

## XGUI 接口说明

### 创建窗口

```
XGUI_WND * XGUI_CreateWindow( char *name ,
    int x, int y, int width, int height, XGUI_WND * parent,
    ERRCODE (* Callback)( XGUI_WND * hWnd , XGUI_MSG * msg ), void *param );
```

参数	char *name	窗口名称,以\0 结尾的 ANSI 字符串
	int x	窗口在屏幕上的 X 坐标
	int y	窗口在屏幕上的 Y 坐标
	int width	窗口宽度
	int height	窗口高度
	XGUI_WND * parent	父窗口句柄,设为 NULL,则将桌面作为父窗口
	Callback	窗口消息处理回调函数,所有的窗口消息都在这里处理.
	void *param	自定义参数
返回值	XGUI_WND *	窗口句柄

1. 返回值为 NULL 的时候,创建窗口失败,原因可能是参数不正确,或者内存不足.
2. x 和 y 坐标可以设置到屏幕之外,但那样你的窗口无法在桌面上显示出来.
3. 坐标系是左上角为 (0, 0), 向下为 y 坐标正方向, 向右为 x 坐标正方向.
4. 父窗口句柄可以设置为 NULL, 这个窗口将被设置为桌面的父窗口.
5. Callback 函数是你自己定义的一个回调函数, 格式如下:

```
ERRCODE MyCallback( XGUI_WND * hWnd , XGUI_MSG * msg )
{
    ...
}
```

当你的窗口发生了消息时, 窗口管理器就会调用你的函数, 并将消息指针和窗口句柄作为参数传递.  
XGUI\_WND \* hWnd 是该窗口的句柄,  
XGUI\_MSG \* msg 是接收到的消息.
6. void \*param 是用户自定义参数, 在创建窗口的时候, 这个参数可以设为任何值, 它会原封不动的作为 XGUI\_MSG\_INIT 消息的数据部分传递到 Callback 函数中. 并且随时可以用 XGUI\_GetWndUsrParam 和 XGUI\_SetWndUsrParam 函数来获取或设置它的值.

### 销毁窗口

```
ERRCODE XGUI_DestroyWindow( XGUI_WND * hWnd );
```

参数	XGUI_WND * hWnd	待销毁的窗口句柄
返回值	ERRCODE	错误码

1. 不要把销毁的窗口再次销毁, 不要传递无效的参数, 否则函数可能会出错.

2. 返回值是 `ERRCODE_NONE`, 才是正常销毁了窗口, 如果返回了其他的错误码, 请为你的程序准备后事吧:)

转换绝对坐标到窗口坐标

```
void XGUI_RectConv_Abs2Wnd( XGUI_RECT * abs, XGUI_RECT * wnd, XGUI_WND * hWnd );
```

//获取指定窗口矩形

```
XGUI_RECT * XGUI_GetWndRect( XGUI_WND * hWnd );
```

//获取窗口高度

```
int XGUI_GetWndHeight( XGUI_WND * hWnd );
```

//获取窗口宽度

```
int XGUI_GetWndWidth( XGUI_WND * hWnd );
```

//获取窗口自定义参数

```
void * XGUI_GetWndUsrParam( XGUI_WND * hWnd );
```

//设置窗口自定义参数

```
void XGUI_SetWndUsrParam( XGUI_WND * hWnd, void *param );
```

//设置桌面显示模式

```
ERRCODE XGUI_SetScreenMode( int width, int height, int mode );
```

//进入窗口循环

```
ERRCODE XGUI_WindowsManager( void );
```

//填充矩形(支持 Alpha 操作)

```
void XGUI_RectFill( XGUI_WND * hWnd, XGUI_RECT *rect, XGUI_COLOR color );
```

//设置窗口无效(需重绘)

```
ERRCODE XGUI_SetWndInval( XGUI_WND * hWnd );
```

//离开 GUI 循环, 但并不释放资源

```
ERRCODE XGUI_Leave( void );
```

//设置焦点窗口

```
ERRCODE XGUI_SetFocusWnd( XGUI_WND * hWnd );
```