PX5 FILE Programmer's Reference Card

11440 West Bernardo Court, Suite 300, San Diego, CA 92127 | (858) 753-1715 | info@px5rtos.com

PX5 FILE APIs

```
int       chmod(char * filename, mode_t new_mode);
int       close(int file_handle);
off_t     lseek(int file_handle, off_t offset, int whence);
int       open(char * filename, int flags);
int       px5_rename(char * oldname, char * newname);
ssize_t read(int file_handle, void * buffer, size_t count);
int       stat(char * filename, struct px5_stat * stat_buffer);
int       unlink(char * filename);
ssize_t   write(int file_handle, void * buffer, size_t count);
```

Directory APIs

```
int       chdir(char * dirname);
int       closedir(DIR * directory_pointer);
char *    getcwd(char * buffer, size_t size);
int       mkdir(char * dirname, mode_t mode);
DIR *     opendir(char * dirname);
struct dirent * readdir(DIR * directory_pointer);
int       rmdir(char * dirname);
void      seekdir(DIR * directory_pointer, off_t offset);
```

Volume APIs

```
int       px5_file_start(void);
int       px5_volume_close(char volume);
int       px5_volume_fault_tolerant_enable(char volume, void * journal_memory,
                                           u_int journal_memory_size);
int       px5_volume_flush(char volume);
int       px5_volume_format(char volume,
                  int (*volume_driver_pointer)(px5_volume_driver_request *request),
                  void * additional_information, void * sector_memory,
                  u_int sector_memory_size, char * oem_name, char * volume_name,
                  u_long volume_id, u_long hidden_sectors, u_long total_sectors,
                  u_long bytes_per_sector, u_long sectors_per_cluster, u_long fats,
                  u_long root_directory_entries,
                  px5_extended_format_information * extended_format_information);
int       px5_volume_open(char volume,
                  int (*volume_driver_pointer)(px5_volume_driver_request *request),
                  void * additional_information, void * sector_memory,
                  u_int sector_memory_size);
```

Media (partition) APIs

```
int       px5_media_close(px5_media *media_pointer);
int       px5_media_partition_read(px5_media *media_pointer,
                                   px5_media_partition_info *partition_info_pointer);
int       px5_media_open(px5_media *media_pointer,
                  int (*media_driver_pointer)(px5_volume_driver_request *request),
                  void *additional_information, void *sector_memory,
                  u_int sector_memory_size);
int       px5_media_partition_info_get(void *first_sector_memory, u_int partition_number,
                  px5_media_partition_info *partition_info_pointer);
```

PX5 FILE Symbols

Version Symbols

PX5_FILE_MAJOR_VERSION
PX5_FILE_MINOR_VERSION
PX5_FILE_UPDATE_VERSION
PX5_FILE_PATCH_VERSION

Error Checking Symbols

PX5_FUNCTION_POINTER_VERIFY_ENABLE
PX5_OBJECT_VERIFY_ENABLE
PX5_STACK_CHECK_ENABLE
PX5_STACK_VERIFY_ENABLE
PX5_PARAMETER_CHECKING_DISABLE

PX5 FILE Configuration Symbols

PX5_ADDITIONAL_FAT_UPDATE_DISABLE
PX5_BIG_ENDIAN
PX5_BYTE_ACCESS
PX5_FILE_DATE_TIME_REFRESH_DISABLE
PX5_FILE_DATE_TIME_REFRESH_PERIOD
PX5_LONGNAME_SIZE
PX5_MAXIMUM_DIRECTORIES
PX5_MAXIMUM_FILES
PX5_MAXIMUM_PATH_SIZE
PX5_MAXIMUM_VOLUMES
PX5_RELEASE_SECTOR_NOTIFICATION_DISABLE
PX5_SUBDIRECTORY_SEARCH_CACHE_SIZE
PX5_VOLUME_CLUSTER_CACHE_SIZE
PX5_VOLUME_SECTOR_CACHE_SIZE

PX5 FILE Error Code

PX5_SUCCESS (0) Success for most APIs	0
PX5_ERROR (-1) Error on non-pthread APIs, use errno for detailed error	-1
EBADF	1
EBUSY	2
EEXIST	3
EFAULT	4
EINVAL	5
EIO	6
EISDIR	7
ENOENT	8
ENOMEM	9
ENOSPC	10
ENOTDIR	11
ENOTEMPTY	12
ERANGE	13
EXDEV	14

[FILE]