

INI 文件读写模块

该模块可以读写 INI 文件,支持字符串,整型,浮点型数据的存储
头文件 `platform_libini.h`
库文件 `advlib`

数据结构

INI文件打开方式

```
typedef enum INI_OPEN_MODE {  
    INI_OPEN_MODE_READ,  
    INI_OPEN_MODE_WRITE,  
    INI_OPEN_MODE_RW,  
    INI_OPEN_MODE_END  
} INI_OPEN_MODE;
```

INI_OPEN_MODE_READ	只读打开, 文件不存在会返回错误码
INI_OPEN_MODE_WRITE	覆写打开, 文件不存在会自动创建, 且原内容被清空
INI_OPEN_MODE_RW	读写打开, 文件不存在会自动创建, 保留原内容

以下的INI函数, 返回值为表示成功, 其他值为失败
下面的参数公共说明

section	INI段名称
key	INI键名称
val	INI键值

函数接口

打开一个INI文件

```
INI_FILE * ini_file_open( const char * path, INI_OPEN_MODE mode );
```

参数	const char * path	INI文件路径
	INI_OPEN_MODE mode	INI打开方式
返回值	INI_FILE *	INI文件句柄

- 1. 返回值为NULL, 则打开失败.
- 2. INI文件使用完毕必须关闭, 以释放资源.

关闭一个INI文件

```
int ini_file_close( INI_FILE * ini_file );
```

参数	INI_FILE * ini_file	INI文件句柄
返回值	int	0成功, 其他值失败

1. 不要关闭不存在的句柄, 不要重复关闭同一个句柄.

读取INI文件中的一个字符串

```
int ini_file_read_string( INI_FILE * ini_file, const char * section, const char * key, char * val, int maxlen );
```

参数	INI_FILE * ini_file	INI文件句柄
	const char * section	INI段名称
	const char * key	INI键名称
	char * val	保存字符串数据的缓存
	int maxlen	最大读取的字符长度
返回值	int	0成功, 其他失败

1. 如果指定位置的INI键, 段不存在, 则返回非0值, 表示读取失败, 并且val的数据不被改变.
2. 如果数据大于maxlen长度, 函数将只截取maxlen长度, 复制到val中.
3. 成功返回0, val中将得到指定的数据.

读取INI文件中的int数据

```
int ini_file_read_int( INI_FILE * ini_file, const char * section, const char * key, int * val );
```

参数	INI_FILE * ini_file	INI文件句柄
	const char * section	INI段名称
	const char * key	INI键名称
	int * val	保存Int数据的地址
返回值	int	0成功, 其他失败

1. 读取的指定键值必须对应的是Int类型, 否则的话, 返回值是不可预知的.
2. Int数据类型是用有符号数字保存的, 比如INT = -1234567

读取INI文件中的unsigned int数据

```
int ini_file_read_uint( INI_FILE * ini_file, const char * section, const char * key, unsigned int * val );
```

参数	INI_FILE * ini_file	INI文件句柄
	const char * section	INI段名称
	const char * key	INI键名称
	unsigned int * val	保存UInt数据的地址
返回值	int	0成功, 其他失败

1. 读取的指定键值必须对应的是unsigned Int类型, 否则的话, 返回值是不可预知的.
2. unsigned Int数据类型是用16进制保存的, 比如UINT = 0x1234567

读取INI文件中的double数据

```
int ini_file_read_double( INI_FILE * ini_file, const char * section, const char * key, double * val );
```

参数	INI_FILE * ini_file	INI文件句柄
	const char * section	INI段名称
	const char * key	INI键名称
	double * val	保存Double数据的地址
返回值	int	0成功, 其他失败

1. 读取的指定键值必须对应的是double类型, 否则的话, 返回值是不可预知的.
2. double数据类型是用普通浮点写法保存的, 比如FLOAT = 1234.5678

写INI文件字符串数据

```
int ini_file_write_string( INI_FILE * ini_file, const char * section, const char * key, const char * val );
```

参数	INI_FILE * ini_file	INI文件句柄
	const char * section	INI段名称
	const char * key	INI键名称
	const char * val	待写入的字符串数据地址
返回值	int	0成功, 其他失败

1. 如果待写入键值不存在, 则程序会创建这个键值, 如果存在, 则改写这个键值.
2. 函数并不判断已经存在的键值类型, 所以你可以直接改写一个键值的属性, 而不会得到任何警告.

写INI文件int数据

```
int ini_file_write_int( INI_FILE * ini_file, const char * section, const char * key, int val );
```

参数	INI_FILE * ini_file	INI文件句柄
	const char * section	INI段名称
	const char * key	INI键名称
	int val	待写入的int数据
返回值	int	0成功, 其他失败

1. 注意事项同上

写INI文件unsigned int数据

```
int ini_file_write_uint( INI_FILE * ini_file, const char * section, const char * key, unsigned int val );
```

参数	INI_FILE * ini_file	INI文件句柄
	const char * section	INI段名称
	const char * key	INI键名称
	int val	待写入的unsigned int数据
返回值	int	0成功, 其他失败

1. 注意事项同上

写INI文件double数据

```
int ini_file_write_double( INI_FILE * ini_file, const char * section, const char * key, double val );
```

参数	INI_FILE * ini_file	INI文件句柄
	const char * section	INI段名称
	const char * key	INI键名称
	int val	待写入的double数据
返回值	int	0成功, 其他失败

1. 注意事项同上