 or VT420 DEC Terminal, enter the terminal setup by pressing <F3> and set up the terminal in the following way:

```
Display->
  Cursor
  Underline Cursor Style
  No Status Display
  24 Lines/Screen
  No Auto Wrap
  VT400 Mode, 7 Bit Controls
General->
  8 Bit Characters
  No New Line
  Normal Cursor Keys
  UPSS: Dec Supplemental
Keyboard->
  Typewriter Keys
  Character Mode
```

Index

MULTISOFT

multisoft Datentechnik
Schönaustr. 7
D-84036 Landshut

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