

Atari ST System Structures

This reference card list frequently used structures and defines. Also some external variables are included. Some Flags are only valid, when MiNT, the multitasking TOS extension from Eric R. Smith is running. These are marked with "(MiNT)". Not all presented structure fields or defines are necessarily useful on the ST, but are provided for compatibility.

AES Object Type	14
Application-Block-Structure	17
Atari ST Executable Fileformat	4
Basepage	2
BIOS Parameter Block	7
Bit Blt Parameter Block Type	12
Bit-Image-Structure (BITBLK)	17
Date and Time	2
Directory Information	3
Disk Transfer Area	8
Diskinfo	8
File Information	3
File Timestamp	9
Font Header	11
Form_Dial Opcodes	19
GEM IMG File Format	20
GEM Meta File Format	20
Graphic- and VDI-Rectangle	15
Group Information	7
HDINFO Structure	9
ICON Block Type	16
Ioctl Commands and Structures	5
KBDVECS Structure	7
Keyboard States	18
Keyboardtables	8
LineA Structure	10
Lseek, Open and Fcntl Flags	4
Memory Form Definition Block Type	13
Message Event, Window Handling	17
Mouse Form Definition Block Type	13
Multi Event Flags	17
OP_TAB of Bit Blt Parameter Block	12
OS Header	10
Parameter-Block-Structure	17
Password Information	7
Resource File Header	16
Resource Structure Defs	19
Serial I/O Buffer	7
Signals	6
Sprite Definition Block Type	14
SRC and DST Description Block Type	13
TEDINFO Structure Type	16
XBRA	9

This reference card is currently based on patchlevel 73 of Jwahr R. Bammi's library and include files.

Additionally, this reference card is postcard-ware. If you find it useful, send a postcard to this address: Frank Ridderbusch, Sander Str. 17, W-4790 Paderborn, Germany.

Basepage

```
#include <basepage.h>
```

```
typedef struct basep {
    char *p_lowtpa;      /* ptr to self (bottom of TPA) */
    char *p_hitpa;      /* ptr to top of TPA + 1 */
    char *p_tbase;      /* base of text segment */
    long p_tlen;        /* len of text segment */
    char *p_dbase;      /* base of data segment */
    long p_dlen;        /* len of data segment */
    char *p_bbase;      /* base Of BSS Segment */
    long p_blen;        /* len of BSS segment */
    char *p_dta;        /* ptr to current DTA */
    struct basep *p_parent; /* ptr to parent's basepage */
    char *p_reserved;
    char *p_env;        /* ptr to environment string */
    char p_junk[8];     /* don't touch under MiNT */
    long p_undef[18];  /* scratch area... don't touch */
    char p_cmdlin[128]; /* command line image */
} BASEPAGE;
```

```
extern BASEPAGE *_base;
```

Routines concerned: **Pexec()**

```
#include <osbind.h>
```

Define	Value	Meaning
PE_LOADGO	0	load and go
PE_LOAD	3	just load
PE_GO	4	just go
PE_CBASEPAGE	5	just create basepage
PE_GO_FREE	6	just go, then free mem
	100	as 0, start async. (MiNT)
	104	as 4, start async. (MiNT)
	200	as 0, but overlay (MiNT)

Date and Time

Routines concerned: **mktime(), asctime(), gmtime(), localtime(), strftime()**

```
#include <time.h>
```

```
struct tm {
    int tm_sec;        /* seconds (0..59) */
    int tm_min;        /* minutes (0..59) */
    int tm_hour;       /* hours (0..23) */
    int tm_mday;       /* day of month (1..31) */
    int tm_mon;        /* month (0..11) */
    int tm_year;       /* year - 1900 */
    int tm_wday;       /* day of week (0=Sun..6=Sat) */
    int tm_yday;       /* day of year (0..365) */
    int tm_isdst;      /* daylight savings */
};
```

Directory Information

Routines concerned: **readdir()**

```
#include <dirent.h>
```

```
struct dirent {
    long d_ino;           /* garbage under TOS */
    off_t d_off;         /* in TOS, entry in list */
    short d_reclen;      /* in TOS, length of d_name */
    struct dirent *d_next; /* ptr to next dirent in list */
    unsigned char d_attribute;

    unsigned short d_time; /* TOS time for file */
    unsigned short d_date; /* TOS date for file */
    long d_size;          /* file size */
    char d_name[1];      /* file name */
};
```

File Information

Routines concerned: **stat()**, **fstat()**, **lstat()**

```
#include <types.h>
```

```
#include <stat.h>
```

```
struct stat {
    u_short st_mode;
    ino_t st_ino;
    dev_t st_dev;
    short st_rdev;
    short st_nlink;
    uid_t st_uid;           /* user id */
    gid_t st_gid;          /* group id */
    off_t st_size;         /* file size */
    off_t st_blksize;      /* bytes per fs block */
    off_t st_blocks;       /* no. of blocks */
    time_t st_mtime;       /* modification time */
    time_t st_atime;       /* access time */
    time_t st_ctime;       /* creation time */
    short st_attr;
};
```

Define	Value	Meaning
S_IFMT	0170000	file type mask
S_IFBLK	0060000	block special device
S_IFCHR	0020000	character special device
S_IFDIR	0040000	directory
S_IFREG	0100000	regular file
S_IFIFO	0120000	named pipe (MiNT)
S_IFLNK	0160000	symbolic link
S_ISUID	0004000	s-bit owner
S_ISGID	0002000	s-bit group
S_ISVTX	0001000	sticky bit
S_IREAD	0000400	readable by owner
S_IWRITE	0000200	writable by owner
S_IEXEC	0000100	executable by owner

Routines concerned: **utime()**

```
#include <types.h>
```

```
struct utimbuf {
    time_t axtime;        /* access time */
    time_t modtime;       /* modification time */
};
```

Atari ST Executable Fileformat

```
#include <st-out.h>
```

```
struct aexec {
    short a_magic;        /* magic number (0x601a) */
    unsigned long a_text; /* text segment size */
    unsigned long a_data; /* initialized data size */
    unsigned long a_bss;  /* uninitialized data size */
    unsigned long a_syms; /* symbol table size */
    unsigned long a_AZero1; /* always zero */
    unsigned long a_ldflgs; /* program load flags */
    unsigned short a_isreloc; /* is reloc info present */
};
```

```
struct asym {
    char a_name[8];       /* symbol name */
    unsigned short a_type; /* type flag */
    unsigned long a_value; /* symbol value */
};
```

Lseek, Open and Fcntl Flags

Routines concerned: **lseek()**

```
#include <unistd.h>
```

Define	Value	Meaning
SEEK_SET	0	from beginning of file
SEEK_CUR	1	from current location
SEEK_END	2	from end of file

Routines concerned: **open()**

```
#include <fcntl.h>
```

Define	Value	Meaning
O_RDONLY	0x00	read only
O_WRONLY	0x01	write only
O_RDWR	0x02	read/write
O_NDELAY	0x04	non-blocking I/O
O_SYNC	0x08	sync after writes (MiNT)
O_APPEND	0x10	position at EOF
O_CREAT	0x20	create new file if needed
O_TRUNC	0x40	make file 0 length
O_EXCL	0x80	error, if file exists

Routines concerned: **access()**

```
#include <fcntl.h>
```

Define	Value	Meaning
F_OK	0	present
X_OK	1	executable
W_OK	2	writable
R_OK	4	readable

Routines concerned: **fcntl()**

```
#include <fcntl.h>
```

Define	Value	Meaning
F_DUPFD	0	Duplicate fildes (MiNT)
F_GETFD	1	Get fildes flags (MiNT)
F_SETFD	2	Set fildes flags (MiNT)
F_GETFL	3	Get file flags (MiNT)
F_SETFL	4	Set file flags (MiNT)
F_GETLK	5	Get file lock (MiNT)
F_SETLK	6	Set file lock (MiNT)

Locking under MiNT currently only works for FIFOs.

```
struct flock {
    short l_type;          /* type of lock */
    long l_start;         /* start of locked reg. */
    long l_len;           /* 0 for rest of file */
    short l_pid;          /* set by F_GETLK */
};
```

Define	Value	Meaning
F_RDLCK	O_RDONLY	Read lock (MiNT)
F_WRLCK	O_WRONLY	Write lock (MiNT)
F_UNLCK	3	Unlock (MiNT)

Ioctl Commands and Structures

Routines concerned: **ioctl()**

```
#include <ioctl.h>
```

```
struct tchars {
    char t_intrc;         /* interrupt character */
    char t_quitc;        /* quit character */
    char t_start;        /* start output character */
    char t_stopc;        /* stop output character */
    char t_eofc;         /* EOF character */
    char t_brkc;         /* break character */
};
```

Define	Value	Meaning
TIOCGETP	((('T' << 8) 0)	get param. - gtty
TIOCSETP	((('T' << 8) 1)	set param. - stty
TIOCSETN	TIOCSETP	set param., no flush
TIOCGETC	((('T' << 8) 2)	get spec. char.
TIOCSETC	((('T' << 8) 3)	set spec. char.
TIOCGITC	((('T' << 8) 4)	get loc. spec. char.
TIOCSLTC	((('T' << 8) 5)	set loc. spec. char.

```
struct ltchars {
    char t_suspc;        /* stop process sig. */
    char t_dsuspc;      /* delayed stop process sig. */
    char t_rprmtc;      /* reprint line */
    char t_flushc;      /* flush output */
    char t_werasc;      /* word erase */
    char t_inxctc;      /* literal next char. */
};
```

The following defines apply only, when MiNT is running.

Define	Value	Meaning
FIONREAD	((('F' << 8) 1)	get # byte to rd
FIONWRITE	((('F' << 8) 2)	
TIOCGPGRP	((('T' << 8) 6)	get pgrp of tty
TIOCSPGRP	((('T' << 8) 7)	set pgrp of tty
TIOCFLUSH	((('T' << 8) 8)	flush buffers
TIOCSTOP	((('T' << 8) 9)	stop output
TIOCSTART	((('T' << 8) 10)	start output
PPROCADDR	((('P' << 8) 1)	
PBASEADDR	((('P' << 8) 2)	

```
struct sgttyb {
    char sg_ispeed;      /* input speed */
    char sg_ospeed;     /* output speed */
    char sg_erase;      /* erase character */
    char sg_kill;       /* kill character */
    short sg_flags;     /* flags */
};
```

Define	Value	Meaning
CRMOD	0x0001	map \r to \r\n on output
CBREAK	0x0002	half-cooked mode
ECHO	0x0004	echo input
XTABS	0x0008	expand tabs on output
RAW	0x0010	no I/O processing
LCASE	0x0020	will never do anything
TANDEM	0x0000	not needed
META	0x0100	Alternate as meta key
EVENP	0x4000	even parity
ODDP	0x8000	odd parity
ANYP		speed betw. 300-9600 Bd
ALDELAY	0	

Signals

Routines concerned: **signal()**, **raise()**

```
#include <signal.h>
```

Define	Value	Meaning
SIGNULL	0	not really a signal
SIGHUP	1	hangup signal
SIGINT	2	sent by ^C
SIGQUIT	3	quit signal
SIGILL	4	illegal instruction
SIGTRAP	5	trace trap
SIGABRT	6	abort signal
SIGIOT	SIGABRT	
SIGPRIV	7	privilege violation
SIGEMT	SIGPRIV	
SIGFPE	8	divide by zero
SIGKILL	9	cannot be ignored
SIGBUS	10	bus error
SIGSEGV	11	illegal mem. reference
SIGSYS	12	bad arg. to a syscall
SIGPIPE	13	broken pipe
SIGALRM	14	alarm clock
SIGTERM	15	software term. signal
SIGURG	16	urg. cond. on I/O chan.
SIGSTOP	17	stop sig. not from term.
SIGTSTP	18	stop sig. from terminal
SIGCONT	19	cont. stopped process
SIGCHLD	20	child stopped or exited
SIGTTIN	21	read by bg proc.
SIGTTOU	22	write by bg proc.
SIGIO	23	I/O possible on a descr.
SIGXCPU	24	CPU time exhausted
SIGXFSZ	25	file size lim. exceeded
SIGVTALRM	26	virtual timer alarm
SIGPROF	27	profiling timer expired
SIGWINCH	28	window size changed
SIGUSR	29	user signal 1
SIGUSR	30	user signal 2

Password Information

Routines concerned: **getpwent()**, **getpwuid()**, **getpwnam()**

```
#include <pwd.h>
```

```
struct passwd {
    char *pw_name;           /* login name */
    char *pw_passwd;        /* encrypted password */
    int pw_uid;             /* numeric user-id */
    int pw_gid;             /* numeric group-id */
    char *pw_gecos;         /* GECOS field */
    char *pw_dir;           /* home directory */
    char *pw_shell;         /* default shell */
};
```

Group Information

Routines concerned: **getgrgid()**, **getgrnam()**

```
#include <grp.h>
```

```
struct group {
    char *gr_name;          /* group name */
    char *gr_passwd;        /* encrypted password */
    int gr_gid;             /* numerical group-id */
    char **gr_mem;          /* group members */
};
```

BIOS Parameter Block

Routines concerned: **Getbpb()**

```
#include <osbind.h>
```

```
typedef struct {
    short recsiz;           /* bytes per sector */
    short clsiz;           /* sectors per cluster */
    short clsizb;          /* bytes per cluster */
    short rdlen;           /* root directory size */
    short fsiz;            /* length of fat */
    short fatrec;          /* start sec. of 2nd FAT */
    short datrec;          /* 1st data sector */
    short numcl;           /* no. of clusters on medium */
    short bflags;          /* some flags */
} _BPB;
```

Serial I/O Buffer

Routines concerned: **lore()**

```
#include <osbind.h>
```

```
typedef struct {
    char *ibuf;            /* ptr to data buffer */
    short ibufsiz;         /* len. of buffer */
    short ibufhd;          /* next write pos. */
    short ibuftl;          /* next read pos. */
    short ibufllow;        /* low watermark */
    short ibufhi;          /* high watermark */
} _IOREC;
```

Keyboardtables

Routines concerned: **Keytbl()**

```
typedef struct {
    void *unshift;         /* ptr normal keys */
    void *shift;           /* ptr to shifted keys */
    void *caps;            /* ptr to capslock keys */
} KEYTAB;
```

KBDVECS Structure

Routines concerned: **Kbdvbase()**

```
#include <osbind.h>
```

```
typedef struct {
    void (*midivec)(void); /* MIDI input */
    void (*vkbderr)(void); /* keyboard error */
    void (*vmiderr)(void); /* MIDI error */
    void (*statvec)(void *); /* read IKBD status */
    void (*mousevec)(void *); /* mouse polling */
    void (*clockvec)(void *); /* clock polling */
    void (*joyvec)(void *); /* joystick polling */
    void (*midisys)(void); /* MIDI systemvector */
    void (*ikbdsys)(void); /* IKBD systemvector */
    char kbstate;
} _KBDVECS;
```

Disk Transfer Area

Routines concerned: **Fgetdta()**, **Fsetdta()**

```
#include <osbind.h>
```

```
struct _dta {
    char dta_buf[21];
    char dta_attribute; /* file attribute */
    unsigned short dta_time;
    unsigned short dta_date;
    long dta_size; /* file size */
    char dta_name[14]; /* file name */
};
```

Define	Value	Meaning
FA_RDONLY	0x01	file is readonly
FA_HIDDEN	0x02	file is hidden
FA_SYSTEM	0x04	system file
FA_LABEL	0x08	entry is volume label
FA_DIR	0x10	entry is a directory
FA_CHANGED	0x20	file has changed

Diskinfo

Routines concerned: **Dfree()**

```
typedef struct {
    long b_free;           /* no. of free clusters */
    long b_total;          /* no. of clusters */
    long b_secsiz;         /* bytes per sector */
    long b_clsiz;          /* sect. per cluster */
} DISKINFO;
```

File Timestamp

Routines concerned: **Fdatetime()**

```
#include <osbind.h>

typedef struct {
    short time;           /* time as in Tgettext */
    short date;          /* date as in Tgetdate */
} _DOSTIME;
```

A good alternative might be:

```
typedef struct {
    timeinfo time;       /* time as in Tgettext */
    dateinfo date;      /* date as in Tgetdate */
} _DOSTIME;
```

```
typedef struct {
    unsigned hours : 5;
    unsigned minutes : 6;
    unsigned seconds : 5;
} timeinfo;
```

```
typedef struct {
    unsigned year : 7;
    unsigned month : 4;
    unsigned day : 5;
} dateinfo;
```

HDINFO Structure

```
#include <sysvars.h>

typedef struct {
    short puns;
    char v_p_un[16];
    long pstart[16];
    short bpbs[1];       /* really 16 BPB's */
} HDINFO;
```

XBRA

```
#include <xbra.h>

typedef struct __xbra {
    long xbra_magic;
    long gnucl_magic;
    xptr next;
    short jump;
    void (*this)(void);
} xbra_struct;
```

Define	Value	Meaning
_XBRA_MAGIC	0x58425241L	/* "XBRA" */
_GNUCL_MAGIC	0x474E5543L	/* "GNUCL" */
_JMP_OPCODE	0x4EF9	
_JSR_OPCODE	0x4EB9	

OS Header

The system variable `_sysbase` points to this structure.

```
typedef struct {
    unsigned short os_entry; /* entry address (BRA ..) */
    unsigned short os_version; /* OS version */
    void (*reset)(void); /* startaddress of OS-code */
    struct _osheader *os_beg; /* base of OS */
    void *os_end; /* start of "free" ram */
    char *os_rsv1; /* default shell */
    char *os_magic; /* ptr to GEM-MAGIC var. */
    long os_dat; /* creation date 0xMMDDYYYY */
    unsigned short os_conf; /* bit 0 = 1 => PAL, NTSC otherwise */
    unsigned short os_dosdate; /* creationdate in TOS form. */
    char **p_root; /* base of OS pool */
    char **pkbshift; /* kb d shift state var */
    char **p_run; /* PID of current proc. */
    char *p_rsv2; /* reserved */
} OSHDR;
```

LineA Structure

Routines concerned: **linea0()**

```
#include <linea.h>

typedef struct {
    short _VPLANES; /* 0 # of planes */
    short _VWRAP; /* 2 bytes / scan line */
    short *_CONTRL; /* 4 ptr to CONTRL array */
    short *_INTIN; /* 8 ptr to INTIN array */
    short *_PTSIN; /* 12 ptr to PTSIN array */
    short *_INTOUT; /* 16 ptr to INTO array */
    short *_PTSOUT; /* 20 ptr to PTSOUT array */
    short _COLBIT0; /* 24 plane 0 color value */
    short _COLBIT1; /* 26 plane 1 color value */
    short _COLBIT2; /* 28 plane 2 color value */
    short _COLBIT3; /* 30 plane 3 color value */
    short _LSTLIN; /* 32 draw last pix. of line (=0) */
    short _LNMASK; /* 34 linemask */
    short _WMODE; /* 36 writing mode */
    short _X1; /* 38 X1 coord. */
    short _Y1; /* 40 Y1 coord. */
    short _X2; /* 42 X2 coord. */
    short _Y2; /* 44 Y2 coord. */
    short *_PATPTR; /* 46 ptr to fill pattern */
    short _PATMSK; /* 50 len. in words of pattern */
    short _MFILL; /* 52 multi plane fill flag */
    short _CLIP; /* 54 clipping flag (=1) */
    short _XMINCL; /* 56 min X of clipp. window */
    short _YMINCL; /* 58 min Y of clipp. window */
    short _XMAXCL; /* 60 max X of clipp. window */
    short _YMAXCL; /* 62 max Y of clipp. window */
    short _XDDA; /* 64 accu for scaling */
    short _DDAINC; /* 66 scaling factor */
}
```

```

short _SCALDIR; /* 68 scaling dir. 0 == down */
short _MONO; /* 70 mono spaced flag */
short _SOURCEX; /* 72 X coord. of char. in form */
short _SOURCEY; /* 74 Y coord. of char. in form */
short _DESTX; /* 76 X coord. of char. on screen */
short _DESTY; /* 78 Y coord. of char. on screen */
short _DELX; /* 80 width of character */
short _DELY; /* 82 height of character */
short *_FBASE; /* 84 ptr to font image */
short _FWIDTH; /* 88 width of font image */
short _STYLE; /* 90 style flags */
short _LITEMASK; /* 92 mask for lightening text */
short _SKEWMASK; /* 94 mask for skewing text */
short _WEIGHT; /* 96 add. width of bold text */
short _ROFF; /* 98 off. above basel. when skewing
*/
short _LOFF; /* 100 off. below basel. when skewing
*/
short _SCALE; /* 102 scaling Flag (=1) */
short _CHUP; /* 104 character rotation vector. */
short _TEXTFG; /* 106 text foreground color */
char *_SCRTPAD; /* 108 1k scratchpad area */
short _SCRPT2; /* 112 Off. of the scaling buffer */
short _TEXTBG; /* 114 text background color */
short _COPYTRAN; /* 116 copy raster form type flag */
short (*_SEEDABORT)(void); /* 118 Ptr to seed abort function */
} __LINEA;

extern __LINEA *_aline; /* from linea0() */

```

Font Header

Routines concerned: **linea8()**

```

#include <linea.h>
typedef struct _font {
short font_id; /* 0 font face identifier */
short size; /* 2 font size in points */
char name[32]; /* 4 face name */
short first_ade; /* 36 low. ADE value in face */
short last_ade; /* 38 high. ADE value in face */
short top; /* 40 dist. topline ↔ baseline */
short ascent; /* 42 dist. ascentline ↔ baseline */
short half; /* 44 dist. halfline ↔ baseline */
short descent; /* 46 dist. decentline ↔ baseline */
short bottom; /* 48 dist. bottomline ↔ baseline */
short max_char_width; /* 50 width of widest char. */
short max_cell_width; /* 52 width of widest char. cell */
short left_offset; /* 54 left offset */
short right_offset; /* 56 right offset */
short thicken; /* 58 add. width of bold char. */
short ul_size; /* 60 width of underline */
short lighten; /* 62 mask for lighten char. */
short skew; /* 64 mask for skewing char. */
short flags; /* 66 flags */
char *h_table; /* 68 ptr to hor. off. table */
short *off_table; /* 72 ptr to char. off. table */
}

```

```

char *dat_table; /* 76 ptr to font data */
short form_width; /* 80 form width */
short form_height; /* 82 form height */
struct _font *next_font; /* 84 ptr to next font */
} __FONT;

```

extern __FONT **_fonts; /* from linea0() */

Bit Blt Parameter Block Type

Routines concerned: **linea7()**

```

#include <linea.h>
typedef struct {
short bb_b_wd; /* width of block */
short bb_b_ht; /* height of block */
short bb_plane_ct; /* number of planes */
short bb_fg_col; /* fg color */
short bb_bg_col; /* bg color */
OP_TAB bb_op_tab; /* logic for fg x bg comb. */
SDDB bb_s; /* src info block */
SDDB bb_d; /* dst info block */
short *bb_p_addr; /* pattern buffer address */
short bb_p_nxln; /* off. to next line in pat. */
short bb_p_nxpl; /* off. to next plane in pat. */
short bb_p_mask; /* pattern index mask */
char bb_fill[24]; /* work space */
} BBPB;

```

OP_TAB of Bit Blt Parameter Block

Routines concerned: **linea7()**

```

#include <linea.h>
typedef struct {
char fg0bg0; /* logic op for FG && BG = 0 */
char fg0bg1; /* logic op for FG = 0 && BG = 1 */
char fg1bg0; /* logic op for FG = 1 && BG = 0 */
char fg1bg1; /* logic op for FG && BG = 1 */
} OP_TAB;

```

Define	Value	Type
ALL_WHITE	0	
S_AND_D	1	
S_AND_NOTD	2	
S_ONLY	3	
NOTS_AND_D	4	
D_ONLY	5	
S_XOR_D	6	
S_OR_D	7	
NOT_SORD	8	
NOT_SXORD	9	
D_INVERT	10	
NOT_D	10	
S_OR_NOTD	11	
NOT_S	12	
NOTS_OR_D	13	
NOT_SANDD	14	
ALL_BLACK	15	

SRC and DST Description Block Type

Routines concerned: **linea7()**

```
#include <linea.h>
typedef struct {
    short bl_xmin;    /* minimum x */
    short bl_ymin;    /* minimum y */
    char *bl_form;    /* short aligned memory form */
    short bl_nxwd;    /* off. to next word in line */
    short bl_nxln;    /* off. to next line in plane */
    short bl_nxpl;    /* off. to next plane */
} SDDB;
```

Memory Form Definition Block Type

Routines concerned: **lineae()**

```
#include <linea.h>
typedef struct {
    char *fd_addr;    /* ptr to memory block */
    short fd_w;        /* form width */
    short fd_h;        /* form height */
    short fd_wdwidth; /* form width in shorts */
    short fd_stand;   /* form fmt 0 = dev spec, 1 = std */
    short fd_nplanes; /* number of memory planes */
    short fd_r1;      /* reserved */
    short fd_r2;      /* reserved */
    short fd_r3;      /* reserved */
} MFDB;
```

Routines concerned: **vro_cpyfm()**, **vrt_cpyfm()**, **vr_trnfm()**

```
#include <gemfast.h>
typedef struct form_definition {
    long fd_addr;
    ... ; /* rest as above */
} MFDB;
```

Mouse Form Definition Block Type

Routines concerned: **lineab()**

```
#include <linea.h>
typedef struct {
    short mf_xhot;    /* off. to x hot spot */
    short mf_yhot;    /* off. to y hot spot */
    short mf_nplanes; /* set to 1 */
    short mf_bg;      /* cursor color */
    short mf_fg;      /* mask color */
    short mf_mask[16]; /* mask data */
    short mf_data[16]; /* cursor data */
} MFORM;
```

Routines concerned: **graf_mouse()**

```
#include <gemfast.h>
typedef struct mouse_form {
    ... ; /* same as above */
} MFORM;
```

Define	Value	Meaning
ARROW	0	
BEE	2	
BUSY_BEE	BEE	/* alias */
TEXT_CRSR	1	
HOURLASS	2	
POINT_HAND	3	
FLAT_HAND	4	
THIN_CROSS	5	
THICK_CROSS	6	
OUTLN_CROSS	7	
USER_DEF	255	
M_OFF	256	mouse on
M_ON	257	mouse off

Sprite Definition Block Type

Routines concerned: **linead()**

```
#include <linea.h>
typedef struct {
    short sp_xhot;    /* off. to x hot spot */
    short sp_yhot;    /* off. to y hot spot */
    short sp_format; /* format SP_VDI or SP_XOR */
    short sp_bg;      /* bg color */
    short sp_fg;      /* fg color */
    short sp_data[32]; /* sprite data */
} SFORM;
```

AES Object Type

Routines concerned: too many to mention.

```
#include <gemfast.h>
typedef struct object {
    short ob_next;    /* → object's next sibling */
    short ob_head;    /* → head of object's children */
    short ob_tail;    /* → tail of object's children */
    unsigned short ob_type; /* type of object */
    unsigned short ob_flags; /* flags */
    unsigned short ob_state; /* state */
    unsigned long ob_spec; /* object-specific data */
    short ob_x;        /* upper left corner of object */
    short ob_y;        /* upper left corner of object */
    short ob_width;   /* width of obj */
    short ob_height;  /* height of obj */
} OBJECT;
```

Define	Value	Object Type
G_BOX	20	box with attributes
G_TEXT	21	graphics text
G_BOXTEXT	22	box with graphics text
G_IMAGE	23	simple picture
G_USERDEF	24	user function
G_IBOX	25	invisible box
G_BUTTON	26	centered text in box
G_BOXCHAR	27	box with single character
G_STRING	28	string
G_FTEXT	29	formatted graphics text
G_FBOXTEXT	30	box w. form. graphics text
G_ICON	31	icon
G_TITLE	32	title of drop down menu

Define	Value	Object Flags
NONE	0x0000	
SELECTABLE	0x0001	
DEFAULT	0x0002	default choice
EXIT	0x0004	exit, when selected
EDITABLE	0x0008	object is editable
RBUTTON	0x0010	
LASTOB	0x0020	last object in tree
TOUCHEXIT	0x0040	
HIDETREE	0x0080	
INDIRECT	0x0100	

Define	Value	Object State
NORMAL	0x0000	
SELECTED	0x0001	
CROSSED	0x0002	
CHECKED	0x0004	
DISABLED	0x0008	not selectable
OUTLINED	0x0010	
SHADOWED	0x0020	

Graphic- and VDI-Rectangle

```
#include <gemfast.h>
typedef struct graphic_rectangle {
    int g_x;
    int g_y;
    int g_w;
    int g_h;
} GRECT;

typedef struct vdi_rectangle {
    int v_x1;
    int v_y1;
    int v_x2;
    int v_y2;
} VRECT;

typedef struct orect {
    struct orect *o_link;
    int o_x;
    int o_y;
    int o_w;
    int o_h;
} ORECT;
```

TEDINFO Structure Type

```
#include <gemfast.h>
typedef struct text_edinfo {
    char *te_ptext; /* ptr to text */
    char *te_ptmplt; /* ptr to template */
    char *te_pvalid; /* ptr to validation chrs. */
    short te_font; /* font */
    short te_junk1; /* junk word */
    short te_just; /* justification */
    short te_color; /* color information word */
    short te_junk2; /* junk word */
    short te_thickness; /* border thickness */
    short te_txtlen; /* length of text string */
    short te_tmplen; /* length of template string */
} TEDINFO;
```

ICON Block Type

```
#include <gemfast.h>
typedef struct icon_block {
    short *ib_pmask; /* ptr to icon mask */
    short *ib_pdata; /* ptr to icon data */
    char *ib_ptext; /* ptr to icon text */
    short ib_char; /* char. to appear in icon */
    short ib_xchar; /* x-pos. of char. */
    short ib_ychar; /* y-pos. of char. */
    short ib_xicon; /* x-pos. of icon */
    short ib_yicon; /* y-pos. of icon */
    short ib_wicon; /* width of icon */
    short ib_hicon; /* height of icon */
    short ib_xtext; /* x-pos. of text */
    short ib_ytext; /* y-pos. of text */
    short ib_wtext; /* textwidth in pixels */
    short ib_htext; /* textheight in pixels */
    short ib_resvd;
} ICONBLK;
```

Resource File Header

```
#include <gemfast.h>
typedef struct rshdr {
    short rsh_vrsn;
    short rsh_object; /* pos. of object field */
    short rsh_tedinfo; /* pos. of TEDINFO structs */
    short rsh_iconblk; /* pos. of ICONBLK structs */
    short rsh_bitblk; /* pos. of BITBLK structs */
    short rsh_frstr; /* pos. of free strings */
    short rsh_string; /* pos. of bound strings */
    short rsh_imdata; /* image data */
    short rsh_frimg; /* pos. of free images */
    short rsh_trindex; /* pos. of object tree table */
    short rsh_nobs; /* counts of various structs */
    short rsh_ntree; /* no. of object trees */
    short rsh_nted; /* no. of TEDINFO structs */
    short rsh_nib; /* no. of ICONBLK structs */
    short rsh_nbb; /* no. of BITBLK structs */
    short rsh_nstring; /* no. of strings */
    short rsh_nimages; /* no. of images */
    short rsh_rssize; /* total bytes in resource */
} RSHDR;
```

Bit-Image-Structure (BITBLK)

```
#include <gemfast.h>
typedef struct bit_block {
    char *bi_pdata; /* ptr to bit forms data */
    short bi_wb; /* width of form in bytes */
    short bi_hl; /* height in lines */
    short bi_x; /* source x in bit form */
    short bi_y; /* source y in bit form */
    short bi_color; /* fg color of bit */
} BITBLK;
```

Application-Block-Structure

```
#include <gemfast.h>

typedef struct user_block {
    int (*ub_code)(void *parmblock);
        /* ptr to user function */
    long ub_parm;
        /* optional parameter */
} USERBLK;
```

Parameter-Block-Structure

```
#include <gemfast.h>

typedef struct parm_block {
    OBJECT *pb_tree;
        /* ptr to object tree */
    short pb_obj;
        /* object number */
    short pb_prevstate;
        /* previous state */
    short pb_currstate;
        /* current state */
    short pb_x, pb_y;
        /* x- & y-pos. of object */
    short pb_w, pb_h;
        /* width & height of object */
    short pb_xc, pb_yc;
        /* x- & y-pos. of clip. rect. */
    short pb_wc, pb_hc;
        /* width & height of clip. rect. */
    long pb_parm;
        /* parameter from USERBLK */
} PARMBLK;
```

Multi Event Flags

Routines concerned: **evnt_multi()**

```
#include <gemfast.h>
```

Define	Value	Event
MU_KEYBD	0x0001	keyboard event
MU_BUTTON	0x0002	mouse button event
MU_M1	0x0004	first mouse event
MU_M2	0x0008	second mouse event
MU_MESAG	0x0010	message event
MU_TIMER	0x0020	timer event

Keyboard States

Routines concerned: **evnt_button()**, **evnt_mouse()**, **evnt_multi()**, **graf_mkstate()**

```
#include <gemfast.h>
```

Define	Value	Meaning
K_RSHIFT	0x0001	right shift key
K_LSHIFT	0x0002	left shift key
K_CTRL	0x0004	control key
K_ALT	0x0008	alternate key

Message Event, Window Handling

```
#include <gemfast.h>
```

Routines concerned: **wind_create()**

Define	Value	Window Attribute
NAME	0x0001	name line
CLOSER	0x0002	closing field
FULLER	0x0004	full size field
MOVER	0x0008	move field
INFO	0x0010	info line
SIZER	0x0020	change size field
UPARROW	0x0040	up arrow
DNARROW	0x0080	down arrow
VSLIDE	0x0100	vertical slider
LFARROW	0x0200	left arrow
RTARROW	0x0400	right arrow
HSLIDE	0x0800	horizontal slider

Routines concerned: **evnt_mesag()**, **evnt_multi()**

Define	Value	Message
MN_SELECTED	10	drop down menu sel.
WM_REDRAW	20	window needs redraw
WM_TOPPED	21	make window current
WM_CLOSED	22	window closed
WM_FULLED	23	window to full size
WM_ARROWED	24	arrow message
WM_HSLID	25	horiz. slider moved
WM_VSLID	26	vert. slider moved
WM_SIZED	27	window size changed
WM_MOVED	28	window moved
WM_NEWTOP	29	window activated
AC_OPEN	40	accessory selected
AC_CLOSE	41	accessory closed

Define	Value	Arrow Message
WA_UPPAGE	0	page up
WA_DNPAGE	1	page down
WA_UPLINE	2	line up
WA_DNLINE	3	line down
WA_LFPAGE	4	page left
WA_RTPAGE	5	page right
WA_LFLINE	6	column left
WA_RTLINE	7	column right

Routines concerned: **wind_set()**

Define	Value	Meaning
WF_NAME	2	set window name
WF_INFO	3	set info line
WF_CURRXYWH	5	set window size
WF_HSLIDE	8	set horiz. slider pos.
WF_VSLIDE	9	set vert. slider pos.
WF_TOP	10	make window current
WF_NEWWDESK	14	new default object tree
WF_HSLSIZE	15	set size of horiz. slider
WF_VSLSIZE	16	set size of vert. slider

Routines concerned: **wind_get()**

Define	Value	Arrow Message
WF_KIND	1	reserved
WF_WORKXYWH	4	calc. window work area
WF_CURRXYWH	5	calc. compl. window sz
WF_PREVXYWH	6	calc. prev. window sz
WF_FULLXYWH	7	calc. max. window size
WF_HSLIDE	8	get horiz. slider pos.
WF_VSLIDE	9	get vert. slider pos.
WF_TOP	10	handle of top window
WF_FIRSTXYWH	11	get first rectangle
WF_NEXTXYWH	12	get next rectangle
WF_RESVD	13	reserved
WF_HSLSIZE	15	get horiz. slider size
WF_VLSIZE	16	get vert. slider size
WF_SCREEN	17	addr. of internal buffer

Routines concerned: **wind_update()**

Define	Value	Type
END_UPDATE	0	screen redraw ready
BEG_UPDATE	1	begin of screen redraw
END_MCTRL	2	
BEG_MCTRL	3	

Form_Dial Opcodes

Routines concerned: **form_dial()**

#include <gemfast.h>

Define	Value	Meaning
FMD_START	0	reserve area for dial. box
FMD_GROW	1	draw growing rectangle
FMD_SHRINK	2	draw shrinking rectangle
FMD_FINISH	3	release area

Resource Structure Defs

Routines concerned: **rsrc_gaddr()**, **rsrc_saddr()**

#include <gemfast.h>

Define	Value	Type
R_TREE	0	ptr to object tree
R_OBJECT	1	ptr to object struct
R_TEDINFO	2	ptr to text object
R_ICONBLK	3	ptr to icon
R_BITBLK	4	ptr to BITBLK
R_STRING	5	ptr to string
R_IMAGEDATA	6	ptr to image
R_OBSPEC	7	ptr to TEDINFO struct
R_TEPTTEXT	8	ptr to text
R_TEPTMPLT	9	ptr to text mask
R_TEPVALID	10	ptr to text template
R_IBPMASK	11	ptr to icon mask
R_IBPDATA	12	ptr to icon data
R_IBPTEXT	13	ptr to icon text
R_BIPDATA	14	ptr to bit pattern
R_FRSTR	15	addr. of ptr to free string
R_FRIMG	16	addr. of ptr to free image

GEM Meta File Format

Routines concerned: **v_meta_extents()**, **vm_pagesize()**, **vm_coords()**

```
typedef struct metaheader {
    short mf_header; /* -1 (0xffff) */
    short mf_hlength; /* length of header (24) */
    short mf_version; /* version number (usually 101 or 301)
                       */
    short mf_ndcrlf; /* NDC/RC flag (0 or 2) */
    short mf_extents[4]; /* set with v_meta_extents() */
    short mf_pagesz[2]; /* pagesize in 1/10 mm */
    short mf_coords[4]; /* coordinate system */
    short mf_imgflag; /* ≠0, if bit image */
    short mf_resvd[9]; /* unused */
} METAHDR;
```

All values in metafiles are in Intel format (Little Endian).

GEM IMG File Format

```
typedef struct imgheader {
    short im_version; /* version number (usually 1) */
    short im_headlength; /* length of header */
    short im_nplanes; /* number of planes */
    short im_patlen; /* no. of bytes for "pattern run" */
    short im_pixwidth; /* width of pixel (mm/1000) */
    short im_pixheight; /* height of pixel (mm/1000) */
    short im_scanwidth; /* width of "scan-line" in pixel */
    short im_nlines; /* number of "scan-lines" */
} IMGHEADER;
```

Copyright © 1991 Frank Ridderbusch
 This multicolumn format was originally designed
 by Stephen Gildea and modified to fit the Atari ST
 system variables by Frank Ridderbusch

Permission is granted to make and distribute copies of this card provided the copyright notice and this permission notice are preserved on all copies.