

FileX Services

© 2006 by Express Logic, Inc.

FileX Directory Services

```
UINT fx_directory_attributes_read(FX_MEDIA *media_ptr, CHAR
    *directory_name, UINT *attributes_ptr);
UINT fx_directory_attributes_set(FX_MEDIA *media_ptr, CHAR
    *directory_name, UINT attributes);
UINT fx_directory_create(FX_MEDIA *media_ptr, CHAR
    *directory_name);
UINT fx_directory_default_get(FX_MEDIA *media_ptr, CHAR
    **return_path_name);
UINT fx_directory_default_set(FX_MEDIA *media_ptr, CHAR
    *new_path_name);
UINT fx_directory_delete(FX_MEDIA *media_ptr, CHAR
    *directory_name);
UINT fx_directory_first_entry_find(FX_MEDIA *media_ptr, CHAR
    *directory_name);
UINT fx_directory_first_full_entry_find(FX_MEDIA *media_ptr,
    CHAR *directory_name, UINT *attributes,
    ULONG *size, UINT *year, UINT *month, UINT
    *day, UINT *hour, UINT *minute, UINT *second);
UINT fx_directory_information_get(FX_MEDIA *media_ptr, CHAR
    *directory_name, UINT *attributes,
    ULONG *size, UINT *year, UINT *month, UINT *day,
    UINT *hour, UINT *minute, UINT *second);
UINT fx_directory_local_path_clear(FX_MEDIA *media_ptr);
UINT fx_directory_local_path_get(FX_MEDIA *media_ptr, CHAR
    **return_path_name);
UINT fx_directory_local_path_restore(FX_MEDIA *media_ptr,
    FX_LOCAL_PATH *local_path_ptr);
UINT fx_directory_local_path_set(FX_MEDIA *media_ptr,
    FX_LOCAL_PATH *local_path_ptr,
    CHAR *new_path_name);
UINT fx_directory_long_name_get(FX_MEDIA *media_ptr, CHAR
    *short_file_name,
    CHAR *long_file_name);
UINT fx_directory_name_test(FX_MEDIA *media_ptr, CHAR
    *directory_name);
UINT fx_directory_next_entry_find(FX_MEDIA *media_ptr, CHAR
    *directory_name);
UINT fx_directory_next_full_entry_find(FX_MEDIA *media_ptr,
    CHAR *directory_name, UINT *attributes,
    ULONG *size, UINT *year, UINT *month, UINT
    *day, UINT *hour, UINT *minute, UINT *second);
UINT fx_directory_rename(FX_MEDIA *media_ptr, CHAR
    *old_directory_name,
    CHAR *new_directory_name);
UINT fx_directory_short_name_get(FX_MEDIA *media_ptr, CHAR
    *long_file_name,
    CHAR *short_file_name);
```

```
UINT fx_media_check(FX_MEDIA *media_ptr, UCHAR
    *scratch_memory_ptr, ULONG scratch_memory_size, ULONG
    error_correction_option, ULONG *errors_detected);
UINT fx_media_format(FX_MEDIA *media_ptr, VOID
    (*driver)(FX_MEDIA *media), VOID *driver_info_ptr, UCHAR
    *memory_ptr, UINT memory_size,
    CHAR *volume_name, UINT number_of_fats, UINT
    directory_entries, UINT hidden_sectors,
    ULONG total_sectors, UINT bytes_per_sector, UINT sectors_per_cluster,
    UINT heads, UINT sectors_per_track);
```

FileX File Services

```
UINT fx_file_allocate(FX_FILE *file_ptr, ULONG size);
UINT fx_file_attributes_read(FX_MEDIA *media_ptr, CHAR
    *file_name, UINT *attributes_ptr);
UINT fx_file_attributes_set(FX_MEDIA *media_ptr, CHAR
    *file_name, UINT attributes);
UINT fx_file_best_effort_allocate(FX_FILE *file_ptr, ULONG
    size, ULONG *actual_size_allocated);
UINT fx_file_close(FX_FILE *file_ptr);
UINT fx_file_create(FX_MEDIA *media_ptr, CHAR *file_name);
UINT fx_file_date_time_set(FX_MEDIA *media_ptr, CHAR *file_name,
    UINT year, UINT month, UINT day, UINT hour, UINT minute, UINT second);
UINT fx_file_delete(FX_MEDIA *media_ptr, CHAR *file_name);
UINT fx_file_open(FX_MEDIA *media_ptr, FX_FILE *file_ptr,
    CHAR *file_name, UINT open_type);
UINT fx_file_read(FX_FILE *file_ptr, VOID *buffer_ptr, ULONG
    request_size, ULONG *actual_size);
UINT fx_file_relative_seek(FX_FILE *file_ptr, ULONG byte_offset, UINT seek_from);
UINT fx_file_rename(FX_MEDIA *media_ptr, CHAR *old_file_name, CHAR *new_file_name);
UINT fx_file_seek(FX_FILE *file_ptr, ULONG byte_offset);
UINT fx_file_truncate(FX_FILE *file_ptr, ULONG size);
UINT fx_file_truncate_release(FX_FILE *file_ptr, ULONG size);
UINT fx_file_write(FX_FILE *file_ptr, VOID *buffer_ptr, ULONG size);
```

FileX Media Services

```
UINT fx_media_abort(FX_MEDIA *media_ptr);
UINT fx_media_cache_invalidate(FX_MEDIA *media_ptr);
UINT fx_media_close(FX_MEDIA *media_ptr);
UINT fx_media_flush(FX_MEDIA *media_ptr);
UINT fx_media_open(FX_MEDIA *media_ptr, CHAR *media_name,
    VOID (*media_driver)(FX_MEDIA *), VOID *driver_info_ptr,
    VOID *memory_ptr, ULONG memory_size);
UINT fx_media_read(FX_MEDIA *media_ptr, ULONG
    logical_sector, VOID *buffer_ptr);
UINT fx_media_space_available(FX_MEDIA *media_ptr, ULONG
    *available_bytes_ptr);
UINT fx_media_volume_get(FX_MEDIA *media_ptr, CHAR
    *volume_name, UINT volume_source);
UINT fx_media_volume_set(FX_MEDIA *media_ptr, CHAR *volume_name);
UINT fx_media_write(FX_MEDIA *media_ptr, ULONG
    logical_sector, VOID *buffer_ptr);
```

FileX System Services

```
UINT fx_system_date_get(UINT *year, UINT *month, UINT *day);
UINT fx_system_date_set(UINT year, UINT month, UINT day);
    VOID fx_system_initialize(VOID);
UINT fx_system_time_get(UINT *hour, UINT *minute, UINT *second);
UINT fx_system_time_set(UINT hour, UINT minute, UINT second);
```

FileX Unicode Services

```
UINT fx_unicode_directory_create(FX_MEDIA *media_ptr,
    UCHAR *source_unicode_name, ULONG source_unicode_length,
    CHAR *short_name);
UINT fx_unicode_file_create(FX_MEDIA *media_ptr,
    UCHAR *source_unicode_name, ULONG source_unicode_length,
    CHAR *short_name);
ULONG fx_unicode_length_get(UCHAR *unicode_name);
UINT fx_unicode_name_get(FX_MEDIA *media_ptr, CHAR
    *source_short_name,
    UCHAR *destination_unicode_name, ULONG
    *destination_unicode_length);
UINT fx_unicode_short_name_get(FX_MEDIA *media_ptr,
    UCHAR *source_unicode_name, ULONG source_unicode_length,
    CHAR *destination_short_name);
```

FileX Conditional Compilation Defines

```
FX_DONT_UPDATE_OPEN_FILES
FX_FAT_MAP_SIZE
FX_FAULT_TOLERANT
FX_FAULT_TOLERANT_DATA
FX_MAX_LAST_NAME_LEN
FX_MAX_LONG_NAME_LEN
FX_MAX_SECTOR_CACHE
FX_MAX_FAT_CACHE
FX_MEDIA_DISABLE_SEARCH_CACHE
FX_MEDIA_STATISTICS_DISABLE
FX_NO_LOCAL_PATH
FX_NO_TIMER
FX_RENAME_PATH_INHERIT
FX_SINGLE_OPEN_LEGACY
FX_SINGLE_THREAD
FX_UPDATE_RATE_IN_SECONDS
FX_UPDATE_RATE_IN_TICKS
```

expresslogic

FileX API Parameters

FX_ARCHIVE	0x20
FX_BOOT_SECTOR	1
FX_DIRECTORY	0x10
FX_DIRECTORY_ERROR	0x02
FX_DIRECTORY_SECTOR	3
FX_FALSE	0
FX_FAT_CHAIN_ERROR	0x01
FX_FILE_SIZE_ERROR	0x08
FX_HIDDEN	0x02
FX_LOST_CLUSTER_ERROR	0x04
FX_NULL	0
FX_OPEN_FOR_READ	0
FX_OPEN_FOR_READ_FAST	2
FX_OPEN_FOR_WRITE	1
FX_READ_ONLY	0x01
FX_SEEK_BACK	3
FX_SEEK_BEGIN	0
FX_SEEK_END	1
FX_SEEK_FORWARD	2
FX_SYSTEM	0x04
FX_TRUE	1
FX_VOLUME	0x08

FileX API Return Values

FX_SUCCESS	0x00
FX_BOOT_ERROR	0x01
FX_MEDIA_INVALID	0x02
FX_FAT_READ_ERROR	0x03
FX_NOT_FOUND	0x04
FX_NOT_A_FILE	0x05
FX_ACCESS_ERROR	0x06
FX_NOT_OPEN	0x07
FX_FILE_CORRUPT	0x08
FX_END_OF_FILE	0x09
FX_NO_MORE_SPACE	0x0A
FX_ALREADY_CREATED	0x0B
FX_INVALID_NAME	0x0C
FX_INVALID_PATH	0x0D
FX_NOT_DIRECTORY	0x0E
FX_NO_MORE_ENTRIES	0x0F
FX_DIR_NOT_EMPTY	0x10
FX_MEDIA_NOT_OPEN	0x11
FX_INVALID_YEAR	0x12
FX_INVALID_MONTH	0x13
FX_INVALID_DAY	0x14
FX_INVALID_HOUR	0x15
FX_INVALID_MINUTE	0x16
FX_INVALID_SECOND	0x17
FX_PTR_ERROR	0x18
FX_INVALID_ATTR	0x19
FX_CALLER_ERROR	0x20
FX_BUFFER_ERROR	0x21
FX_NOT_IMPLEMENTED	0x22
FX_WRITE_PROTECT	0x23
FX_INVALID_OPTION	0x24
FX_SECTOR_INVALID	0x89
FX_IO_ERROR	0x90
FX_NOT_ENOUGH_MEMORY	0x91
FX_ERROR_FIXED	0x92
FX_ERROR_NOT_FIXED	0x93

ASCII Character Codes in HEX

		most significant nibble							
		0_	1_	2_	3_	4_	5_	6_	7_
least significant nibble	0	NUL	DLE	SP	0	@	P	'	p
	1	SOH	DC1	!	1	A	Q	a	q
	2	STX	DC2	"	2	B	R	b	r
	3	ETX	DC3	#	3	C	S	c	s
	4	EOT	DC4	\$	4	D	T	d	t
	5	ENQ	NAK	%	5	E	U	e	u
	6	ACK	SYN	&	6	F	V	f	v
	7	BEL	ETB	'	7	G	W	g	w
	8	BS	CAN	(8	H	X	h	x
	9	HT	EM)	9	I	Y	i	y
	A	LF	SUB	*	:	J	Z	j	z
	B	VT	ESC	+	;	K	[K	}
	C	FF	FS	,	<	L	\	l	l
	D	CR	GS	-	=	M]	m	}
	E	SO	RS	.	>	N	^	n	~
	F	SI	US	/	?	O	_	o	DEL

Power	Decimal Value	Hexadecimal Value	Power	Decimal Value	Hexadecimal Value
0	1	1	17	131,072	0x20000
1	2	2	18	262,144	0x40000
2	4	4	19	524,288	0x80000
3	8	8	20	1,048,576	0x100000
4	16	0x10	21	2,097,152	0x200000
5	32	0x20	22	4,194,304	0x400000
6	64	0x40	23	8,388,608	0x800000
7	128	0x80	24	16,777,216	0x1000000
8	256	0x100	25	33,554,432	0x2000000
9	512	0x200	26	67,108,864	0x4000000
10	1,024	0x400	27	134,217,728	0x8000000
11	2,048	0x800	28	268,435,456	0x10000000
12	4,096	0x1000	29	536,870,912	0x20000000
13	8,192	0x2000	30	1,073,741,824	0x40000000
14	16,384	0x4000	31	2,147,483,647	0x80000000
15	32,768	0x8000	32	4,294,967,295	0x100000000
16	65,536	0x10000			

ANSI C Format Specifications

%d or %i	decimal integer	%u	unsigned decimal integer
%ld or %li	long decimal integer	%lu	long unsigned decimal integer
%hd or %hi	short decimal integer	%hu	short unsigned decimal integer
%o	octal integer	%c	ASCII character
%lo	long octal integer	%s	ASCII string, null terminated
%ho	short octal integer	%f	floating point
%x or %X	hexadecimal integer	%e or %E	double precision floating point
%lx or %lX	long hexadecimal integer	%g or %G	double precision floating point
%hx or %hX	short hexadecimal integer	%p	void pointer

FILEX

Programmers Guide

expresslogic