HiGV

API 参考

文档版本　09

发布日期　2021-06-08

商标声明

 、HISILICON、海思和其他海思商标均为海思技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

上海海思技术有限公司

地址：　深圳市龙岗区坂田华为总部办公楼　邮编：518129

网址：　http://www.hisilicon.com/cn/

客户服务邮箱：　support@hisilicon.com

前 言

概述

HiGV 是上海海思数字媒体处理平台提供的 GUI 图形控件开发框架，它提供了任务管理，控件管理，消息模型，语言设置等 GUI 开发接口，并且提供了多种控件类型，主要用于客户基于上海海思平台上开发 GUI 图形应用程序。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3559A	V100
Hi3559C	V100
Hi3556A	V100
Hi3559	V200
Hi3556	V200
Hi3519A	V100
Hi3518E	V300
Hi3516D	V300
Hi3516C	V500
Hi3562	V100
Hi3566	V100
Hi3569	V100
Hi3568	V100

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师
- 软件测试工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
危险	表示如不避免则将会导致死亡或严重伤害的具有高等级风险的危害。
警告	表示如不避免则可能导致死亡或严重伤害的具有中等级风险的危害。
注意	表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。
须知	用于传递设备或环境安全警示信息。如不避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
说明	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2021-12-01	09	第 9 次正式版本发布 2.1.3 小节，HI_GV_Widget_Create、HI_GV_Widget_GetCloneHandle、HIGV_FOCUS_SWITCH_E【注意】涉及修改 2.1.4 小节，HI_GV_Res_CreateStyle、HI_GV_Res_PrintCurLoadResInfo 涉及修改 2.1.5 小节，HI_GV_Msg_SendSync、HI_GV_Msg_SendAsync、HI_GV_Msg_PostAsync【注意】涉及修改 2.1.6.1 小节，HI_GV_WM_GetBindTouchStatus 涉及修改；新增 HI_GV_WM_SetWndMemMode 和 HI_GV_WM_GetWndMemMode 2.1.6.2 小节，新增 HIGV_WndMemType 2.1.8 小节，HI_GV_GraphicContext_DrawImage、HI_GV_GraphicContext_SetBgColor 涉及修改 2.1.9 小节，HI_GV_Anim_GetInfo 涉及修改 2.1.11 小节，HI_GV_Lan_Register 涉及修改 2.1.14 小节，HI_GV_SendInputEvent【注意】涉及修改 2.2.1 小节，HI_GV_Button_SetToggleColor 涉及修改 2.2.2 小节，HI_GV_Win_SyncShow 涉及修改 2.2.3 小节，HI_GV_ScrollGrid_GetCellValue~HI_GV_ScrollGrid_GetCellNum【注意】涉及修改 2.2.4 小节，HI_GV_ScrollBar_SetPos 涉及修改 2.2.5 小节，HIGV_SCROLLVIEW_Init_S 涉及修改 2.2.7.1 小节，HI_GV_Clock_SetSwitchItemStep 涉及修改 2.2.9.2 小节，新增 HIGO_BLTOPT_S 2.2.12 小节，HigvWheelViewStyle 涉及修改 2.2.15.2 小节，新增 HigvIMEWindowInit 2.2.17 小节，HI_GV_MSGBOX_Show 涉及修改 2.2.19.1 小节，删除 HI_GV_MULTIEDIT_ActiveWithIME 2.2.19.2 小节，HigvMultiEditStyle 涉及修改 新增 2.2.20 小节

修订日期	版本	修订说明
2021-04-30	08	第 8 次正式版本发布 2.1.3 小节，新增 HI_GV_Widget_IsRefresh 2.1.4 小节，HIGV_FONT_S 涉及修改 2.1.8 小节，HI_GV_GraphicContext_Create~HI_GV_GraphicContext_AddClipRect 涉及修改 2.1.10 小节，HIGV_TOUCH_EVENT_S 涉及修改 2.2.3 小节，HIGV_SCROLLGRID_STYLE_S【定义】和【成员】涉及修改 2.2.10 小节，HI_GV_ImageEx_SetPos、HI_GV_ImageEx_SetInterval【注意】涉及修改
2021-03-15	07	第 7 次正式版本发布 2.2.9 小节，HI_GV_Image_DrawMemImage 和 HI_GV_Image_FreeMemSurface【注意】涉及修改 2.2.17 小节，HI_GV_MSGBOX_ButtonLayout 和 HI_GV_MSGBOX_GetButtonHandle 涉及修改 2.2.18 小节，HI_GV_CHARTS_Init 和 HI_GV_CHARTS_BindLineData【注意】涉及修改
2020-12-12	06	第 6 次正式版本发布 2.1.5 小节涉及修改 2.1.12 小节，HI_GV_DDB_GetCellData 和 HI_GV_DDB_SetCellData【描述】涉及修改 2.1.13 小节，HI_GV_ADM_CreateDefault, HI_GV_ADM_CreateDefaultByHandle 涉及修改 2.2.5.2 小节，新增 HigvScrollViewCord 2.2.6.1 小节, HI_GV_List_InitEx, HI_GV_List_GetSelCell 涉及修改 2.2.6.2 小节，HIGV_LIST_ATTRIBUTE_S【成员】涉及修改 2.2.8.1 小节，HI_GV_Track_SetSliImage【注意】涉及修改 2.2.16 小节，HI_GV_EDIT_SetType 的【参数】和【注意】涉及修改 2.2.17 小节，新增 HI_GV_MSGBOX_Show

修订日期	版本	修订说明
2020-05-30	05	第 5 次正式版本发布 2.1.8 小节，HI_GV_GraphicContext_DrawLine 涉及修改。 新增 2.2.19
2020-04-15	04	第 4 次正式版本发布 2.2.1.1~2.2.15.1 小节中的【语法】和【参数】均涉及修改 2.2.8 小节，删除 HI_GV_ScrollGrid_RegisterWidget 2.2.13 小节，删除 HI_GV_SlideUnlock_RegisterWidget 新增 2.2.18 小节
2020-02-28	03	第 3 次正式版本发布 2.1.14 小节，删除 HI_GV_InitInputSuite 2.2.15 小节，新增 HI_GV_IMEWINDOW_FlexibleSoftKB 新增 2.2.17 小节
2020-01-20	02	第 2 次正式版本发布 新增 2.1.15 小节 2.2.14.1 小节涉及修改 2.2.14.2 小节，HIGV_SCALEVIEW_STYLE_S 涉及修改 2.2.16 小节，新增 HI_GV_EDIT_ShowReplace 新增表 3-2
2019-11-30	01	第 1 次正式版本发布 2.1.1 小节，新增 HI_GV_SetSyncDraw 删除“Spin 控件”和“Imageview 控件”小节。 新增“2.2.15 输入法窗口控件”和“2.2.16 Edit 控件”小节
2019-10-29	00B08	第 8 次临时版本发布 新增 2.2.16 小节
2018-12-29	00B07	第 7 次临时版本发布 添加 Hi3519AV100 的相关内容 2.1.1 小节，新增 HI_GV_SetVsyncType、HI_GV_GetVsyncType 和

修订日期	版本	修订说明
		HIGV_VSYNC_E 2.2.9 小节，新增 HI_GV_Track_EnableGesture 和 HI_GV_Track_IsGestureEnable
2018-11-15	00B06	第 6 次临时版本发布 2.1.4.1 小节，新增 HI_GV_Res_ReleaseResInfo 2.2.3.1 小节，新增 HI_GV_ScrollGrid_EnableGesture、HI_GV_ScrollGrid_IsGestureEnable
2018-10-30	00B05	第 5 次临时版本发布。 2.1.3 小节，HI_GV_Widget_SetSkin 涉及修改 2.1.4.1 小节，新增 HI_GV_Font_GetTextExtent、HI_GV_AnimInfo_Create、HI_GV_AnimInfo_CreateByHandle、HI_GV_AnimInfo_Destroy、HI_GV_AnimInfo_Get 2.1.4.2 小节，新增 HIGV_ANIM_TYPE_E、HIGV_ANIM_REPEAT_TYPE_E、HIGV_ANIM_ROLL_DIRECTION_E、HIGV_ANIM_TRANSLATE_INFO_S、HIGV_ANIM_ALPHA_INFO_S、HIGV_ANIM_ANY_INFO_S、HIGV_ANIM_ROLL_INFO_S、HIGV_ANIM_INFO_S 2.1.5.1 小节，HI_GV_Msg_SendSync、HI_GV_Msg_SendAsync、HI_GV_Msg_PostAsync、HI_GV_Msg_SendAsyncWithData 涉及修改 2.1.6.1 小节，新增 HI_GV_WM_BindTouchMsg、HI_GV_WM_GetBindTouchStatus 2.1.9.1 小节，HI_GV_TweenAnimAddTween 涉及修改 2.1.12.1 小节，删除 HI_GV_DDB_DumpToFile、HI_GV_DDB_ReadFromFile 2.2.2.1 小节，HI_GV_Win_EndSyncShow 涉及修改 2.2.4.1 小节，新增 HI_GV_ScrollBar_SetStatus HI_GV_ScrollBar_SetSlideRes、HI_GV_ScrollBar_SetButtonImg 涉及修改 2.2.6.1 小节，HI_GV_Spin_AddItemByID、HI_GV_List_IsListBoxType 涉及修改 2.2.13.1 小节，HI_GV_WheelView_SetUpImage、HI_GV_WheelView_SetDownImage、HI_GV_PARSER_LoadViewById、HI_GV_PARSER_ViewGetWinsHandle 涉及修改
2018-09-29	00B04	第 4 次临时版本发布。

修订日期	版本	修订说明
		2.1.1 小节，HI_GV_App_Create【注意】涉及修改 2.1.2 小节，HI_GV_Layer_Create 和 HI_GV_Layer_CreateEx【注意】涉及修改 2.1.3 小节，HI_GV_Widget_Create【注意】涉及修改 2.1.4 小节，HI_GV_Res_CreateID、HI_GV_Res_CreateID_NoPrefixPath、HI_GV_Res_CreateStyle、HI_GV_Res_CreateStyleByHandle、HI_GV_Font_Create、HI_GV_Font_CreateByHandle 和 HI_GV_FontSet_Create【注意】涉及修改 2.1.5 小节，HI_GV_Msg_SendSync、HI_GV_Msg_SendAsync 和 HI_GV_Msg_PostAsync【注意】涉及修改 2.1.8 小节，HI_GV_GraphicContext_Create【注意】涉及修改 2.1.9 小节涉及修改 2.1.13 小节，HI_GV_ADM_Create、HI_GV_ADM_CreateByHandle、HI_GV_ADM_CreateDefault 和 HI_GV_ADM_CreateDefaultByHandle【注意】涉及修改 2.1.15 小节，HI_GV_Timer_Create【注意】涉及修改 2.2.13 小节，HI_GV_WheelView_Create【注意】涉及修改
2018-07-20	00B03	第 3 次临时版本发布。 2.1.1 小节，新增 HI_GV_SetVsyncLostFrame 和 HI_GV_IsVsyncLostFrame 2.1.9 小节，HI_GV_Anim_CreateInstance【注意】涉及修改 2.2.11 小节，新增 HI_GV_ImageEx_SetRepeatCount
2018-06-06	00B02	第 2 次临时版本发布。 2.1 小节，新增 HI_GV_SetSWVsyncPeriod~ HI_GV_IsRefreshCombine 删除 Hi3559V100/Hi3556V100 的相关内容
2018-04-11	00B01	第 1 次临时版本发布。

目 录

插图目录

表格目录

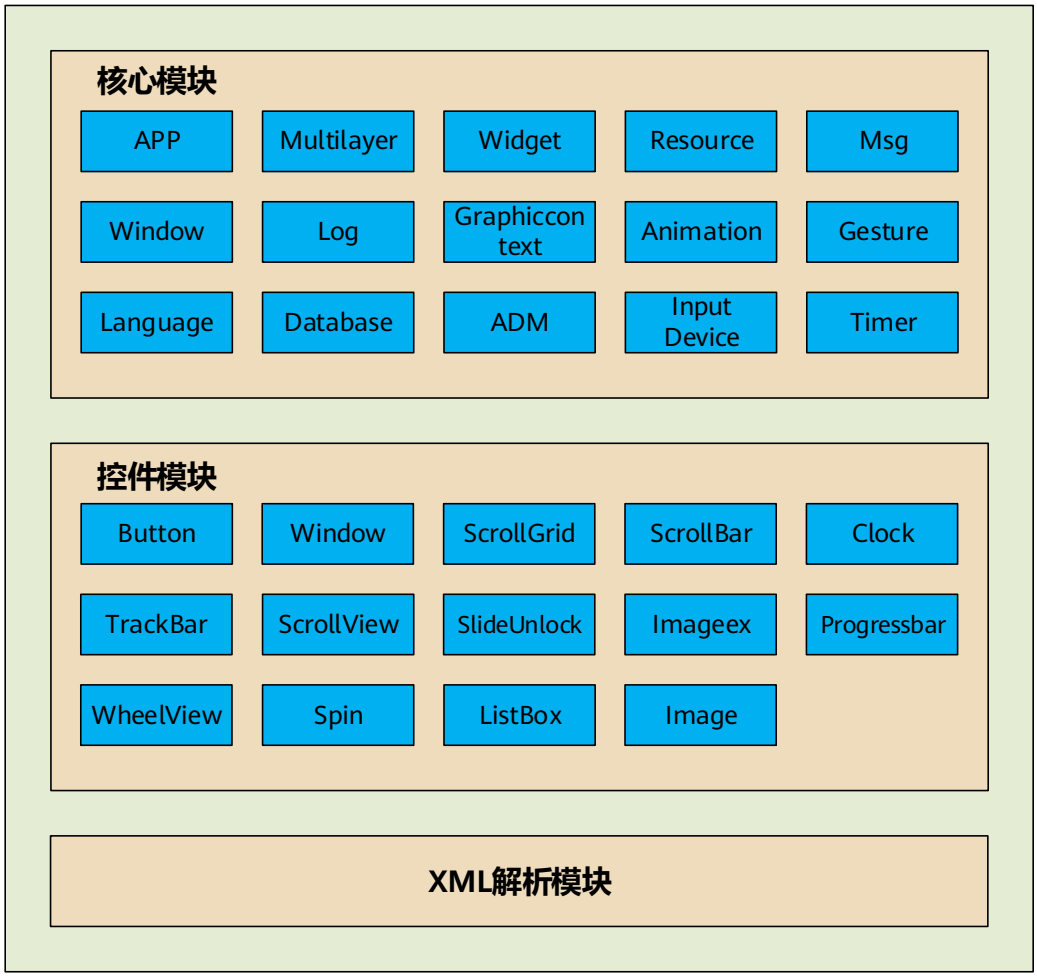
1 概述

1.1 概述

HiGV 是一个轻量级的 GUI 系统，主要是为上海海思芯片平台提供统一的轻量级、高效、易用的 GUI 解决方案。该系统采用分层机制实现，其中底层图形库依赖 HiGO 库，而 HiGO 建立在基本的图形驱动(FrameBuffer、TDE、图片编解码等)之上。

1.2 组成部分

图1-1 HiGV 组件框架图



如图 1-1 所示，HiGV API 接口主要由三部分组成，即核心模块，控件模块和 XML 解析模块。

- 核心模块

  包含 APP（应用程序管理）、Multi layer（多图层管理）、Widget（控件管理）、Resource（资源管理）、Msg（消息管理）、Window（窗口管理）、Log（调试打印管理）、Graphiccontext（绘制上下文）等子模块构成。

- 控件模块

  包含 Button 控件、ScrollGird 控件、ScrollBar 控件、ScrollView 控件、Image 控件、ListBox 控件、WheelView 控件等控件子模块构成。

- XML 解析模块

XML 解析模块只包含一个 XML 解析子模块。

1.3 注意事项

本文主要描述了 HiGV 的 API 接口和数据类型，在上海海思平台上进行 GUI 开发还可参考《HiGV 开发指南》和《HiGV 标签 使用指南》文档，HiGO API 接口说明不在本文描述的范围内。

2 API 参考

2.1 核心模块

2.1.1 应用程序管理

应用程序管理模块是 HiGV 应用程序控制的核心，负责初始化和去初始化、创建和销毁应用程序实例、启动和停止应用程序。

2.1.1.1 接口描述

HiGV 应用程序管理提供以下 API：

- HI_GV_Init：初始化 HiGV。
- HI_GV_Deinit：去初始化 HiGV。
- HI_GV_App_Create：创建 HiGV 应用程序实例。
- HI_GV_App_Destroy：销毁应用程序实例。
- HI_GV_App_Start：启动 HiGV 应用程序。
- HI_GV_App_Stop：停止 HiGV 应用程序。
- HI_GV_App_Active：激活应用程序前台运行。
- HI_GV_App_DeActive：激活应用程序后台运行。
- HI_GV_App_IsActive：获取应用是否处于前台运行状态。
- HI_GV_SetMaxAssignHandle：设置静态指定的 Handle 的最大值。
- HI_GV_GetReserveStartStaticHandle：获取控件构造树保留的静态 handle 起始值。
- HI_GV_SetVsyncType：设置 VSYNC 的信号源类型。

- HI_GV_GetVsyncType：获取 VSYNC 的信号源类型。
- HI_GV_SetSWVsyncPeriod：设置软件 VSYNC 的信号周期。
- HI_GV_GetSWVsyncPeriod：获取软件 VSYNC 的信号周期。
- HI_GV_SetVsyncLostFrame：设置 VSYNC 丢帧策略。
- HI_GV_IsVsyncLostFrame：获取 VSYNC 的丢帧策略。
- HI_GV_SetLostFrameThreshold：设置 VSYNC 丢帧的阈值。
- HI_GV_GetLostFrameThreshold：获取 VSYNC 丢帧的阈值。
- HI_GV_RenderCmdSync：Render 命令同步接口，直到 Render 命令执行完成，接口返回成功。
- HI_GV_SetRefreshCombine：设置 Refresh 命令是否合并。
- HI_GV_IsRefreshCombine：获取 Refresh 命令是否合并状态。
- HI_GV_SetSyncDraw：设置是否进行同步绘制。

HI_GV_Init

【描述】

初始化 HiGV。

【语法】

```
HI_S32 HI_GV_Init(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

参考第 4 章 “开发指引”。

HI_GV_Deinit

【描述】

去初始化 HiGV。

【语法】

```
HI_VOID HI_GV_Deinit(HI_VOID);
```

【参数】

无。

【返回值】

无

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

参考第 4 章 “开发指引”。

HI_GV_App_Create

【描述】

创建 HiGV 应用程序实例。

【语法】

```
HI_S32 HI_GV_App_Create(const HI_CHAR*name, HIGV_HANDLE * appHandle);
```


【参数】

参数名称	描述	输入/输出
name	应用程序名。	输入
appHandle	应用程序实例句柄	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

- name 不能为空。
- 开发者调用 HI_GV_App_Create 创建的资源，程序结束后，由开发者主动调用 HI_GV_App_Destroy 来释放响应资源，否则会有内存泄露。

【举例】

参考第 4 章 “开发指引”。

HI_GV_App_Destroy

【描述】

销毁应用程序。

【语法】

```
HI_S32 HI_GV_App_Destroy(HIGV_HANDLE appHandle);
```

【参数】

参数名称	描述	输入/输出
appHandle	应用程序句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

应用程序句柄必须已创建。

【举例】

参考第 4 章 “开发指引”。

HI_GV_App_Start

【描述】

启动应用程序。

【语法】

```
HI_S32 HI_GV_App_Start(HIGV_HANDLE appHandle);
```

【参数】

参数名称	描述	输入/输出
appHandle	应用程序句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

- 应用程序句柄必须已创建。
- 调用 HI_GV_App_Start 接口后会在 HiGV 内部启动一个 while 循环不停的接收消息和处理消息，直到调用 HI_GV_App_Stop 接口才会停止。

【举例】

参考第 4 章 “开发指引”。

HI_GV_App_Stop

【描述】

停止应用程序消息循环。

【语法】

```
HI_S32 HI_GV_App_Stop(HIGV_HANDLE appHandle);
```

【参数】

参数名称	描述	输入/输出
appHandle	应用句柄。	输入

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

应用程序句柄必须已创建。

【举例】

参考第 4 章 “开发指引”。

HI_GV_App_Active

【描述】

激活应用程序，使得该应用程序可以显示到屏幕上。

【语法】

```
HI_S32 HI_GV_App_Active(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_App_DeActive

【描述】

激活应用后台运行。

【语法】

```
HI_S32 HI_GV_App_DeActive(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_App_IsActive

【描述】

获取应用是否处于前台运行状态。

【语法】

```
HI_BOOL HI_GV_App_IsActive(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
HI_TRUE	前台运行状态。
HI_FALSE	后台运行状态。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_SetMaxAssignHandle

【描述】

设置静态指定的 Handle 的最大值，只能指定一次。

【语法】

```
HI_S32 HI_GV_SetMaxAssignHandle(HIGV_HANDLE maxHandle);
```

【参数】

参数名称	描述	输入/输出
maxHandle	Handle 最大值。	输入

【返回值】

返回值	描述
0	成功。
非 0 值	参考错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_GetReserveStartStaticHandle

【描述】

获取控件构造树保留的静态 handle 起始值。

【语法】

```
HIGV_HANDLE HI_GV_GetReserveStartStaticHandle(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
HIGV_HANDLE	Handle 起始值。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_SetVsyncType

【描述】

设置 VSYNC 的信号源类型。

【语法】

```
HI_S32 HI_GV_SetVsyncType(HIGV_VSYNC_E vsyncType);
```

【参数】

参数名称	描述	输入/输出
vsyncType	vsync 信号源类型。	输入

【返回值】

返回值	描述
0	成功。
非 0 值	参考错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_GetVsyncType

【描述】

获取 VSYNC 的信号源类型。

【语法】

```
HIGV_VSYNC_E HI_GV_GetVsyncType(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
HIGV_VSYNC_HW	硬件 VSYNC 源。
HIGV_VSYNC_SW	软件 VSYNC 源。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_SetSWVsyncPeriod

【描述】

设置软件 VSYNC 的信号周期。

【语法】


```
HI_S32 HI_GV_SetSWVsyncPeriod(HI_S64 period);
```

【参数】

参数名称	描述	输入/输出
period	VSYNC 周期，单位：微秒。	输入

【返回值】

返回值	描述
0	成功。
非 0 值	参考错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

- 在硬件不支持 VSYNC 的时候可以设置软件 VSYNC 周期。
- LCD 屏幕的 VSYNC 周期一般设置为 17*1000us。
- HDMI 屏幕的 VSYNC 周期一般设置为 33*1000us。

【举例】

无。

HI_GV_GetSWVsyncPeriod

【描述】

获取软件 VSYNC 的信号周期。

【语法】

```
HI_S32 HI_GV_GetSWVsyncPeriod(HI_S64* period);
```

【参数】

参数名称	描述	输入/输出
period	VSYNC 周期，单位：微秒。	输出

【返回值】

返回值	描述
0	成功。
非 0 值	参考错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_SetVsyncLostFrame

【描述】

设置 VSYNC 丢帧策略。

【语法】

```
HI_S32 HI_GV_SetVsyncLostFrame(HI_BOOL isLostFrame);
```

【参数】

参数名称	描述	输入/输出
isLostFrame	是否丢帧，丢帧策略设置为 HI_TRUE，不丢帧策略设置为 HI_FALSE。	输入

【返回值】

返回值	描述
0	成功。
非 0 值	参考错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

用于配合 VSYNC 使用，对 UI 滑动流畅性和跟手度要求比较高的场景下可设置为丢帧策略，如设置界面。

【举例】

无。

HI_GV_IsVsyncLostFrame

【描述】

获取 VSYNC 的丢帧策略。

【语法】

```
HI_BOOL HI_GV_IsVsyncLostFrame(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
HI_TRUE	丢帧策略。
HI_FALSE	不丢帧策略。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_SetLostFrameThreshold

【描述】

设置 VSYNC 丢帧的阈值。

【语法】

```
HI_S32 HI_GV_SetLostFrameThreshold(HI_U32 threshold);
```

【参数】

参数名称	描述	输入/输出
threshold	丢帧阈值，单位：微秒。	输入

【返回值】

返回值	描述
0	成功。
非 0 值	参考错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

丢帧阈值和 VSYNC 周期有直接关系，一般可以设置为 VSYNC 周期的二分之一或三分之一。

【举例】

无。

HI_GV_GetLostFrameThreshold

【描述】

获取 VSYNC 丢帧的阈值。

【语法】

```
HI_S32 HI_GV_GetLostFrameThreshold(HI_U32* threshold);
```

【参数】

参数名称	描述	输入/输出
threshold	丢帧阈值，单位：微秒。	输出

【返回值】

返回值	描述
0	成功。
非 0 值	参考错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_RenderCmdSync

【描述】

Render 命令同步接口，执行这个接口会阻塞，直到 Render 命令执行完成。

【语法】

```
HI_S32 HI_GV_RenderCmdSync(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0 值	参考错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

此接口为阻塞接口，调用后会保证异步绘制上下文命令执行完成再返回，用于大小屏切换的场景，在切换之前调用此接口同步命令。

【举例】

无。

HI_GV_SetRefreshCombine

【描述】

设置 Refresh 命令是否合并。

【语法】

```
HI_S32 HI_GV_SetRefreshCombine(HI_BOOL isCombine);
```

【参数】

参数名称	描述	输入/输出
isCombine	是否合并，合并设置为 HI_TRUE，不合并设置为 HI_FALSE。	输入

【返回值】

返回值	描述
0	成功。
非 0 值	参考错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

用于小屏和大屏之间切换的场景，在切换前设置为 HI_FALSE，保证 Refresh 命令不合并，切换完成后再设置为 HI_TRUE。

【举例】

无。

HI_GV_IsRefreshCombine

【描述】

获取 Refresh 命令是否合并状态。

【语法】

```
HI_BOOL HI_GV_IsRefreshCombine(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
HI_TRUE	合并状态。
HI_FALSE	不合并状态。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_SetSyncDraw

【描述】

设置是否进行同步绘制。

【语法】

```
HI_S32 HI_GV_SetSyncDraw(HI_BOOL isSync);
```

【参数】

参数名称	描述	输入/输出
isSync	是否同步，HI_TRUE 表示同步，HI_FALSE 表示异步。	输入

【返回值】

返回值	描述
0	成功。
非 0 值	参考错误码。

【需求】

- 头文件：hi_gv_app.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

设置是否进行同步绘制，即绘制和刷新都使用 UI 线程，而不用单独的绘制线程，默认方式是异步绘制，此接口在 HI_GV_Init 初始化之前调用。

【举例】

无

2.1.1.2 数据类型

相关数据类型、数据结构定义如下：

- PTR_KEYHOOK_CallBack：按键钩子回调函数。
- HIGV_VSYNC_E：VSYNC 信号源枚举。
- HIGV_HANDLE：句柄类型。

PTR_KEYHOOK_CallBack

【说明】

按键钩子回调函数。

【定义】

```
typedef HI_S32 (*PTR_KEYHOOK_CallBack)(HI_U32 msgId, HI_PARAM param1, HI_PARAM param2);
```

【成员】

成员名称	描述
msgId	消息 ID，MSG_KEYDOWN 或 MSG_KEYUP。
param1	消息参数 1，按键值。
param2	消息参数 2，特殊键状态(预留)。

【注意事项】

返回值为 HI_GV_KEYHOOK_GOON 时，消息继续向下传递；返回值为 HI_GV_KEYHOOK_STOP 时，消息不再向下传递。

【相关数据类型及接口】

无。

HIGV_VSYNC_E

【说明】

VSYNC 信号源枚举。

【定义】

```
typedef enum
{
    HIGV_VSYNC_HW = 0,
    HIGV_VSYNC_SW,
    HIGV_VSYNC_BUTT
} HIGV_VSYNC_E;
```

【成员】

成员名称	描述
HIGV_VSYNC_HW	硬件 VSYNC 信号源。
HIGV_VSYNC_SW	软件 VSYNC 信号源。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_HANDLE

【说明】

定义句柄类型。

【定义】

```
typedef unsigned int HIGV_HANDLE;
```

【成员】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

2.1.2 多图层管理

2.1.2.1 接口描述

多图层管理模块提供以下 API：

- HI_GV_Layer_Create：创建图层。
- HI_GV_Layer_CreateEx：创建图层，用于在 XML 中已指定句柄。
- HI_GV_Layer_Destroy：销毁图层。
- HI_GV_Layer_Show：显示/隐藏图层。
- HI_GV_Layer_SetDefault：设置缺省图层。
- HI_GV_Layer_GetDefault：获取缺省图层。
- HI_GV_Layer_GetActiveWindow：获取图层活动窗口句柄。
- HI_GV_Layer_GetHigoLayer：通过 HIGV 图层句柄获取 HIGO 图层句柄。
- HI_GV_Layer_ShareSource：设置两个层同源输出。
- HI_GV_Layer_GetActiveLayer：获取活动图层句柄。
- HI_GV_Layer_SetActiveLayer：设置活动图层句柄。
- HI_GV_Layer_SetRotateMode：设置图形旋转度数。

HI_GV_Layer_Create

【描述】

创建图层。

【语法】

```
HI_S32 HI_GV_Layer_Create(const HIGO_LAYER_INFO_S *layerInfo, HIGV_HANDLE layerHandle);
```

【参数】

参数名称	描述	输入/输出
layerInfo	图层信息。	输入
layerHandle	存放图层句柄信息的指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_mlayer.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

- HIGO_LAYER_INFO_S 结构体类型请参考 HIGO 头文件（hi_go_gdev.h）。
- 开发者调用 HI_GV_Layer_Create 创建的资源，使用结束后，由开发者主动调用 HI_GV_Layer_Destroy 来释放响应资源，否则会有内存泄露。

【举例】

参考第 4 章 “开发指引”。

HI_GV_Layer_CreateEx

【描述】

创建图层，用于在 XML 中已指定句柄。

【语法】

```
HI_S32 HI_GV_Layer_CreateEx(const HIGO_LAYER_INFO_S *layerInfo, HIGV_HANDLE
```



```
layerHandle);
```

【参数】

参数名称	描述	输入/输出
layerInfo	图层信息。	输入
layerHandle	XML 设置的图层句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_mlayer.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

- HIGO_LAYER_INFO_S 结构体类型请参考 HIGO 头文件（hi_go_gdev.h）。
- 开发者调用 HI_GV_Layer_CreateEx 创建的资源，使用结束后，由开发者主动调用 HI_GV_Layer_Destroy 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_Layer_Destroy

【描述】

销毁图层。

【语法】

```
HI_S32 HI_GV_Layer_Destroy(HIGV_HANDLE layerHandle);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_mlayer.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

layerHandle 必须为已存在的句柄。

【举例】

参考第 4 章 “开发指引”。

HI_GV_Layer_Show

【描述】

显示或隐藏图层。

【语法】

```
HI_S32 HI_GV_Layer_Show(HIGV_HANDLE layerHandle, HI_BOOL visible);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入
visible	图层是否可见。 HI_TRUE：显示；	输入

参数名称	描述	输入/输出
	HI_FALSE：表示隐藏。	

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_mlayer.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

layerHandle 必须为已设置存在的句柄。

【举例】

无。

HI_GV_Layer_SetDefault

【描述】

设置缺省图层，设置缺省图层后，如果创建窗口不指定图层，则默认为此图层。

【语法】

```
HI_S32 HI_GV_Layer_SetDefault(HIGV_HANDLE layerHandle);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_mlayer.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

layerHandle 必须为已设置存在的句柄。

【举例】

无。

HI_GV_Layer_GetDefault

【描述】

获取缺省图层。

【语法】

```
HI_S32 HI_GV_Layer_GetDefault(HIGV_HANDLE* defaultLayer);
```

【参数】

参数名称	描述	输入/输出
defaultLayer	默认图层句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_mlayer.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Layer_GetActiveWindow

【描述】

获取图层活动窗口句柄。

【语法】

```
HI_S32 HI_GV_Layer_GetActiveWindow(HIGV_HANDLE layerHandle, HIGV_HANDLE *window);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入
window	活动窗口句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_mlayer.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

layerHandle 必须为已设置存在的句柄。

【举例】

无。

HI_GV_Layer_GetHigoLayer

【描述】

通过 HIGV 图层句柄获取 HIGO 图层句柄。

【语法】

```
HI_S32 HI_GV_Layer_GetHigoLayer(HIGV_HANDLE layerHandle, HIGV_HANDLE *higoLayer);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入
higoLayer	HiGo 图层句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_mlayer.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

layerHandle 必须为已设置存在的句柄。

【举例】

无。

HI_GV_Layer_ShareSource

【描述】

设置两个层同源输出。

【语法】

```
HI_S32 HI_GV_Layer_ShareSource(HIGV_HANDLE ownerLayerHandle, HIGV_HANDLE clientLayerHandle);
```

【参数】

参数名称	描述	输入/输出
ownerrLayerHandle	已经拥有输出内容的 layer。	输入
clientLayerHandle	需要与 ownerLayerHandle 共享源的 layer。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_mlayer.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Layer_GetActiveLayer

【描述】

获取活动图层句柄。

【语法】

```
HI_S32 HI_GV_Layer_GetActiveLayer(HIGV_HANDLE *layerHandle);
```

【参数】

参数名称	描述	输入/输出
layerHandle	活动图层句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_mlayer.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Layer_SetActiveLayer

【描述】

设置活动图层句柄。

【语法】


```
HI_S32 HI_GV_Layer_SetActiveLayer(HIGV_HANDLE layerHandle);
```

【参数】

参数名称	描述	输入/输出
layerHandle	活动图层句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_mlayer.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Layer_SetRotateMode

【描述】

设置图形旋转度数。

【语法】

```
HI_S32 HI_GV_Layer_SetRotateMode(HIGV_HANDLE layerHandle, HIGV_ROTATE_E rotate);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入

参数名称	描述	输入/输出
rotate	顺时针旋转度数。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_mlayer.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

- layerHandle 必须为已设置存在的句柄。
- Hi3559AV100 芯片可以支持像素格式 HIGO_PF_8888、HIGO_PF_1555 和 HIGO_PF_4444 做旋转。

【举例】

参考第 4 章 “开发指引”。

2.1.2.2 数据类型

相关数据类型、数据结构定义如下：

HIGV_ROTATE_E：定义图层顺时针旋转度数枚举。

HIGV_ROTATE_E

【说明】

定义图层顺时针旋转度数枚举。

【定义】

```
typedef enum
{
```



```
    HIGV_ROTATE_NONE = 0,
    HIGV_ROTATE_90,
    HIGV_ROTATE_180,
    HIGV_ROTATE_270,
    HIGV_ROTATE_BUTT
} HIGV_ROTATE_E;
```

【成员】

成员名称	描述
HIGV_ROTATE_NONE	无旋转。
HIGV_ROTATE_90	顺时针旋转 90 度。
HIGV_ROTATE_180	顺时针旋转 180 度。
HIGV_ROTATE_270	顺时针旋转 270 度。

【注意事项】

无。

【相关数据类型及接口】

无。

2.1.3 控件管理

2.1.3.1 接口描述

控件管理模块提供以下 API：

- HI_GV_Widget_Active：设置控件焦点为活动的。
- HI_GV_Widget_Create：创建控件。
- HI_GV_Widget_Destroy：销毁控件。
- HI_GV_Widget_Enable：使能控件，对窗口无效。
- HI_GV_Widget_GetActiveWidget：获取活动子窗口。
- HI_GV_Widget_GetFont：获取字体。

- HI_GV_Widget_GetParent：获取当前控件的父控件，如果当前控件是窗口，则父为 NULL。
- HI_GV_Widget_GetSiblings：获取控件的左右上下兄弟控件。
- HI_GV_Widget_Hide：隐藏控件。
- HI_GV_Widget_IsActive：判断控件是否为当前的焦点控件。
- HI_GV_Widget_IsEnable：判断控件是否使能。
- HI_GV_Widget_IsShow：判断控件是否显示。
- HI_GV_Widget_IsRefresh：判断控件是否绘制完成（目前仅支持判断窗口）。
- HI_GV_Widget_GetSurface：获取控件的 Surface。
- HI_GV_Widget_Paint：异步绘制该控件。
- HI_GV_Widget_SetFont：设置字体。
- HI_GV_Widget_SetMsgProc：注册消息处理函数。
- HI_GV_Widget_SetSkin：设置控件皮肤。
- HI_GV_Widget_SetSiblings：设置控件的兄弟。
- HI_GV_Widget_Show：显示控件。
- HI_GV_Widget_Move：移动控件。
- HI_GV_Widget_MoveToTop：把当前窗口置顶，对非窗口控件无效。
- HI_GV_Widget_Highlight：高亮控件。
- HI_GV_Widget_IsHighlight：控件是否高亮。
- HI_GV_Widget_SetTransparent：设置控件是否透明。
- HI_GV_Widget_SetAlign：设置文本对齐方式。
- HI_GV_Widget_SetText：设置文本。
- HI_GV_Widget_SetTextByID：通过字符串 ID 设置文本。
- HI_GV_Widget_GetText：获取控件文本。
- HI_GV_Widget_GetTextID：获取控件文本 ID。
- HI_GV_Widget_BindDataSource：绑定数据模型。
- HI_GV_Widget_UnbindDataSource：解绑数据模型。
- HI_GV_Widget_GetDataSource：获取数据源句柄。
- HI_GV_Widget_GetWindow：获取控件所在主窗口句柄。
- HI_GV_Widget_Update：及时绘制控件。

- HI_GV_Widget_Refresh：将控件指定区域立即显示在屏幕上。
- HI_GV_Widget_SyncDB：从数据源获取数据，通知控件显示最新的数据。
- HI_GV_Widget_GetRect：获取控件的区域，坐标相对于父控件；若当前控件是窗口，则相对于所属图层。
- HI_GV_Widget_SetWinColorKey：设置窗口的 ColorKey。
- HI_GV_Widget_IsWindow：控件是否是窗口。
- HI_GV_Widget_Depaint：取消绘制。
- HI_GV_Widget_Screen2Widget：将屏幕坐标转换为控件坐标。
- HI_GV_Widget_Widget2Screen：将控件坐标转换为屏幕坐标。
- HI_GV_Widget_GetWidgetByPos：获取坐标所在顶层控件。
- HI_GV_Widget_GetWidgetByPos_TouchDevice：获取坐标所在顶层控件，适用于触摸屏设备。
- HI_GV_Widget_BindScrollBar：绑定滚动条。
- HI_GV_Widget_GetScrollBar：获取该控件绑定的 Scrollbar。
- HI_GV_Widget_SetAllCheckStatus：将该控件下所有 CheckBox 设置为选中或非选中状态。
- HI_GV_Widget_Resize：改变窗口大小，对控件无效。
- HI_GV_Widget_IsNeedIMEWindow：查询目标控件是否需要弹出输入法窗口。
- HI_GV_Widget_GetLayer：获取 Widget 所在图层句柄。
- HI_GV_Widget_SetPrivate：设置控件的私有数据。
- HI_GV_Widget_GetPrivate：获取控件的私有数据。
- HI_GV_Widget_GetCloneHandle：获取指定图层控件的克隆句柄。
- HI_GV_Widget_EraseBackground：设置控件是否擦除背景。
- HI_GV_Widget_RgisterWidget：注册控件类型信息。
- HI_GV_Widget_UnRgisterWidget：反注册控件信息。+
- HI_GV_Widget_CanActive：为弹出框增加的接口，即使有模态窗口，模态窗口上的弹出框窗口也可以激活。
- HI_GV_Widget_EnableMirror：使能是否允许镜像控件。
- HI_GV_Widget_GetMirrorAttr：获取镜像控件属性。
- HI_GV_Widget_SetTextDir：强制控件文本的方向。
- HI_GV_Widget_SetFocusMode：设置控件焦点切换模式。

- HI_GV_Widget_SetMargin：设置文字或图片显示的边缘间隔。
- HI_GV_Widget_GetMargin：获取边缘间隔。
- HI_GV_Widget_EnableResPach：使能配置图片资源的相对路径。
- HI_GV_Widget_ActiveShow：显示并激活控件。
- HI_GV_Widget_SetSkinBlitOpt：设置控件图片类皮肤的搬移混合操作运算。

HI_GV_Widget_Active

【描述】

设置控件焦点为活动的，对不可设置为焦点的控件无效。

【语法】

```
HI_S32 HI_GV_Widget_Active(HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

参考第 4 章 “开发指引”。

HI_GV_Widget_Create

【描述】

创建控件。

【语法】

```
HI_S32 HI_GV_Widget_Create(const HIGV_WCREATE_S *creatInfo, HIGV_HANDLE *
widgetHandle);
```

【参数】

参数名称	描述	输入/输出
createInfo	控件信息结构体指针。	输入
widgetHandle	控件句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

- 开发者调用 HI_GV_Widget_Create 创建的资源，使用结束后，由开发者主动调用 HI_GV_Widget_Destroy 来释放响应资源，否则会有内存泄露。
- 接口创建控件后，要显示出来，需要调用手动调用 HI_GV_Widget_Show、HI_GV_Widget_Paint 或者 HI_GV_Widget_Active 接口来显示、或者激活控件。

【举例】

无。

HI_GV_Widget_Destroy

【描述】

销毁控件，若销毁的是容器类控件，则还会销毁该容器内的所有子控件。

【语法】

```
HI_S32 HI_GV_Widget_Destroy(HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_Enable

【描述】

使能控件，对窗口无效。

【语法】

```
HI_S32 HI_GV_Widget_Enable(HIGV_HANDLE widgetHandle, HI_BOOL enable);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
enable	使能标志。 HI_TRUE：使能； HI_FALSE：去使能。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetActiveWidget

【描述】

获取活动子窗口或容器类控件内的焦点控件。

【语法】

```
HI_S32 HI_GV_Widget_GetActiveWidget(HIGV_HANDLE parentHandle, HIGV_HANDLE*
activeWidgetHandle);
```

【参数】

参数名称	描述	输入/输出
parentHandle	父控件句柄。	输入
activeWidgetHandle	活动子控件指针句柄或焦点控件句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetFont

【描述】

获取字体。

【语法】

```
HI_S32 HI_GV_Widget_GetFont(HIGV_HANDLE widgetHandle, HIGV_HANDLE *fontHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
fontHandle	字体对象指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetParent

【描述】

获取当前控件的父控件，如果当前控件是窗口，则父为 NULL。

【语法】

```
HI_S32    HI_GV_Widget_GetParent(HIGV_HANDLE widgetHandle, HIGV_HANDLE*
parentHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
parentHandle	父控件句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetSiblings

【描述】

获取该控件的上下左右兄弟控件，用于焦点切换场景，通过方向按键切换焦点。

【语法】


```
HI_S32 HI_GV_Widget_GetSiblings(HIGV_HANDLE widgetHandle, HIGV_HANDLE*
leftSibingHandle, HIGV_HANDLE* rightSibingHandle, HIGV_HANDLE* upSibingHandle,
HIGV_HANDLE* downSibingHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
leftSibingHandle	左邻居控件句柄指针。	输出
rightSibingHandle	右邻居控件句柄指针。	输出
upSibingHandle	上邻居控件句柄指针。	输出
downSibingHandle	下邻居控件句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_Hide

【描述】

隐藏控件。

【语法】

```
HI_S32 HI_GV_Widget_Hide(HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

隐藏焦点控件前先调用 HI_GV_Widget_Active 接口将焦点切换至下一个场景。

【举例】

无。

HI_GV_Widget_IsActive

【描述】

判断控件是否为当前的焦点控件。

【语法】

```
HI_BOOL HI_GV_Widget_IsActive(HIGV_HANDLE widgetHandle);
```


【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

【返回值】

返回值	描述
HI_TRUE	控件是活动控件。
HI_FALSE	控件不是活动控件。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_IsEnable

【描述】

获取控件使能状态，对窗口无效。

【语法】

```
HI_BOOL HI_GV_Widget_IsEnable(HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

【返回值】

返回值	描述
HI_TRUE	使能状态。
HI_FALSE	禁用状态。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_IsShow

【描述】

判断控件是否显示。

【语法】

```
HI_BOOL HI_GV_Widget_IsShow(HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

【返回值】

返回值	描述
HI_TRUE	显示状态。
HI_FALSE	隐藏状态或者非有效控件 handle。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_IsRefresh

【描述】

判断控件是否绘制刷新完成。（目前仅支持判断窗口）。

【语法】

```
HI_BOOL HI_GV_Widget_Is Refresh (HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

【返回值】

返回值	描述
HI_TRUE	绘制完成状态。
HI_FALSE	非绘制完成状态或者非有效的窗口 handle。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

目前仅支持窗口控件。

【举例】

无。

HI_GV_Widget_GetSurface

【描述】

获取控件的 Surface。

【语法】

```
HI_S32    HI_GV_Widget_GetSurface(HIGV_HANDLE widgetHandle, HIGV_HANDLE* surfaceHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
surfaceHandle	Surface 句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_Paint

【描述】

异步绘制该控件，产生 HIGV_MSG_PAINT 消息。

【语法】

```
HI_S32    HI_GV_Widget_Paint(HIGV_HANDLE widgetHandle, const HI_RECT* rect);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
rect	绘制更新的区域指针，为 HI_NULL 时绘制整个控件。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h

- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_SetFont

【描述】

设置字体。

【语法】

```
HI_S32    HI_GV_Widget_SetFont(HIGV_HANDLE widgetHandle, HIGV_HANDLEfontHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
fontHandle	控件字体对象。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_SetMsgProc

【描述】

注册控件信息处理函数。

【语法】

```
HI_S32    HI_GV_Widget_SetMsgProc(HIGV_HANDLE widgetHandle, HI_U32 msg,
HIGV_MSG_PROC customProc,HIGV_PROCORDER_E procOrder);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
msg	消息。	输入
customProc	消息处理函数。	输入
procOrder	消息处理优先级。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_SetSkin

【描述】

设置控件皮肤。

【语法】

```
HI_S32  HI_GV_Widget_SetSkin(HIGV_HANDLE widgetHandle, HI_U32 skinIndexHandle,
HI_RESID skinHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
skinIndexHandle	皮肤状态索引。	输入
skinHandle	皮肤资源句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_SetSiblings

【描述】

设置控件兄弟关系，可通过方向按键将焦点切换至对应的兄弟控件。

【语法】

```
HI_S32    HI_GV_Widget_SetSiblings(HIGV_HANDLE widgetHandle,
                                   HIGV_HANDLE    leftHandle,
                                   HIGV_HANDLE    rightHandle,
                                   HIGV_HANDLE    upHandle,
                                   HIGV_HANDLE    downHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
leftHandle	左邻居控件句柄。	输入
rightHandle	右邻居控件句柄。	输入
upHandle	上邻居控件句柄。	输入
downHandle	下邻居控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_Show

【描述】

显示控件。

【语法】

```
HI_S32    HI_GV_Widget_Show(HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

操作已显示的控件不会产生重绘。

【举例】

无。

HI_GV_Widget_Move

【描述】

修改控件在父容器中的相对位置，自动重绘。

【语法】

```
HI_S32    HI_GV_Widget_Move(HIGV_HANDLE widgetHandle, HIGV_CORD x, HIGV_CORD y);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
x	移动后的 x 坐标，单位：像素。	输入
y	移动后的 y 坐标，单位：像素。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_MoveToTop

【描述】

当前窗口置顶，对控非窗口件无效。

【语法】

```
HI_S32   HI_GV_Widget_MoveToTop(HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_Highlight

【描述】

高亮控件。

【语法】

```
HI_S32    HI_GV_Widget_Highlight(HIGV_HANDLE widgetHandle, HI_BOOL highlight);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
highlight	高亮标志。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_IsHighlight

【描述】

控件是否高亮。

【语法】

```
HI_BOOL HI_GV_Widget_IsHighlight(HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

【返回值】

返回值	描述
HI_TRUE	高亮。
HI_FALSE	非高亮。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_SetTransparent

【描述】

设置控件的透明风格。

【语法】

```
HI_S32 HI_GV_Widget_SetTransparent(HIGV_HANDLE widgetHandle, HI_BOOL isTrans);
```


【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
isTrans	透明标志。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_SetAlign

【描述】

设置文本对齐方式。

【语法】

```
HI_S32    HI_GV_Widget_SetAlign(HIGV_HANDLE widgetHandle, HI_U32 align);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
align	文本对齐方式。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_SetText

【描述】

设置控件显示的文本字串。

【语法】

```
HI_S32    HI_GV_Widget_SetText(HIGV_HANDLE widgetHandle, const HI_CHAR* text);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

参数名称	描述	输入/输出
text	控件文本内容指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

- 只作用于可显示文字且不可绑定数据模型的非编辑类控件。
- 目前只支持 UTF-8 字符集编码。

【举例】

无。

HI_GV_Widget_SetTextByID

【描述】

设置控件用于显示文字的多语言字串 ID。

【语法】

```
HI_S32 HI_GV_Widget_SetTextByID(HIGV_HANDLE widgetHandle, const HI_U32 strID);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

参数名称	描述	输入/输出
strID	控件文本 ID。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

只作用于可显示文字且不可绑定数据模型的非编辑类控件。

【举例】

无。

HI_GV_Widget_GetText

【描述】

获取当前语言环境下显示的文本字串。

【语法】

```
HI_S32    HI_GV_Widget_GetText(HIGV_HANDLE widgetHandle, HI_CHAR* buffer, HI_U32 bufferLen);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

参数名称	描述	输入/输出
buffer	控件文本内容指针。	输出
bufferLen	输出 Buffer 长度。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetTextID

【描述】

获取控件显示文本的多语言字串 ID。

【语法】

```
HI_S32    HI_GV_Widget_GetTextID(HIGV_HANDLE widgetHandle, HI_U32* strID);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

参数名称	描述	输入/输出
strID	输出文本 ID 指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_BindDataSource

【描述】

绑定数据模型。

【语法】

```
HI_S32    HI_GV_Widget_BindDataSource(HIGV_HANDLE widgetHandle, HIGV_HANDLE dataSource);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

参数名称	描述	输入/输出
dataSource	数据模型句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

只作用于可通过数据模型获取数据并显示控件。

【举例】

无。

HI_GV_Widget_UnbindDataSource

【描述】

解绑数据模型。

【语法】

```
HI_S32    HI_GV_Widget_UnbindDataSource(HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetDataSource

【描述】

获取已绑定的数据模型句柄。

【语法】

```
HI_S32    HI_GV_Widget_GetDataSource(HIGV_HANDLE widgetHandle, HIGV_HANDLE* dataSource);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
dataSource	数据源句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetWindow

【描述】

获取控件所在的父窗口句柄。

【语法】

```
HI_S32    HI_GV_Widget_GetWindow(HIGV_HANDLE widgetHandle, HIGV_HANDLE* windowHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
windowHandle	父窗口句柄。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_Update

【描述】

即时重绘控件，不产出 HIGV_MSG_PAINT 消息。

【语法】

```
HI_S32    HI_GV_Widget_Update(HIGV_HANDLE widgetHandle, const HI_RECT* rect);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
rect	刷新区域指针，为 HI_NULL 时绘制整个控件。	输入

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_Refresh

【描述】

将控件的指定区域立即显示在屏幕上，只作用于窗口。

【语法】

```
HI_S32    HI_GV_Widget_Refresh(HIGV_HANDLE widgetHandle, const HI_RECT* refreshRect);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
refreshRect	刷新区域指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_SyncDB

【描述】

从数据源获取数据，通知控件显示最新的数据。

【语法】

```
HI_S32    HI_GV_Widget_SyncDB(HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h

- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetRect

【描述】

获取控件的区域，坐标相对于父控件；若当前控件是窗口，则相对于所属图层。

【语法】

```
HI_S32    HI_GV_Widget_GetRect(HIGV_HANDLE widgetHandle, HI_RECT* rect);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
rect	控件区域范围指针，由 x、y、w、h 组成，x、y 表示控件左上角的坐标位置，w、h 表示控件的宽高。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_SetWinColorKey

【描述】

设置窗口 Colorkey。

【语法】

```
HI_S32  HI_GV_Widget_SetWinColorKey(HIGV_HANDLE windowHandle, HI_COLOR colorKey);
```

【参数】

参数名称	描述	输入/输出
windowHandle	控件句柄。	输入
colorKey	Colorkey 值。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_IsWindow

【描述】

判断控件是否为窗口。

【语法】

```
HI_BOOL HI_GV_Widget_IsWindow(HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

【返回值】

返回值	描述
HI_TRUE	窗口。
HI_FALSE	非窗口。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_Depaint

【描述】

取消绘制。

【语法】

```
HI_S32    HI_GV_Widget_Depaint(HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_Screen2Widget

【描述】

将屏幕相对坐标转换为控件相对坐标，默认图层为整个屏幕。

【语法】

```
HI_S32  HI_GV_Widget_Screen2Widget(HIGV_HANDLE widgetHandle, HIGV_CORD screenX, HIGV_CORD screenY, HIGV_CORD* widgetX, HIGV_CORD* widgetY);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
screenX	屏幕坐标 X，单位：像素。	输入
screenY	屏幕坐标 Y，单位：像素。	输入
widgetX	控件坐标 X 指针，单位：像素。	输出
widgetY	控件坐标 Y 指针，单位：像素。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_Widget2Screen

【描述】

将控件坐标转为屏幕坐标。

【语法】

```
HI_S32  HI_GV_Widget_Widget2Screen(HIGV_HANDLE widgetHandle, HIGV_CORD widgetX, HIGV_CORD widgetY, HIGV_CORD* screenX, HIGV_CORD* screenY);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
widgetX	控件坐标 X，单位：像素。	输入
widgetY	控件坐标 Y，单位：像素。	输入
screenX	屏幕坐标 X 指针，单位：像素。	输出
screenY	屏幕坐标 Y 指针，单位：像素。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetWidgetByPos

【描述】

获取坐标所在顶层控件。

【语法】

```
HI_S32    HI_GV_Widget_GetWidgetByPos(HIGV_CORD screenX, HIGV_CORD screenY,
HIGV_HANDLE* widgetHandle);
```

【参数】

参数名称	描述	输入/输出
screenX	控件坐标 X，单位：像素。	输入
screenY	控件坐标 Y，单位：像素。	输入
widgetHandle	获取的屏幕顶层控件。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetWidgetByPos_TouchDevice

【描述】

获取坐标所在顶层控件，适用于触摸屏设备。

【语法】

```
HI_S32  HI_GV_Widget_GetWidgetByPos_TouchDevice(HIGV_CORD screenX, HIGV_CORD screenY, HIGV_HANDLE* widgetHandle);
```

【参数】

参数名称	描述	输入/输出
screenX	控件坐标 X，单位：像素。	输入
screenY	控件坐标 Y，单位：像素。	输入
widgetHandle	获取的屏幕顶层控件指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_BindScrollBar

【描述】

控件绑定滚动条。

【语法】

```
HI_S32    HI_GV_Widget_BindScrollBar(HIGV_HANDLE widgetHandle, HIGV_HANDLE scrollBarHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
scrollBarHandle	滚动条控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetScrollBar

【描述】

获取该控件绑定的 ScrollBar。

【语法】

```
HI_S32 HI_GV_Widget_GetScrollBar(HIGV_HANDLE widgetHandle, HIGV_HANDLE*
scrollBarHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
scrollBarHandle	绑定的 ScrollBar 句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_SetAllCheckStatus

【描述】

设置容器内所有 HIGV_BS_CHECK 风格的 button 控件的选中状态。

【语法】


```
HI_S32 HI_GV_Widget_SetAllCheckStatus(HIGV_HANDLE parentHandle,
HIGV_BUTTON_STATUS_E status);
```

【参数】

参数名称	描述	输入/输出
parentHandle	父控件句柄。	输入
status	状态。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_Resize

【描述】

改变窗口大小，只对窗口有效。

【语法】

```
HI_S32    HI_GV_Widget_Resize(HIGV_HANDLE widgetHandle,
                              HI_S32 width,
                              HI_S32 height);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
width	改变后的窗口宽度，单位：像素。	输入
height	改变后的窗口高度，单位：像素。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_IsNeedIMEWindow

【描述】

查询目标控件是否需要弹出输入法窗口。

【语法】

```
HI_BOOL HI_GV_Widget_IsNeedIMEWindow(HIGV_HANDLE widgetHandle, HI_U32* supportIMEType);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
supportIMEType	支持的输入法类型指针。	输入

【返回值】

返回值	描述
HI_TRUE	需要。
HI_FALSE	不需要。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetLayer

【描述】

获取窗口所属图层。

【语法】

```
HI_S32 HI_GV_Widget_GetLayer(HIGV_HANDLE widgetHandle, HIGV_HANDLE* layerHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

参数名称	描述	输入/输出
layerHandle	图层句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_SetPrivate

【描述】

设置 Widget 的私有数据。

【语法】

```
HI_S32 HI_GV_Widget_SetPrivate(HIGV_HANDLE widgetHandle, const HI_VOID* privateData);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
privateData	私有数据指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetPrivate

【描述】

获取 Widget 的私有数据。

【语法】

```
HI_S32 HI_GV_Widget_GetPrivate(HIGV_HANDLE widgetHandle, HI_VOID** privateData);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
privateData	私有数据指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetCloneHandle

【描述】

获取指定图层控件的克隆句柄。

【语法】

```
HI_S32 HI_GV_Widget_GetCloneHandle(HIGV_HANDLE widgetHandle, HIGV_HANDLE layerHandle, HIGV_HANDLE* cloneHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
layerHandle	图层句柄。	输入
cloneHandle	克隆的句柄。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

默认情况该函数功能不支持，需要打开 WIDGET_CLONE_SUPPORT 使用。

【举例】

无。

HI_GV_Widget_EraseBackground

【描述】

设置控件是否擦除背景。

【语法】

```
HI_S32 HI_GV_Widget_EraseBackground(HIGV_HANDLE widgetHandle, HI_BOOL isErasebg);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
isErasebg	是否擦除背景。	输入

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

窗口共享模式默认是不擦除背景。

【举例】

无。

HI_GV_Widget_RgisterWidget

【描述】

注册控件类型信息，用户可以通过该接口注册自定义的控件类型信息。

【语法】

```
HI_S32 HI_GV_Widget_RgisterWidget(HIGV_WIDGET_TYPEINFO_S typeInfo);
```

【参数】

参数名称	描述	输入/输出
typeInfo	控件类型信息。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_UnRgisterWidget

【描述】

反注册控件信息。

【语法】

```
HI_S32 HI_GV_Widget_UnRgisterWidget(HI_U32 typeID);
```

【参数】

参数名称	描述	输入/输出
typeID	自定义控件类型 ID。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_CanActive

【描述】

为弹出框增加的接口，即使有模态窗口，模态窗口上的弹出框窗口也可以激活。

【语法】

```
HI_S32 HI_GV_Widget_CanActive(HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

慎重使用该接口，否则模态窗口将失去意义。

【举例】

无。

HI_GV_Widget_EnableMirror

【描述】

使能是否允许镜像控件。

【语法】

```
HI_S32 HI_GV_Widget_EnableMirror(HIGV_HANDLE widgetHandle, HI_BOOL posMirror,
HI_BOOL interiorMirror);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
posMirror	是否允许切换语言时在父容器中镜像。	输入
interiorMirror	是否允许切换语言时在控件内部镜像。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetMirrorAttr

【描述】

获取镜像控件属性。

【语法】

```
HI_S32 HI_GV_Widget_GetMirrorAttr(HIGV_HANDLE widgetHandle, HI_BOOL* posMirror,
HI_BOOL* interiorMirror);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
posMirror	是否允许切换语言时在父容器中镜像。	输出
interiorMirror	是否允许切换语言时在控件内部镜像。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_SetTextDir

【描述】

强制控件文本的方向。

【语法】

```
HI_S32 HI_GV_Widget_SetTextDir(HIGV_HANDLE widgetHandle, HIGV_TEXTDIR_E direction);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
direction	文本的方向。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_SetFocusMode

【描述】

设置控件焦点切换模式。

【语法】

```
HI_S32 HI_GV_Widget_SetFocusMode(HIGV_HANDLE widgetHandle, HIGV_FOCUS_SWITCH_E focusMode);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
focusMode	焦点切换模式	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_SetMargin

【描述】

设置文字或图片显示的边缘间隔。

【语法】


```
HI_S32 HI_GV_Widget_SetMargin(HIGV_HANDLE widgetHandle, HI_U32 leftMargin, HI_U32 rightMargin, HI_U32 topMargin, HI_U32 bottomMargin);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
leftMargin	左边缘间隔。	输入
rightMargin	右边缘间隔。	输入
topMargin	上边缘间隔。	输入
bottomMargin	下边缘间隔。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_GetMargin

【描述】

获取边缘间隔。

【语法】

```
HI_S32 HI_GV_Widget_GetMargin(HIGV_HANDLE widgetHandle, HI_U32* leftMargin,  HI_U32* rightMargin, HI_U32* topMargin, HI_U32* bottomMargin);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
leftMargin	左边缘间隔。	输出
rightMargin	右边缘间隔。	输出
topMargin	上边缘间隔。	输出
bottomMargin	下边缘间隔。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_EnableResPach

【描述】

使能配置图片资源的相对路径。

【语法】

```
HI_S32 HI_GV_Widget_EnableResPach(HIGV_HANDLE widgetHandle, HI_BOOL enable);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
enable	使能开关。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_ActiveShow

【描述】

显示并激活控件。

【语法】

```
HI_S32 HI_GV_Widget_ActiveShow(HIGV_HANDLE widgetHandle, HI_BOOL activeShow);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
activeShow	HI_TRUE：使能； HI_FALSE：非使能。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Widget_SetSkinBlitOpt

【描述】

设置控件图片类皮肤的搬移混合操作运算。

【语法】

```
HI_S32 HI_GV_Widget_SetSkinBlitOpt(HIGV_HANDLE widgetHandle, const HIGO_BLTOPT_S* blitOpt);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
blitOpt	搬移混合操作运算属性。 详情请参考《HiGO API 参考文档》。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_widget.h、higo.h、hi_gv_conf.h、hi_gv_errno.h、hi_gv_resm.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

请参考 higo 头文件或相关文档设置混合运算参数。

【举例】

无。

2.1.3.2 数据类型

相关数据类型、数据结构定义如下：

- HIGV_WIDGET_TYPEINFO_S：控件类型信息。
- HIGV_CORD：HIGV 坐标类型。
- HIGV_TEXTDIR_E：强制控件语言方向。
- HIGV_FOCUS_SWITCH_E：焦点切换模式。
- HIGV_PROCORDER_E：标识同一个事件消息用户注册的事件函数与控件原始的事件函数的处理先后顺序。
- HIGV_WCREATE_S：控件创建属性。

- WidgetBasicParam：控件基础属性。
- HI_RECT：矩形区域。

HIGV_WIDGET_TYPEINFO_S

【说明】

控件类型信息。

【定义】

```
typedef struct hiHIGV_WIDGET_TYPEINFO_S
{
    HI_CHAR    Name[33];
    HI_U32 TypeID;
            HIGV_WIDGET_INIT_FUNC WidgetInitFunc;
            HI_U32 WidgetPaserSize;
} HIGV_WIDGET_TYPEINFO_S;
```

【成员】

成员名称	描述
Name[33]	控件名。
TypeID	TypeID 应从 HIGV_EXTWIDGET_START 开始编号。
WidgetInitFunc	用户自定义控件创建回调函数。
WidgetPaserSize	控件解析模块所定义结构的大小，如果该控件已在 XML 中解析创建则需要设置，否则可设为 0。

【注意事项】

HIGV_WIDGET_INIT_FUNC 是一个回调函数，定义为：

```
typedef HI_VOID*    (*HIGV_WIDGET_INIT_FUNC)( const WidgetBasicParam
*widgetBasicParam)。
```

【相关数据类型及接口】

WidgetBasicParam

HIGV_CORD

【说明】

HiGV 坐标类型。

【定义】

```
typedef HI_S32 HIGV_CORD;
```

【成员】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_TEXTDIR_E

【说明】

强制控件语言方向。

【定义】

```
typedef enum
{
    HIGV_TEXTDIR_NEUTRAL = 0,
    HIGV_TEXTDIR_LTR,
    HIGV_TEXTDIR_RTL,
    HIGV_TEXTDIR_BUTT
 } HIGV_TEXTDIR_E;
```

【成员】

成员名称	描述
HIGV_TEXTDIR_NEUTRAL	默认值。
HIGV_TEXTDIR_LTR	从左到右。

成员名称	描述
HIGV_TEXTDIR_RTL	从右到左。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_FOCUS_SWITCH_E

【说明】

焦点切换模式。

【定义】

```
typedef enum
{
    HIGV_FOCUS_SWITCH_AUTO = 0,
    HIGV_FOCUS_SWITCH_MANUAL,
    HIGV_FOCUS_SWITCH_BUTT
 } HIGV_FOCUS_SWITCH_E;
```

【成员】

成员名称	描述
HIGV_FOCUS_SWITCH_AUTO	焦点自动切换模式。
HIGV_SWITCH_MANUAL	焦点手动切换模式。

【注意事项】

只针对按键操作，手动切换模式需要按 ENTER 进入或退出控件的操作编辑状态。

【相关数据类型及接口】

无。

HIGV_PROCORDER_E

【说明】

标识同一个事件消息用户注册的事件函数与控件原始的事件函数的处理先后顺序。

【定义】

```
typedef enum
{
    HIGV_PROCORDER_BEFORE,
    HIGV_PROCORDER_AFTER,
    HIGV_PROCORDER_BUTT
} HIGV_PROCORDER_E;
```

【成员】

成员名称	描述
HIGV_PROCORDER_BEFORE	用户注册的事件函数先调用。
HIGV_PROCORDER_AFTER	用户注册的事件函数后调用。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_WCREATE_S

【说明】

控件创建属性。

【定义】

```
typedef struct hiHIGV_WCREATE_S
{
    HI_U32          type;
    HI_RECT         rect;
    HIGV_HANDLE     hParent;
    HI_U32          style;
```



```
    HI_U32          level;
    HI_VOID*        pPrivate;
} HIGV_WCREATE_S;
```

【成员】

成员名称	描述
type	控件类型。
rect	控件位置及大小。
hParent	父控件句柄。
style	控件风格。
level	窗口层次，只针对窗口生效，范围为[0, 15]。
pPrivate	各 Widget 创建需要的私有数据。

【注意事项】

- type 为控件类型，定义在头文件 hi_gv_widget.h 中的 HIGV_WIDGET_E 枚举类型。
- style 为控件风格，定义在头文件 hi_gv_widget.h 中 HIGV_STYLE_DEFAULT 等宏定义。

【相关数据类型及接口】

无。

WidgetBasicParam

【说明】

控件创建属性。

【定义】

```
typedef struct {
    HI_U32 style;
    const HI_RECT *rect;
    HIGV_HANDLE parentHandle;
    HI_U32 level;
```



```
} WidgetBasicParam;
```

【成员】

成员名称	描述
type	控件类型。
rect	控件位置及大小。
parentHandle	父控件句柄。
level	窗口层次，只针对窗口生效，范围为[0, 15]。

【相关数据类型及接口】

无。

HI_RECT

【说明】

矩形区域。

【定义】

```
typedef struct
{
    HI_S32 x, y;
    HI_S32 w, h;
} HI_RECT;
```

【相关数据类型及接口】

无。

2.1.4 资源管理

2.1.4.1 接口描述

资源管理模块提供以下 API：

- HI_GV_Res_CreateID：创建资源 ID。
- HI_GV_Res_CreateID_NoPrefixPath：创建资源 ID，不读取资源前缀环境变量。

- HI_GV_Res_DisablePrefixPath：强制应用不使用资源路径的环境变量。
- HI_GV_Res_DestroyID：销毁资源句柄。
- HI_GV_Res_CreateStyle：生成 Style。
- HI_GV_Res_CreateStyleByHandle：根据指定的 RESID 生成 Style。
- HI_GV_Res_DestroyStyle：销毁 Style。
- HI_GV_Res_SetResident：设置资源常驻内存。
- HI_GV_Font_Create：创建字体。
- HI_GV_Font_CreateByHandle：根据指定的句柄创建字体，为 XML2Bin 提供。
- HI_GV_Font_GetTextExtent：获取字符串的宽高。
- HI_GV_Font_Destroy：删除字体。
- HI_GV_Font_SetSystemDefault：设置系统默认字体。
- HI_GV_Font_GetSystemDefault：获取系统缺省字体。
- HI_GV_Resm_DestroyAllStyle：销毁所有 Style。
- HI_GV_Resm_DestroyAllRes：销毁所有图片、字体资源。
- HI_GV_Resm_ForceUnloadAllRes：强制释放所有图片、字体资源。
- HI_GV_Resm_SetDecSurfInfo：设置解码后表面信息。
- HI_GV_Resm_GetDecSurfInfo：获取解码后表面信息。
- HI_GV_Res_GetResInfo：获取图片资源 ID 对应的 surface 信息。
- HI_GV_Res_ReleaseResInfo：释放图片资源 ID 对应的 surface 信息。
- HI_GV_FontSet_Create：创建字体集。
- HI_GV_FontSet_AddFont：添加字体到字体集中。
- HI_GV_FontSet_Query：查询字体集中风格与 hFont 相同的字体。
- HI_GV_FontSet_Destroy：删除字体集。
- HI_GV_Res_GetStyleFileName：获取 Style 文件名。
- HI_GV_Res_PrintCurLoadResInfo：打印资源信息。
- HI_GV_AnimInfo_Create：创建动画信息。
- HI_GV_AnimInfo_CreateByHandle：根据指定的句柄创建动画信息，为 XML2Bin 提供。
- HI_GV_AnimInfo_Destroy：删除动画信息。
- HI_GV_AnimInfo_Get：获取动画信息。

HI_GV_Res_CreateID

【描述】

创建资源 ID。

【语法】

```
HI_S32 HI_GV_Res_CreateID(const HI_CHAR* fileName, HIGV_RESTYPE_E resType, HI_RESID* resID);
```

【参数】

参数名称	描述	输入/输出
fileName	图片文件名指针。	输入
resType	资源类型。	输入
resID	资源 ID 指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

- 图片在系统内以 ID 标识。
- 该接口受环境变量影响，如果设置了资源路径前缀环境变量，则会自动在资源路径前面增加该前缀。字体资源相对路径前缀(HIGV_RES_FONT_PATH)，图片资源相对路径前缀(HIGV_RES_IMAGE_PATH)。
- 开发者调用 HI_GV_Res_CreateID 创建的资源，使用结束后，由开发者主动调用 HI_GV_Res_DestroyID 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_Res_CreateID_NoPrefixPath

【描述】

创建资源 ID，不读取资源前缀环境变量。

【语法】

```
HI_S32 HI_GV_Res_CreateID_NoPrefixPath(const HI_CHAR* fileName, HIGV_RESTYPE_E resType, HI_RESID* resID);
```

【参数】

参数名称	描述	输入/输出
fileName	图片文件名指针。	输入
resType	资源类型。	输入
resID	资源 ID 指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

开发者调用 HI_GV_Res_CreateID_NoPrefixPath 创建的资源，使用结束后，由开发者主动调用 HI_GV_Res_DestroyID 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_Res_DisablePrefixPath

【描述】

强制应用不使用资源路径的环境变量。

【语法】

```
HI_S32 HI_GV_Res_DisablePrefixPath(const HI_BOOL disable);
```

【参数】

参数名称	描述	输入/输出
disable	使能布尔值。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Res_DestroyID

【描述】

销毁资源 ID。

【语法】

```
HI_S32 HI_GV_Res_DestroyID(HI_RESID resID);
```

【参数】

参数名称	描述	输入/输出
resID	资源 ID	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Res_CreateStyle

【描述】

创建皮肤。

【语法】

```
HI_S32 HI_GV_Res_CreateStyle(const HIGV_STYLE_S* style, HIGV_HANDLE* styleHandle);
```

【参数】

参数名称	描述	输入/输出
stlye	皮肤各子项资源 ID 信息指针。	输入
styleHandle	皮肤句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

开发者调用 HI_GV_Res_CreateStyle 创建的资源，使用结束后，由开发者主动调用 HI_GV_Res_DestroyStyle 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_Res_CreateStyleByHandle

【描述】

根据指定的 RESID 生成皮肤。

【语法】

```
HI_S32 HI_GV_Res_CreateStyleByHandle(const HIGV_STYLE_S* style, HIGV_HANDLE styleHandle);
```

【参数】

参数名称	描述	输入/输出
style	创建皮肤各子项资源 ID 信息参数指针。	输入
styleHandle	句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

开发者调用 HI_GV_Res_CreateStyleByHandle 创建的资源，使用结束后，由开发者主动调用 HI_GV_Res_DestroyStyle 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_Res_DestroyStyle

【描述】

销毁皮肤。

【语法】

```
HI_S32 HI_GV_Res_DestroyStyle(HIGV_HANDLE styleHandle);
```

【参数】

参数名称	描述	输入/输出
styleHandle	Style 句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Res_SetResident

【描述】

设置资源常驻内存。

【语法】

```
HI_S32 HI_GV_Res_SetResident(HI_RESID resID);
```

【参数】

参数名称	描述	输入/输出
resID	资源句柄。	输入

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

- 使用场景：如果资源比较常用或者资源解码太慢，需要一直常驻，用了这个以后不受 isrelease 影响，任何时候都不会释放，此接口一般不会使用。

【举例】

无。

HI_GV_Font_Create

【描述】

创建字体。

【语法】

```
HI_S32 HI_GV_Font_Create(const HIGV_FONT_S* fontInfo, HIGV_HANDLE* fontHandle);
```

【参数】

参数名称	描述	输入/输出
fontInfo	字体创建信息指针。	输入
fontHandle	字体句柄指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

开发者调用 HI_GV_Font_Create 创建的资源，使用结束后，由开发者主动调用 HI_GV_Font_Destroy 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_Font_CreateByHandle

【描述】

根据指定的句柄创建字体，为 XML2Bin 提供。

【语法】

```
HI_S32 HI_GV_Font_CreateByHandle(const HIGV_FONT_S* fontInfo, HIGV_HANDLE fontHandle);
```

【参数】

参数名称	描述	输入/输出
fontInfo	字体创建信息指针。	输入
fontHandle	字体句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h

- 库文件：libhigv.a、libhigv.so

【注意】

开发者调用 HI_GV_Font_CreateByHandle 创建的资源，使用结束后，由开发者主动调用 HI_GV_Font_Destroy 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_Font_GetTextExtent

【描述】

获取字符串的宽高。

【语法】

```
HI_S32 HI_GV_Font_GetTextExtent(HIGV_HANDLE fontHandle, HI_RESID strHandle, HI_S32* width, HI_S32* height);
```

【参数】

参数名称	描述	输入/输出
fontHandle	字体句柄	输入
strHandle	字符串句柄	输入
width	字符串宽度指针	输出
height	字符串高度指针	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Font_Destroy

【描述】

删除字体。

【语法】

```
HI_S32 HI_GV_Font_Destroy(HIGV_HANDLE fontHandle);
```

【参数】

参数名称	描述	输入/输出
fontHandle	字体句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Font_SetSystemDefault

【描述】

设置系统默认字体。

【语法】

```
HI_S32 HI_GV_Font_SetSystemDefault(HIGV_HANDLE fontHandle);
```

【参数】

参数名称	描述	输入/输出
fontHandle	字体资源句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Font_GetSystemDefault

【描述】

获取系统缺省字体。

【语法】

```
HI_S32 HI_GV_Font_GetSystemDefault(HIGV_HANDLE* fontHandle);
```

【参数】

参数名称	描述	输入/输出
fontHandle	字体句柄	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Resm_DestroyAllStyle

【描述】

销毁所有皮肤。

【语法】

```
HI_S32 HI_GV_Resm_DestroyAllStyle(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Resm_DestroyAllRes

【描述】

销毁所有图片、字体资源。

【语法】

```
HI_S32 HI_GV_Resm_DestroyAllRes(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Resm_ForceUnloadAllRes

【描述】

强制释放所有图片、字体资源。

【语法】

```
HI_S32 HI_GV_Resm_ForceUnloadAllRes(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Resm_SetDecSurfInfo

【描述】

设置解码后表面信息。

【语法】

```
HI_S32 HI_GV_Resm_SetDecSurfInfo(const HIGV_DEC_SUFINFO_S* decSufinfo);
```

【参数】

参数名称	描述	输入/输出
decSufinfo	解码表面信息指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Resm_GetDecSurfInfo

【描述】

获取解码后表面信息。

【语法】

```
HI_S32 HI_GV_Resm_GetDecSurfInfo(HIGV_DEC_SUFINFO_S* decSufinfo);
```

【参数】

参数名称	描述	输入/输出
decSufinfo	解码表面信息指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Res_GetResInfo

【描述】

获取图片资源 ID 对应的 surface 信息。

【语法】

```
HI_S32 HI_GV_Res_GetResInfo(HI_RESID resID, HIGV_HANDLE* resHandle);
```

【参数】

参数名称	描述	输入/输出
resID	资源 ID。	输入
resHandle	资源 surface 句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Res_ReleaseResInfo

【描述】

释放图片资源 ID 对应的 surface 信息。

【语法】

```
HI_S32 HI_GV_Res_ReleaseResInfo(HI_RESID resID);
```

【参数】

参数名称	描述	输入/输出
resID	资源 ID。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_FontSet_Create

【描述】

创建字体集。

【语法】

```
HI_S32 HI_GV_FontSet_Create(HIGV_HANDLE* fontSet);
```

【参数】

参数名称	描述	输入/输出
fontSet	字体句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

开发者调用 HI_GV_FontSet_Create 创建的资源，使用结束后，由开发者主动调用 HI_GV_FontSet_Destroy 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_FontSet_AddFont

【描述】

添加字体到字体集中。

【语法】

```
HI_S32 HI_GV_FontSet_AddFont(HIGV_HANDLE fontHandle, const HI_CHAR* supportLan, HIGV_HANDLE fontSetHandle);
```

【参数】

参数名称	描述	输入/输出
fontHandle	字体句柄。	输入
supportLan	字体支持的语言指针；多种语言使用分号隔开，如"cn;en;"，语言代码建议使用 ISO-639 标准。	输入
fontSetHandle	字体集句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_FontSet_Query

【描述】

查询字体集中风格与 hFont 相同的字体。

【语法】

```
HI_S32 HI_GV_FontSet_Query(const HI_CHAR* lan, HIGV_HANDLE fontHandle, HIGV_HANDLE* supprotFontHandle);
```

【参数】

参数名称	描述	输入/输出
lan	字体支持的语言指针；语言代码建议使用 ISO-639 标准。	输入
fontHandle	字体句柄。	输入
supprotFontHandle	字体集中风格与 fontHandle 相同的字体。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_FontSet_Destroy

【描述】

删除字体集。

【语法】

```
HI_S32 HI_GV_FontSet_Destroy(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Res_GetStyleFileName

【描述】

获取 Style 文件名。

【语法】

```
HI_S32 HI_GV_Res_GetStyleFileName(HI_RESID styleHandle, HIGV_STYLEFILENAME_S*
styleFileNameInfo);
```

【参数】

参数名称	描述	输入/输出
styleHandle	style 句柄。	输入
styleFileNameInfo	获取组成 style 的资源文件名指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

目前该功能未开启，需要申明_MEM_COUNT 宏。

【举例】

无。

HI_GV_Res_PrintCurLoadResInfo

【描述】

打印 HIGV_RESTYPE_IMG 资源内存占用信息。

【语法】

```
HI_S32 HI_GV_Res_PrintCurLoadResInfo(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_AnimInfo_Create

【描述】

创建动画信息。

【语法】

```
HI_S32 HI_GV_AnimInfo_Create(const HIGV_ANIM_INFO_S* animInfo, HIGV_HANDLE
*animHandle);
```

【参数】

参数名称	描述	输入/输出
animInfo	创建动画信息指针。	输入
animHandle	动画信息句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_AnimInfo_CreateByHandle

【描述】

根据指定的句柄创建动画信息，为 XML2Bin 提供。

【语法】

```
HI_S32 HI_GV_AnimInfo_CreateByHandle(const HIGV_ANIM_INFO_S* animInfo, HIGV_HANDLE animHandle);
```

【参数】

参数名称	描述	输入/输出
animInfo	创建动画信息。	输入

参数名称	描述	输入/输出
animHandle	动画信息句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_AnimInfo_Destroy

【描述】

删除动画信息。

【语法】

```
HI_S32 HI_GV_AnimInfo_Destroy(HIGV_HANDLE animHandle);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画信息句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_AnimInfo_Get

【描述】

获取动画信息。

【语法】

```
HI_S32 HI_GV_AnimInfo_Get(HIGV_HANDLE animHandle, HIGV_ANIM_INFO_S *animInfo);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画信息句柄。	输入
animInfo	动画信息。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.1.4.2 数据类型

相关数据类型、数据结构定义如下：

- HI_RESID：资源 ID。
- HIGV_RESTYPE_E：资源类别。
- HIGV_STYLE_TYPE_E：皮肤类型。
- HIGV_STYLE_MEMBER_U：Style 联合。
- HIGV_STYLE_S：Style 结构。
- HIGV_FONT_S：字体结构。
- HIGV_DEC_SUFINFO_S：解码表面信息结构。
- HIGV_STYLEFILENAME_S：资源文件名结构。
- HIGV_ANIM_TYPE_E：动画类型。
- HIGV_ANIM_REPEAT_TYPE_E：动画重复类型。
- HIGV_ANIM_ROLL_DIRECTION_E：卷帘方向。
- HIGV_ANIM_TRANSLATE_INFO_S：平移信息。
- HIGV_ANIM_ALPHA_INFO_S：Alpha 信息。
- HIGV_ANIM_ANY_INFO_S：通用信息。
- HIGV_ANIM_ROLL_INFO_S：卷帘信息。
- HIGV_ANIM_INFO_S：动画信息。

HI_RESID

【说明】

资源 ID。

【定义】

```
typedef     HI_PARAM HI_RESID;
```

【成员】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_RESTYPE_E

【说明】

资源类别。

【定义】

```
typedef enum hiHIGV_RESTYPE_E
{
    HIGV_RESTYPE_IMG,
    HIGV_RESTYPE_FONT,
    HIGV_RESTYPE_BUTT
} HIGV_RESTYPE_E;
```

【成员】

成员名称	描述
HIGV_RESTYPE_IMG	图片资源。
HIGV_RESTYPE_FONT	字体资源。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_STYLE_TYPE_E

【说明】

皮肤类型。

【定义】

```
typedef enum hiHIGV_STYLE_TYPE_E
{
    HIGV_STYLETYPE_COLOR = 0,
    HIGV_STYLETYPE_PIC
} HIGV_STYLE_TYPE_E;
```

【成员】

成员名称	描述
HIGV_STYLETYPE_COLOR	颜色。
HIGV_STYLETYPE_PIC	图片。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_STYLE_MEMBER_U

【说明】

Style 联合。

【定义】

```
typedef union hiHIGV_STYLE_MEMBER_U
{
    HI_COLOR Color;
    HI_RESID ResId;
} HIGV_STYLE_MEMBER_U;
```


【成员】

成员名称	描述
Color	颜色值。
ResId	资源 ID。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_STYLE_S

【说明】

Style 结构。

【定义】

```
typedef struct hiHIGV_STYLE_S
{
    HIGV_STYLE_TYPE_E     StyleType;
    HIGV_STYLE_MEMBER_U Top;
    HIGV_STYLE_MEMBER_U Bottom;
    HIGV_STYLE_MEMBER_U Left;
    HIGV_STYLE_MEMBER_U Right;
    HIGV_STYLE_MEMBER_U LeftTop;
    HIGV_STYLE_MEMBER_U LeftBottom;
    HIGV_STYLE_MEMBER_U RightTop;
    HIGV_STYLE_MEMBER_U RightBottom;
    HIGV_STYLE_MEMBER_U BackGround;
    HI_COLOR                FontColor;
    HI_U32                bNoDrawBg;
    HI_U32                LineWidth;
} HIGV_STYLE_S;
```

【成员】

成员名称	描述
StyleType	Style 类型。
Top	上边缘颜色值或者图片资源 ID。
Bottom	下边缘颜色值或者图片资源 ID。
Left	左边缘颜色值或者图片资源 ID。
Right	右边缘颜色值或者图片资源 ID。
LeftTop	左上角颜色值或者图片资源 ID。
LeftBottom	左下角颜色值或者图片资源 ID。
RightTop	右上角颜色值或者图片资源 ID。
RightBottom	右下角颜色值或者图片资源 ID。
BackGround	背景颜色值或图片资源 ID。
FontColor	文字颜色值。
bNoDrawBg	控制是否画背景颜色。
LineWidth	控制填充颜色宽度。

【注意事项】

无。

【相关数据类型及接口】

- HIGV_STYLE_TYPE_E
- HIGV_STYLE_MEMBER_U

HIGV_FONT_S

【说明】

字体结构。

【定义】

```
typedef struct hiHIGV_FONT_S
{
```



```
    HI_RESID SbFontID;
    HI_RESID MbFontID;
    HI_U32     Size;
    HI_BOOL    bBold;
    HI_BOOL    bItalic;
    HI_U32     lineSpace;
} HIGV_FONT_S;
```

【成员】

成员名称	描述
SbFontID	首选字库。
MbFontID	备选字库。
Size	字体大小。
bBold	是否加粗。
bItalic	是否斜体。
lineSpace	行间距。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_DEC_SUFINFO_S

【说明】

解码表面信息结构。

【定义】

```
typedef struct hiHIGV_DEC_SUFINFO_S
{
    HIGO_PF_E PixFormat;
    HIGO_MEMTYPE_E MemType;
```



```
    HI_BOOL IsPubPalette;
} HIGV_DEC_SUFINFO_S;
```

【成员】

成员名称	描述
PixFormat	解码后表面的像素格式，默认为HIGO_PF_8888。
MemType	解码后表面的存储类别，默认为 MMZ。
IsPubPalette	解码图片是否使用公共调色板,只对输出格式有效。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_STYLEFILENAME_S

【说明】

资源文件名结构。

【定义】

```
typedef struct
{
    HI_CHAR* pTopName;
    HI_CHAR* pBottomName;
    HI_CHAR* pLeftName;
    HI_CHAR* pRightName;
    HI_CHAR* pLeftTopName;
    HI_CHAR* pLeftBottomName;
    HI_CHAR* pRightTopName;
    HI_CHAR* pRightBottomName;
    HI_CHAR* pBackGroundName;
} HIGV_STYLEFILENAME_S;
```


【成员】

成员名称	描述
pTopName	顶部的资源文件路径。
pBottomName	底部的资源文件路径。
pLeftName	左边的资源文件路径。
pRightName	右边的资源文件路径。
pLeftTopName	左上的资源文件路径。
pLeftBottomName	左下的资源文件路径。
pRightTopName	右上的资源文件路径。
pRightBottomName	右下的资源文件路径。
pBackGroundName	背景的资源文件路径。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_ANIM_TYPE_E

【说明】

动画类型。

【定义】

```
typedef enum hiHIGV_ANIM_TYPE_E
{
    HIGV_ANIMTYPE_TRANSLATE = 0,
    HIGV_ANIMTYPE_ALPHA,
    HIGV_ANIMTYPE_ROLL,
    HIGV_ANIMTYPE_ANY,
    HIGV_ANIMTYPE_INVALIDATE
```



```
} HIGV_ANIM_TYPE_E;
```

【成员】

成员名称	描述
HIGV_ANIMTYPE_TRANSLATE	平移动画。
HIGV_ANIMTYPE_ALPHA	Alpha 渐变动画。
HIGV_ANIMTYPE_ROLL	卷帘动画
HIGV_ANIMTYPE_ANY	任意动画类型（用于扩展）

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_ANIM_REPEAT_TYPE_E

【说明】

动画重复类型。

【定义】

```
typedef enum hiHIGV_ANIM_REPEAT_TYPE_E
{
    HIGV_REPEATE_TYPE_RESTART = 0,
    HIGV_REPEATE_TYPE_REVERSE
} HIGV_ANIM_REPEAT_TYPE_E;
```

【成员】

成员名称	描述
HIGV_REPEATE_TYPE_RESTART	从头开始。
HIGV_REPEATE_TYPE_REVERSE	从结束的位置继续。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_ANIM_ROLL_DIRECTION_E

【说明】

卷帘方向。

【定义】

```
typedef enum
{
    HIGV_ANIM_ROLL_UP   = 0,
    HIGV_ANIM_ROLL_DOWN,
    HIGV_ANIM_ROLL_LEFT,
    HIGV_ANIM_ROLL_RIGHT
} HIGV_ANIM_ROLL_DIRECTION_E;
```

【成员】

成员名称	描述
HIGV_ANIM_ROLL_UP	向上
HIGV_ANIM_ROLL_DOWN	向下
HIGV_ANIM_ROLL_LEFT	向左
HIGV_ANIM_ROLL_RIGHT	向右

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_ANIM_TRANSLATE_INFO_S

【说明】

平移信息。

【定义】

```
typedef struct hiHIGV_ANIM_TRANSLATE_INFO_S
{
    HI_S32          FromX;
    HI_S32          FromY;
    HI_S32          ToX;
    HI_S32          ToY;
} HIGV_ANIM_TRANSLATE_INFO_S;
```

【成员】

成员名称	描述
FromX	开始坐标 X
FromY	开始坐标 Y
ToX	结束坐标 X
ToY	结束坐标 Y

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_ANIM_ALPHA_INFO_S

【说明】

Alpha 信息。

【定义】

```
typedef struct hiHIGV_ANIM_ALPHA_INFO_S
{
```



```
    HI_U32          FromApha;
    HI_U32          ToAlpha;
} HIGV_ANIM_ALPHA_INFO_S;
```

【成员】

成员名称	描述
FromApha	开始 Alpha
ToAlpha	结束 Alpha

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_ANIM_ANY_INFO_S

【说明】

通用信息。

【定义】

```
typedef struct hiHIGV_ANIM_ANY_INFO_S
{
    HI_S32          FromValue;
    HI_S32          ToValue;
} HIGV_ANIM_ANY_INFO_S;
```

【成员】

成员名称	描述
FromValue	开始 Value
ToValue	结束 Value

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_ANIM_ROLL_INFO_S

【说明】

卷帘信息。

【定义】

```
typedef struct hiHIGV_ANIM_ROLL_INFO_S
{
    HIGV_ANIM_ROLL_DIRECTION_E    DirectionType;
    HI_S32                        FromLoc;
    HI_S32                        ToLoc;
} HIGV_ANIM_ROLL_INFO_S;
```

【成员】

成员名称	描述
DirectionType	卷帘方向
FromLoc	开始位置
ToLoc	结束位置

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_ANIM_INFO_S

【说明】

动画信息。

【定义】

```
typedef struct hiHIGV_ANIM_INFO_S
{
    HI_U32              AnimHandle;
    HI_U32              DurationMs;
    HI_S32              RepeatCount;
    HIGV_ANIM_REPEAT_TYPE_E RepeatMode ;
    HI_U32              DelayStart;
    HIGV_ANIM_TYPE_E AnimType;
    union
    {
        HIGV_ANIM_TRANSLATE_INFO_S Translate;
        HIGV_ANIM_ALPHA_INFO_S Alpha;
        HIGV_ANIM_ROLL_INFO_S Roll;
        HIGV_ANIM_ANY_INFO_S Any;
    }AnimParam;
} HIGV_ANIM_INFO_S;
```

【成员】

成员名称	描述
AnimHandle	Anim ID
DurationMs	持续时间
RepeatCount	循环次数
RepeatMode	循环模式
DelayStart	延迟执行时间
AnimType	动画类型
AnimParam	动画信息

【注意事项】

无。

【相关数据类型及接口】

无。

2.1.5 消息管理

2.1.5.1 接口描述

消息管理模块提供以下 API：

- HI_GV_Msg_SendSync：发送同步消息。
- HI_GV_Msg_SendAsync：发送不可丢失的异步消息。
- HI_GV_Msg_PostAsync：发送可丢失的异步消息。
- HI_GV_Msg_SendAsyncWithData：发送带附加数据不可丢失的异步消息。

HI_GV_Msg_SendSync

【描述】

发送同步消息。

【语法】

```
HI_S32 HI_GV_Msg_SendSync(HIGV_HANDLE widgetHandle, HI_U32 msgId, HI_PARAM param1, HI_PARAM param2, HI_U32 timeOut);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	窗口或控件句柄。	输入
msgId	消息 ID，具体定义参考头文件 hi_gv_widget.h	输入
param1	参数 1，其含义由 msgId 确定，对按键消息来说是按键值。	输入
param2	参数 2，其含义由 msgId 确定，对按键消息来说是特殊按键(如 shift、ctrl 等)状态。	输入
timeOut	超时时间，单位：ms。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_msg.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

- msgId 具体消息定义参考头文件 hi_gv_widget.h。
- 触摸手势事件消息不能通过此接口发送，而只能使用 HI_GV_Msg_SendAsyncWithData 接口发送，包括：HIGV_MSG_TOUCH、HIGV_MSG_GESTURE_TAP、HIGV_MSG_GESTURE_LONGTAP、HIGV_MSG_GESTURE_FLING、HIGV_MSG_GESTURE_SCROLL 和 HIGV_MSG_GESTURE_PINCH。
- 鼠标光标事件消息，需要将消息发送给窗口管理，因为光标的显示由窗口管理负责。并且鼠标的坐标为相对位移坐标。

【举例】

```
HI_GV_Msg_SendSync(HOME_WINDOW, HIGV_MSG_EVENT, 0, 0, 1000)
HI_GV_Msg_SendSync(HIGV_WND_MANAGER_HANDLE, HIGV_MSG_MOUSEMOVE, 0, (16 << 16) | (8 & 0xffff), 1000)
```

HI_GV_Msg_SendAsync

【描述】

发送不可丢失的异步消息。

【语法】

```
HI_S32 HI_GV_Msg_SendAsync(HIGV_HANDLE widgetHandle, HI_U32 msgId, HI_PARAM param1, HI_PARAM param2);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	窗口或控件句柄。	输入
msgId	消息 ID，具体定义参考头文件 hi_gv_widget.h。	输入
param1	参数 1，其含义由 msgId 确定，对按键消息来说是按键值。	输入
param2	参数 2，其含义由 msgId 确定，对按键消息来说是特殊按键(如 shift、ctrl 等)状态。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_msg.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

- msgId 具体消息定义参考头文件 hi_gv_widget.h。
- 触摸手势事件消息不能通过此接口发送，而只能使用 HI_GV_Msg_SendAsyncWithData 接口发送，包括：HIGV_MSG_TOUCH、HIGV_MSG_GESTURE_TAP、HIGV_MSG_GESTURE_LONGTAP、HIGV_MSG_GESTURE_FLING、HIGV_MSG_GESTURE_SCROLL 和 HIGV_MSG_GESTURE_PINCH。
- 鼠标光标事件消息，需要将消息发送给窗口管理，因为光标的显示由窗口管理负责。并且鼠标的坐标为相对位移坐标。

【举例】

```
HI_GV_Msg_SendAsync (HIGV_WND_MANAGER_HANDLE, HIGV_MSG_MOUSEMOVE, 0, (16 << 16) | (8 & 0xffff))
```


HI_GV_Msg_PostAsync

【描述】

发送可丢失的异步消息。

【语法】

```
HI_S32 HI_GV_Msg_PostAsync(HIGV_HANDLE widgetHandle, HI_U32 msgId, HI_PARAM param1, HI_PARAM param2);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	窗口或控件句柄。	输入
msgId	消息 ID，具体定义参考头文件 hi_gv_widget.h。	输入
param1	参数 1，其含义由 msgId 确定，对按键消息来说是按键值。	输入
param2	参数 2，其含义由 msgId 确定，对按键消息来说是特殊按键(如 shift、ctrl 等)状态。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_msg.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

- msgId 具体消息定义参考头文件 hi_gv_widget.h。
- 触摸手势事件消息不能通过此接口发送，而只能使用 HI_GV_Msg_SendAsyncWithData 接口发送，包括：HIGV_MSG_TOUCH、

HIGV_MSG_GESTURE_TAP、HIGV_MSG_GESTURE_LONGTAP、HIGV_MSG_GESTURE_FLING、HIGV_MSG_GESTURE_SCROLL 和 HIGV_MSG_GESTURE_PINCH。

- 鼠标光标事件消息，需要将消息发送给窗口管理，因为光标的显示由窗口管理负责。并且鼠标的坐标为相对位移坐标。

【举例】

```
HI_GV_Msg_ PostAsync (HIGV_WND_MANAGER_HANDLE, HIGV_MSG_MOUSEMOVE, 0, (16 << 16) | (8 & 0xffff))
```

HI_GV_Msg_SendAsyncWithData

【描述】

发送带附加数据不可丢失的异步消息。

【语法】

```
HI_S32 HI_GV_Msg_SendAsyncWithData(HIGV_HANDLE widgetHandle, HI_U32 msgId, HI_VOID* buf, HI_PARAM bufLen);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	窗口或控件句柄。	输入
msgId	消息 ID，具体定义参考头文件 hi_gv_widget.h	输入
buf	消息附带的数据指针。	输入
bufLen	消息附带的数据长度。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_msg.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

- msgId 具体消息定义参考头文件 hi_gv_widget.h。
- 数据长度必须与数据内容匹配，否则会产生不可预估的问题。

【举例】

无。

2.1.5.2 数据类型

相关数据类型、数据结构定义如下：

- HIGV_MSGINFO_S：消息信息结构。
- HIGV_KEYTYPE_E：按键类型。

HIGV_MSGINFO_S

【说明】

消息信息结构。

【定义】

```
typedef struct hiHIGV_MSGINFO_S
{
    HI_U32 HighLevelMsgNum;
    HI_U32 MiddleLevelMsgNum;
        HI_U32 LowLevelMsgNum;
        HI_U32 CanLoseMsgNum;
    HI_U32 LoseMsgNum;
    HI_U32 NoLoseMsgNum;
} HIGV_MSGINFO_S;
```

【成员】

成员名称	描述
HighLevelMsgNum	高优先级队列中消息数。
MiddleLevelMsgNum	中优先级队列中消息数。

成员名称	描述
LowLevelMsgNum	低优先级中消息数。
CanLoseMsgNum	可丢失消息队列中消息数。
LoseMsgNum	可丢失消息队列中实际已发生的丢消息数。
NoLoseMsgNum	不可丢失消息队列中的消息数。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_KEYTYPE_E

【说明】

按键类型。

【定义】

```
typedef enum hiHIGV_KEYTYPE_E
{
    HIGV_KEYTYPE_UNICODE   = 0x0,
    HIGV_KEYTYPE_FUNCTION = 0x10000,
} HIGV_KEYTYPE_E;
```

【成员】

成员名称	描述
HIGV_KEYTYPE_UNICODE	ascii 码表中定义的键值。
HIGV_KEYTYPE_FUNCTION	功能键。

【注意事项】

无。

【相关数据类型及接口】

无。

2.1.6 窗口管理

2.1.6.1 接口描述

窗口管理模块提供以下 API：

- HI_GV_WM_PushWindow：将窗口入栈。
- HI_GV_WM_PopWindow：将窗口出栈。
- HI_GV_WM_GetTopWindow：获取栈顶窗口。
- HI_GV_WM_ClearAllWindow：清除栈中窗口。
- HI_GV_WM_GetMouseCapture：获得鼠标捕获窗口。
- HI_GV_WM_SetMouseCapture：设置鼠标捕获窗口。
- HI_GV_WM_GetHigoLayer：获取图层对应的 HIGO 图层句柄。
- HI_GV_WM_BindTouchMsg：触摸消息和控件绑定。
- HI_GV_WM_GetBindTouchStatus：获取绑定状态。
- HI_GV_WM_SetWndMemMode：设置窗口内存模式。
- HI_GV_WM_GetWndMemMode：获取窗口内存模式。

HI_GV_WM_PushWindow

【描述】

将窗口入栈。

【语法】

```
HI_S32 HI_GV_WM_PushWindow(HIGV_HANDLE layerHandle, HIGV_HANDLE windowHandle);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入
windowHandle	窗口句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_wm.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WM_PopWindow

【描述】

将窗口出栈。

【语法】

```
HI_S32 HI_GV_WM_PopWindow(HIGV_HANDLE layerHandle, HIGV_HANDLE windowHandle);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入
windowHandle	窗口句柄。	输入

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_wm.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WM_GetTopWindow

【描述】

获取栈顶窗口。

【语法】

```
HI_S32 HI_GV_WM_GetTopWindow(HIGV_HANDLE layerHandle, HIGV_HANDLE* windowHandle);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入
windowHandle	窗口句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_wm.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WM_ClearAllWindow

【描述】

清除栈中窗口。

【语法】

```
HI_S32 HI_GV_WM_ClearAllWindow(HIGV_HANDLE layerHandle);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_wm.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WM_GetMouseCapture

【描述】

获得鼠标捕获窗口。

【语法】

```
HI_S32 HI_GV_WM_GetMouseCapture(HIGV_HANDLE layerHandle, HIGV_HANDLE* widgetHandle);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入
widgetHandle	捕获鼠标的窗口。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_wm.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WM_SetMouseCapture

【描述】

设置鼠标捕获窗口。

【语法】

```
HI_S32 HI_GV_WM_SetMouseCapture(HIGV_HANDLE layerHandle, HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入
widgetHandle	鼠标捕获窗口。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_wm.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WM_GetHigoLayer

【描述】

获取图层对应的 HIGO 图层句柄。

【语法】

```
HI_S32 HI_GV_WM_GetHigoLayer(HIGV_HANDLE layerHandle, HIGV_HANDLE* higoLayer);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入
higoLayer	HIGO 图层句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_wm.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WM_BindTouchMsg

【描述】

触摸消息和控件绑定。

【语法】

```
HI_S32 HI_GV_WM_BindTouchMsg(HIGV_HANDLE layerHandle, HI_BOOL isBind);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入
isBind	是否绑定控件。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_wm.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WM_GetBindTouchStatus

【描述】

获取绑定状态。

【语法】

```
HI_S32 HI_GV_WM_GetBindTouchStatus(HIGV_HANDLE layerHandle, HI_U32 *isBind);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入
isBind	绑定状态指针。 HI_TRUE：已绑定； HI_FALSE：未绑定。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_wm.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WM_SetWndMemMode

【描述】

设置窗口内存模式。

【语法】

```
HI_S32 HI_GV_WM_SetWndMemMode(HIGV_HANDLE layerHandle, HIGV_WndMemType mode);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入
mode	窗口内存模式。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_wm.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WM_GetWndMemMode

【描述】

获取窗口内存模式。

【语法】

```
HI_S32 HI_GV_WM_GetWndMemMode(HIGV_HANDLE layerHandle, HIGV_WndMemType *mode);
```

【参数】

参数名称	描述	输入/输出
layerHandle	图层句柄。	输入

参数名称	描述	输入/输出
mode	窗口内存模式。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_wm.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.1.6.2 数据类型

相关数据类型、数据结构定义如下：

- HIGV_MAX_WINNUM：窗口栈容量。
- HIGV_WndMemType：窗口内存模式。

HIGV_MAX_WINNUM

【说明】

窗口栈容量。

【定义】

```
#define HIGV_MAX_WINNUM 50
```

【成员】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_WndMemType

【说明】

窗口内存模式。

【定义】

```
typedef enum {
    HI_GV_WNDMEM_SHARED = 0,
    HI_GV_WNDMEM_SEP,
    HI_GV_WNDMEM_MAX,
} HIGV_WndMemType;
```

【成员】

成员名称	描述
HI_GV_WNDMEM_SHARED	所有窗口共用一块内存。
HI_GV_WNDMEM_SEP	单独使用一块内存。

【注意事项】

HI_GV_WNDMEM_SHARED 模式下，必须要保证所有的窗口和图层的像素格式一致。

【相关数据类型及接口】

无。

2.1.7 调试打印管理

2.1.7.1 接口描述

调试打印管理模块提供以下 API：

- HI_GV_Log_SetLevel：设置模块的输出 LOG 的级别。
- HI_GV_Log_GetLevel：获取模块的输出 LOG 的级别。
- HI_GV_Log_Output：输出模块的调试打印信息。

HI_GV_Log_SetLevel

【描述】

设置模块的输出 LOG 的级别。

【语法】

```
HI_S32 HI_GV_Log_SetLevel(const HI_CHAR* domain, HIGV_LOG_LEVEL_E level);
```

【参数】

参数名称	描述	输入/输出
domain	模块名字指针。	输入
level	LOG 输出级别。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_log.h、hi_gv.h、hi_gv_errno.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Log_GetLevel

【描述】

获取模块的输出 LOG 的级别。

【语法】

```
HIGV_LOG_LEVEL_E HI_GV_Log_GetLevel(const HI_CHAR* domain);
```

【参数】

参数名称	描述	输入/输出
domain	模块名字指针。	输入

【返回值】

返回值	描述
HIGV_LOG_LEVEL_E	LOG 输出级别。

【需求】

- 头文件：hi_gv_log.h、hi_gv.h、hi_gv_errno.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Log_Output

【描述】

输出模块的调试打印信息。

【语法】

```
HI_S32 HI_GV_Log_Output(const HI_CHAR* domain, HIGV_LOG_LEVEL_E level, HI_S32 error,
const HI_CHAR* fileName, const HI_CHAR* functionName, HI_U32 line, const HI_CHAR* text, ...);
```


【参数】

参数名称	描述	输入/输出
domain	模块名字指针。	输入
level	log 输出级别。	输入
error	错误码。	输入
fileName	文件名称	输入
functionName	函数名称	输入
line	行号	输入
text	输出内容	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_log.h、hi_gv.h、hi_gv_errno.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.1.7.2 数据类型

相关数据类型、数据结构定义如下：

HIGV_LOG_LEVEL_E：调试打印级别。

HIGV_LOG_LEVEL_E

【说明】

调试打印级别。

【定义】

```
typedef enum
{
    HIGV_LOG_UNKNOWN = 0,
    HIGV_LOG_DEFAULT,
    HIGV_LOG_VERBOSE,
    HIGV_LOG_DEBUG,
    HIGV_LOG_INFO,
    HIGV_LOG_WARNING,
    HIGV_LOG_ERROR,
    HIGV_LOG_FATAL,
    HIGV_LOG_SILENT,
    HIGV_LOG_BUTT
} HIGV_LOG_LEVEL_E;
```

【成员】

成员名称	描述
HIGV_LOG_VERBOSE	最低的优先级。
HIGV_LOG_DEBUG	打印 Debug、Info、Warning 和 Error 级别调试打印信息。
HIGV_LOG_INFO	打印 Info、Warning 和 Error 级别调试打印信息。
HIGV_LOG_WARNING	打印 Warning 和 Error 级别调试打印信息。
HIGV_LOG_ERROR	打印 Error 级别调试打印信息。
HIGV_LOG_FATAL	打印 Fatal 级别调试打印信息。

【注意事项】

无。

【相关数据类型及接口】

无。

2.1.8 绘制上下文

2.1.8.1 接口描述

绘制上下文模块提供以下 API：

- HI_GV_GraphicContext_Create：创建一个控件的绘制环境。
- HI_GV_GraphicContext_Destroy：销毁绘制环境。
- HI_GV_GraphicContext_Begin：开始绘制，添加默认的剪切矩形。
- HI_GV_GraphicContext_End：结束绘制，删除剪切矩形。
- HI_GV_GraphicContext_AddClipRect：添加剪切矩形。
- HI_GV_GraphicContext_SetClipRect：重设剪切矩形。
- HI_GV_GraphicContext_DecodeImg：解码本地图片。
- HI_GV_GraphicContext_DecodeMemImg：解码内存图片。
- HI_GV_GraphicContext_FreeImageSurface：释放解码图片 surface。
- HI_GV_GraphicContext_SetFgColor：设置前景色。
- HI_GV_GraphicContext_SetBgColor：设置文本的背景色。
- HI_GV_GraphicContext_DrawLine：画线。
- HI_GV_GraphicContext_DrawImage：绘制 surface 图片。
- HI_GV_GraphicContext_DrawText：设置绘制文本。
- HI_GV_GraphicContext_DrawTextByID：绘制多语言字串。
- HI_GV_GraphicContext_SetFont：设置文本字体。
- HI_GV_GraphicContext_FillRect：填充区域。

HI_GV_GraphicContext_Create

【描述】

创建一个控件的绘制环境。

【语法】

```
HI_S32 HI_GV_GraphicContext_Create(HIGV_HANDLE widgetHandle, HIGV_HANDLE*
contextHandle);
```


【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
contextHandle	绘制句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

- 与 HI_GV_GraphicContext_Destroy 接口配套使用。
- 开发者调用 HI_GV_GraphicContext_Create 创建的资源，使用结束后，由开发者主动调用 HI_GV_GraphicContext_Destroy 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_GraphicContext_Destroy

【描述】

销毁绘制环境。

【语法】

```
HI_S32 HI_GV_GraphicContext_Destroy(HIGV_HANDLE contextHandle);
```

【参数】

参数名称	描述	输入/输出
contextHandle	绘制句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_GraphicContext_Begin

【描述】

开始绘制，添加默认的剪切矩形。

【语法】

```
HI_S32 HI_GV_GraphicContext_Begin(HIGV_HANDLE contextHandle);
```

【参数】

参数名称	描述	输入/输出
contextHandle	绘制句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

与 HI_GV_GraphicContext_End 接口配套使用。

【举例】

无。

HI_GV_GraphicContext_End

【描述】

结束绘制，删除剪切矩形。

【语法】

```
HI_S32 HI_GV_GraphicContext_End(HIGV_HANDLE contextHandle);
```

【参数】

参数名称	描述	输入/输出
contextHandle	绘制句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_GraphicContext_AddClipRect

【描述】

添加剪切矩形。

【语法】

```
HI_S32 HI_GV_GraphicContext_AddClipRect(HIGV_HANDLE contextHandle, const HI_RECT* rect);
```

【参数】

参数名称	描述	输入/输出
contextHandle	绘制句柄。	输入
rect	传入矩形指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_GraphicContext_SetClipRect

【描述】

重设剪切矩形。

【语法】

```
HI_S32 HI_GV_GraphicContext_SetClipRect(HIGV_HANDLE contextHandle, const HI_RECT* rect);
```

【参数】

参数名称	描述	输入/输出
contextHandle	绘制句柄。	输入
rect	传入矩形指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_GraphicContext_DecodeImg

【描述】

解码本地图片。

【语法】

```
HI_S32 HI_GV_GraphicContext_DecodeImg(const HI_CHAR* fileName, HI_U32 imgWidth, HI_U32 imgHeight, HIGV_HANDLE* imgSurface);
```

【参数】

参数名称	描述	输入/输出
fileName	图片路径。	输入
imgWidth	解码后 surface 宽度，单位：像素。	输入
imgHeight	解码后 surface 高度，单位：像素。	输入
imgSurface	解码获得的 surface 句柄。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

- 与 HI_GV_GraphicContext_FreeImageSurface 接口配套使用。
- 如果带入宽高为零，图片大小为默认大小。

【举例】

无。

HI_GV_GraphicContext_DecodeMemImg

【描述】

解码内存图片。

【语法】

```
HI_S32 HI_GV_GraphicContext_DecodeMemImg(HI_CHAR* addr, HI_U32 length, HI_U32 imgWidth, HI_U32 imgHeight, HIGV_HANDLE* imgSurface);
```

【参数】

参数名称	描述	输入/输出
addr	内存图片地址。	输入
length	内存长度。	输入
imgWidth	解码后 surface 宽度，单位：像素。	输入
imgHeight	解码后 surface 高度，单位：像素。	输入
imgSurface	解码获得的 surface 句柄。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

- 与 HI_GV_GraphicContext_FreeImageSurface 接口配套使用。
- 如果带入宽高为零，图片大小为默认大小。

【举例】

无。

HI_GV_GraphicContext_FreeImageSurface

【描述】

释放解码图片 surface。

【语法】

```
HI_S32 HI_GV_GraphicContext_FreeImageSurface(HIGV_HANDLE imgSurface);
```

【参数】

参数名称	描述	输入/输出
imgSurface	图片 surface 句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_GraphicContext_SetFgColor

【描述】

设置前景色。

【语法】

```
HI_S32 HI_GV_GraphicContext_SetFgColor(HIGV_HANDLE contextHandle, HI_COLOR foreGroundColor);
```

【参数】

参数名称	描述	输入/输出
contextHandle	GC 句柄。	输入
foreGroundColor	前景色。 主要用于填充颜色或者设置字体颜色。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_GraphicContext_SetBgColor

【描述】

设置文本的背景色。

【语法】

```
HI_S32 HI_GV_GraphicContext_SetBgColor(HIGV_HANDLE contextHandle, HI_COLOR backGroudColor);
```

【参数】

参数名称	描述	输入/输出
contextHandle	GC 句柄。	输入
backGroudColor	文本的背景色。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

默认不设置背景色，设置后需要取消背景色，背景色传入 0 即可。

【举例】

无。

HI_GV_GraphicContext_DrawLine

【描述】

画线。

【语法】

```
HI_S32 HI_GV_GraphicContext_DrawLine(HIGV_HANDLE contextHandle, HI_U32 startX, HI_U32
```



```
startY, HI_U32 endX, HI_U32 endY, HI_U32 width);
```

【参数】

参数名称	描述	输入/输出
contextHandle	GC 句柄。	输入
startX	起点坐标 X。	输入
startY	起点坐标 Y。	输入
endX	终点坐标 X。	输入
endY	终点坐标 Y。	输入
width	直线线宽，单位：像素	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_GraphicContext_DrawImage

【描述】

绘制 surface 图片。

【语法】

```
HI_S32 HI_GV_GraphicContext_DrawImage(HIGV_HANDLE contextHandle, const HI_RECT*
aimRect, HIGV_HANDLE image, const HI_RECT* sourceRect, const HIGO_BLTOPT_S* blitOpt);
```

【参数】

参数名称	描述	输入/输出
contextHandle	绘制句柄。	输入
aimRect	绘制到 image 控件的矩形区域指针，为 NULL 时从控件坐标 0,0 开始绘制。	输入
image	surface 句柄。	输入
srcRect	surface 的源矩形区域指针,为 NULL 时绘制整个 surface。	输入
blitOpt	搬移混合操作运算属性。 详情请参考《HiGO API 参考文档》。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h、higo.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_GraphicContext_DrawText

【描述】

设置绘制文本。

【语法】

```
HI_S32 HI_GV_GraphicContext_DrawText(HIGV_HANDLE contextHandle, const HI_CHAR* text,
const HI_RECT* rect, HI_U32 align);
```

【参数】

参数名称	描述	输入/输出
contextHandle	绘制句柄。	输入
text	绘制文本内容指针。	输入
rect	绑定控件 surface 的绘制矩形指针。	输入
align	文本对齐方式。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_GraphicContext_DrawTextByID

【描述】

绘制多语言字串。

【语法】

```
HI_S32 HI_GV_GraphicContext_DrawTextByID(HIGV_HANDLE contextHandle, const HI_U32 strID, const HI_RECT* rect, HI_U32 align);
```

【参数】

参数名称	描述	输入/输出
contextHandle	绘制句柄。	输入
strID	多语言字串 ID。	输入
rect	绑定控件 surface 的绘制矩形指针。	输入
align	文本对齐方式。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_GraphicContext_SetFont

【描述】

设置文本字体。

【语法】

```
HI_S32 HI_GV_GraphicContext_SetFont(HIGV_HANDLE contextHandle, HIGV_HANDLE fontHandle);
```

【参数】

参数名称	描述	输入/输出
contextHandle	绘制句柄。	输入
fontHandle	字体句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_GraphicContext_FillRect

【描述】

填充区域。

【语法】

```
HI_S32 HI_GV_GraphicContext_FillRect(HIGV_HANDLE contextHandle, const HI_RECT* rect);
```

【参数】

参数名称	描述	输入/输出
contextHandle	绘制句柄。	输入
rect	在绑定 surface 上填充的区域指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_graphiccontext.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

与 HI_GV_GraphicContext_SetBgColor 接口配合使用。

【举例】

无。

2.1.9 动画控制

2.1.9.1 接口描述

须知

目前规格上只支持同时运行一种动画（包括 gif 播放动画）,一个窗口里面最多支持 512 个动画对象。且暂时不支持控件叠加场景时做动画。

动画控制模块提供两种动画方式：第一种方式是基于接口创建动画实例，提供更多动画细节给开发者，并提供动画轨迹变化曲线值等，其他由开发者自行实现；第二种方式是基于 XML 动画属性文件来创建动画实例，可以提供基本补间动画（包括平移、Alpha 渐变、卷帘）以及动画处理流程框架，相对简单易入手。API 接口如下：

第一种方式 API：

- HI_GV_TweenAnimCreate：创建补间动画实例。
- HI_GV_TweenAnimSetDuration：设置动画持续的时间。
- HI_GV_TweenAnimGetDuration：获取动画持续的时间。
- HI_GV_TweenAnimGetCurrentTime：获取动画当前运行的时间。
- HI_GV_TweenAnimSetDelay：设置动画延迟运行的时间。
- HI_GV_TweenAnimSetLoops：设置动画循环的次数。
- HI_GV_TweenAnimSetStartedListener：设置动画启动的回调函数。
- HI_GV_TweenAnimSetFinishedListener：设置动画完成的回调函数。
- HI_GV_TweenAnimSetExecListener：设置动画正在运行时的回调函数。
- HI_GV_TweenAnimStart：启动动画。
- HI_GV_TweenAnimStop：停止动画。
- HI_GV_TweenAnimResume：恢复动画。
- HI_GV_TweenAnimPause：暂停动画。
- HI_GV_TweenAnimAddTween：设置补间动画属性。
- HI_GV_TweenAnimGetTweenValue：获取补间动画值。
- HI_GV_TweenAnimSetTweenRange：设置补间动画范围。
- HI_GV_TweenAnimDestroy：销毁补间动画实例。

第二种方式 API（目前规格上只支持同时运行一种动画，特别是卷帘动画）：

- HI_GV_Anim_CreateInstance：创建补间动画实例。
- HI_GV_Anim_DestroyInstance：销毁补间动画实例。
- HI_GV_Anim_GetInfo：获取补间动画相关信息。

- HI_GV_Anim_Start：启动补间动画。
- HI_GV_Anim_Stop：停止补间动画。

HI_GV_TweenAnimCreate

【描述】

创建补间动画实例。

【语法】

```
HIGV_HANDLE HI_GV_TweenAnimCreate(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
非 0	实例句柄。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

- 与 HI_GV_TweenAnimDestroy 接口配合使用。
- 开发者调用 HI_GV_TweenAnimCreate 创建的资源，使用结束后，由开发者主动调用 HI_GV_TweenAnimDestroy 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_TweenAnimSetDuration

【描述】

设置动画持续的时间。

【语法】


```
HI_S32 HI_GV_TweenAnimSetDuration(HIGV_HANDLE animHandle, HI_U32 ms);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入
ms	动画持续的时间，单位：ms。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_TweenAnimGetDuration

【描述】

获取动画持续的时间。

【语法】

```
HI_S32 HI_GV_TweenAnimGetDuration(HIGV_HANDLE animHandle, HI_S32* duration);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入
duration	动画持续时间指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_TweenAnimGetCurrentTime

【描述】

获取动画当前运行的时间。

【语法】

```
HI_S32 HI_GV_TweenAnimGetCurrentTime(HIGV_HANDLE animHandle, HI_S64* currentTime);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入
currentTime	动画当前已运行时间指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_TweenAnimSetDelay

【描述】

设置动画延迟运行的时间。

【语法】

```
HI_S32 HI_GV_TweenAnimSetDelay(HIGV_HANDLE animHandle, HI_U32 ms);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入
ms	时间，单位：ms。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_TweenAnimSetLoops

【描述】

设置动画循环的次数。

【语法】

```
HI_S32 HI_GV_TweenAnimSetLoops(HIGV_HANDLE animHandle, HI_U32 loopCount);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入
loopCount	循环次数。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_TweenAnimSetStartedListener

【描述】

设置动画启动的回调函数。

【语法】

```
HI_S32 HI_GV_TweenAnimSetStartedListener(HIGV_HANDLE animHandle, higv_notify_func_t func);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入
func	回调函数。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h

- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_TweenAnimSetFinishedListener

【描述】

设置动画完成的回调函数。

【语法】

```
HI_S32 HI_GV_TweenAnimSetFinishedListener(HIGV_HANDLE animHandle, higv_notify_func_t func);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入
func	回调函数。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_TweenAnimSetExecListener

【描述】

设置动画正在运行时的回调函数。

【语法】

```
HI_S32 HI_GV_TweenAnimSetExecListener(HIGV_HANDLE animHandle, higv_notify_func_t func);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入
func	回调函数。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_TweenAnimStart

【描述】

启动动画。

【语法】

```
HI_S32 HI_GV_TweenAnimStart(HIGV_HANDLE animHandle);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_TweenAnimStop

【描述】

停止动画。

【语法】

```
HI_S32 HI_GV_TweenAnimStop(HIGV_HANDLE animHandle);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_TweenAnimResume

【描述】

恢复动画。

【语法】

```
HI_S32 HI_GV_TweenAnimResume(HIGV_HANDLE animHandle);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_TweenAnimPause

【描述】

暂停动画。

【语法】

```
HI_S32 HI_GV_TweenAnimPause(HIGV_HANDLE animHandle);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_TweenAnimAddTween

【描述】

设置补间动画属性。

【语法】

```
HI_S32 HI_GV_TweenAnimAddTween(HIGV_HANDLE animHandle,
                               HIGV_TWEEN_TRANSITION_E transition,
                               HIGV_TWEEN_EASE_E easing,
                               HI_FLOAT initialValue,
                               HI_FLOAT endValue);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入
transition	动画风格。	输入
easing	动画淡入淡出属性。	输入
initialValue	初始值。	输入
endValue	结束值。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_TweenAnimGetTweenValue

【描述】

获取补间动画值。

【语法】

```
HI_FLOAT HI_GV_TweenAnimGetTweenValue(HIGV_HANDLE animHandle, HI_U32 index);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入
index	索引。	输入

【返回值】

返回值	描述
浮点值	索引动画数值。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

index 参数为补间动画类型的索引，第一次调用 HI_GV_TweenAnimAddTween 接口 index 为 0，第二次调用 HI_GV_TweenAnimAddTween 接口 index 为 1，以后依次增加 1。

【举例】

无。

HI_GV_TweenAnimSetTweenRange

【描述】

设置补间动画范围。

【语法】

```
HI_S32 HI_GV_TweenAnimSetTweenRange(HIGV_HANDLE animHandle, HI_U32 index,
HI_FLOAT initialValue, HI_FLOAT endValue);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入
index	索引。	输入
initialValue	初始值	输入
endValue	结束值。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

index 参数为补间动画类型的索引，第一次调用 HI_GV_TweenAnimAddTween 接口 index 为 0，第二次调用 HI_GV_TweenAnimAddTween 接口 index 为 1，以后依次增加 1。

【举例】

无。

HI_GV_TweenAnimDestroy

【描述】

销毁补间动画实例。

【语法】

```
HI_S32 HI_GV_TweenAnimDestroy(HIGV_HANDLE animHandle);
```

【参数】

参数名称	描述	输入/输出
animHandle	动画句柄。	输入

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Anim_CreateInstance

【描述】

创建补间动画实例。

【语法】

```
HI_S32 HI_GV_Anim_CreateInstance(HIGV_HANDLE animInfoHandle, HIGV_HANDLE widgetHandle, HIGV_HANDLE * instanceHandle);
```

【参数】

参数名称	描述	输入/输出
animInfoHandle	动画信息句柄(在 XML 资源里面定义的)。	输入
widgetHandle	控件句柄	输入
instanceHandle	动画实例句柄指针	输出

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

- 同一个窗口内，最多支持创建 512 个动画实例，包含 XML 创建和接口动态创建。
- 开发者调用 HI_GV_Anim_CreateInstance 创建的资源，使用结束后，由开发者主动调用 HI_GV_Anim_DestroyInstance 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_Anim_DestroyInstance

【描述】

销毁补间动画实例。

【语法】

```
HI_S32 HI_GV_Anim_DestroyInstance(HIGV_HANDLE windowHandle, HIGV_HANDLE widgetHandle);
```

【参数】

参数名称	描述	输入/输出
windowHandle	动画所在窗口句柄	输入
widgetHandle	要销毁的动画实例句柄	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Anim_GetInfo

【描述】

获取动画实例相关联信息。

【语法】

```
HI_S32 HI_GV_Anim_GetInfo(HIGV_HANDLE windowHandle, HIGV_HANDLE animHandle,
HIGV_HANDLE * animInfoHandle, HIGV_HANDLE *widgetHandle);
```

【参数】

参数名称	描述	输入/输出
windowHandle	动画所在窗口句柄	输入
animHandle	动画实例句柄	输入
animInfoHandle	动画信息句柄	输出
widgetHandle	与动画实例绑定的控件句柄	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Anim_Start

【描述】

开始动画。

【语法】

```
HI_S32 HI_GV_Anim_Start(HIGV_HANDLE windowHandle, HIGV_HANDLE animHandle);
```

【参数】

参数名称	描述	输入/输出
windowHandle	动画实例所属窗口句柄。	输入
animHandle	动画实例句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Anim_Stop

【描述】

停止动画。

【语法】

```
HI_S32 HI_GV_Anim_Stop(HIGV_HANDLE windowHandle, HIGV_HANDLE animHandle);
```

【参数】

参数名称	描述	输入/输出
windowHandle	动画实例所属窗口句柄。	输入
animHandle	动画实例句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_animation.h、hi_type.h

- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.1.9.2 数据类型

相关数据类型、数据结构定义如下：

- HIGV_TWEEN_TRANSITION_E：动画类型。
- HIGV_TWEEN_EASE_E：淡入淡出类型。

HIGV_TWEEN_TRANSITION_E

【说明】

动画类型。

【定义】

```
typedef enum
{
    HIGV_TWEEN_TRANSITION_LINEAR,
    HIGV_TWEEN_TRANSITION_SINE,
    HIGV_TWEEN_TRANSITION_QUINT,
    HIGV_TWEEN_TRANSITION_QUART,
    HIGV_TWEEN_TRANSITION_EXPO,
    HIGV_TWEEN_TRANSITION_ELASTIC,
    HIGV_TWEEN_TRANSITION_CUBIC,
    HIGV_TWEEN_TRANSITION_BOUNCE,
    HIGV_TWEEN_TRANSITION_BACK,
    HIGV_TWEEN_TRANSITION_BUTT
}HIGV_TWEEN_TRANSITION_E;
```

【成员】

成员名称	描述
HIGV_TWEEN_TRANSITION_LINEAR	线性运动。

成员名称	描述
HIGV_TWEEN_TRANSITION_SINE	正弦运动。
HIGV_TWEEN_TRANSITION_QUINT	Quint 运动。
HIGV_TWEEN_TRANSITION_QUART	Quart 运动。
HIGV_TWEEN_TRANSITION_EXPO	Expo 运动。
HIGV_TWEEN_TRANSITION_ELASTIC	Elastic 运动。
HIGV_TWEEN_TRANSITION_CUBIC,	Cubic 运动。
HIGV_TWEEN_TRANSITION_BOUNCE	Bounce 运动。
HIGV_TWEEN_TRANSITION_BACK	Back 运动。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_TWEEN_EASE_E

【说明】

淡入淡出类型。

【定义】

```
typedef enum
{
    HIGV_TWEEN_EASE_IN,
    HIGV_TWEEN_EASE_OUT,
    HIGV_TWEEN_EASE_IN_OUT,
    HIGV_TWEEN_EASE_BUTT
}HIGV_TWEEN_EASE_E;
```

【成员】

成员名称	描述
HIGV_TWEEN_EASE_IN	加速。
HIGV_TWEEN_EASE_OUT	减速。
HIGV_TWEEN_EASE_IN_OUT	先加速再减速。

【注意事项】

无。

【相关数据类型及接口】

无。

2.1.10 触摸手势

2.1.10.1 接口描述

触摸手势模块提供以下 API：

- HI_GV_Gesture_RegisterDevice：触摸设备注册接口。
- HI_GV_Gesture_SetTouchHook：设置触摸回调钩子函数。
- HI_GV_Gesture_SetScrollFTV：设置首次触发滑动事件的阈值。
- HI_GV_Gesture_GetScrollFTV：获取首次触发滑动事件的阈值。
- HI_GV_Gesture_SetScrollTV：设置非首次触发滑动事件的阈值。
- HI_GV_Gesture_GetScrollTV：获取非首次触发滑动事件的阈值
- HI_GV_Gesture_Enable：设置触摸可用或禁用接口。
- HI_GV_Gesture_IsEnabled：判断触摸是否可用接口。

HI_GV_Gesture_RegisterDevice

【描述】

触摸设备注册接口。

【语法】

```
HI_S32 HI_GV_Gesture_RegisterDevice(const HIGV_TOUCH_DEVICE_INPUT_S * inputDevice);
```

【参数】

参数名称	描述	输入/输出
inputDevice	触摸设备注册结构体指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参考错误码。

【需求】

- 头文件：hi_gv_gesture.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Gesture_SetTouchHook

【描述】

设置触摸回调钩子函数。

【语法】

```
HI_S32 HI_GV_Gesture_SetTouchHook(PTR_TOUCHHOOK_CallBack touchCallBack);
```

【参数】

参数名称	描述	输入/输出
touchCallBack	钩子函数。	输入

【返回值】

返回值	描述
0	成功。
非 0	参考错误码。

【需求】

- 头文件：hi_gv_gesture.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Gesture_SetScrollFTV

【描述】

设置首次触发滑动事件的阈值。

【语法】

```
HI_S32 HI_GV_Gesture_SetScrollFTV(HI_U32 scrollThreshold);
```

【参数】

参数名称	描述	输入/输出
scrollThreshold	触发事件的阈值。	输入

【返回值】

返回值	描述
0	成功。
非 0	参考错误码。

【需求】

- 头文件：hi_gv_gesture.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Gesture_GetScrollFTV

【描述】

获取首次触发滑动事件的阈值。

【语法】

```
HI_S32 HI_GV_Gesture_GetScrollFTV(HI_U32* scrollThreshold);
```

【参数】

参数名称	描述	输入/输出
scrollThreshold	触发事件的阈值指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参考错误码。

【需求】

- 头文件：hi_gv_gesture.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Gesture_SetScrollTV

【描述】

设置非首次触发滑动事件的阈值。

【语法】

```
HI_S32 HI_GV_Gesture_SetScrollTV(HI_U32 scrollThreshold);
```

【参数】

参数名称	描述	输入/输出
scrollThreshold	触发事件的阈值。	输入

【返回值】

返回值	描述
0	成功。
非 0	参考错误码。

【需求】

- 头文件：hi_gv_gesture.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Gesture_GetScrollTV

【描述】

获取非首次触发滑动事件的阈值。

【语法】

```
HI_S32 HI_GV_Gesture_GetScrollTV(HI_U32* scrollThreshold);
```

【参数】

参数名称	描述	输入/输出
scrollThreshold	触发事件的阈值指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参考错误码。

【需求】

- 头文件：hi_gv_gesture.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Gesture_Enable

【描述】

设置触摸可用或禁用接口。

【语法】


```
HI_S32 HI_GV_Gesture_Enable(HI_BOOL isEnable);
```

【参数】

参数名称	描述	输入/输出
isEnable	设置触摸是否可用。 HI_TRUE：可用； HI_FALSE：禁用。	输入

【返回值】

返回值	描述
HI_SUCCESS	成功。
HI_FAILURE	失败。

【需求】

- 头文件：hi_gv_gesture.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Gesture_IsEnabled

【描述】

判断触摸是否可用接口。

【语法】

```
HI_BOOL HI_GV_Gesture_IsEnabled(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
HI_TRUE	触摸可用。
HI_FALSE	触摸禁用。

【需求】

- 头文件：hi_gv_gesture.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.1.10.2 数据类型

相关数据类型、数据结构定义如下：

- HIGV_TOUCH_E：触摸基础事件枚举。
- HIGV_TOUCH_POINT_S：触摸单点结构。
- HIGV_TOUCH_EVENT_S：触摸事件结构。
- HIGV_GESTURE_E：手势动作类型枚举。
- HIGV_GESTURE_TAP_S：轻触手势结构。
- HIGV_GESTURE_LONGTAP_S：长按手势结构。
- HIGV_GESTURE_FLING_S：轻扫手势结构。
- HIGV_GESTURE_SCROLL_S：滑动手势结构。
- HIGV_GESTURE_PINCH_S：捏合手势结构。
- HIGV_GESTURE_U：手势事件联合。
- HIGV_GESTURE_EVENT_S：手势事件结构。
- HIGV_TOUCH_INPUTINFO_S：触摸消息事件结构体。

- HIGV_TOUCH_DEVICE_INPUT_S：触摸设备注册事件结构体。
- PTR_TOUCHHOOK_CallBack：触摸钩子回调函数。

HIGV_TOUCH_E

【说明】

触摸基础事件枚举。

【定义】

```
typedef enum hiHIGV_TOUCH_E
{
    HIGV_TOUCH_START = 1,
    HIGV_TOUCH_END,
    HIGV_TOUCH_MOVE,
    HIGV_TOUCH_CANCEL,
    HIGV_TOUCH_POINTER_START,
    HIGV_TOUCH_POINTER_END,
    HIGV_TOUCH_BUTT
} HIGV_TOUCH_E;
```

【成员】

成员名称	描述
HIGV_TOUCH_START	单指触摸到屏幕事件。
HIGV_TOUCH_END	单指离开屏幕事件。
HIGV_TOUCH_MOVE	单指在屏幕上滑动事件。
HIGV_TOUCH_CANCEL	单指取消事件。
HIGV_TOUCH_POINTER_START	第二个手指触摸到屏幕事件。
HIGV_TOUCH_POINTER_END	第二个手指离开屏幕事件。

【注意事项】

HIGV_TOUCH_CANCEL 事件暂时无用。

【相关数据类型及接口】

无。

HIGV_TOUCH_POINT_S

【说明】

触摸单点结构。

【定义】

```
typedef struct hiHIGV_TOUCH_POINT_S
{
    HI_U32 id;
    HI_S32 x;
    HI_S32 y;
    HI_S32 pressure;
    HI_U64 timeStamp;
    HIGV_TOUCH_E type;
} HIGV_TOUCH_POINT_S;
```

【成员】

成员名称	描述
id	ID 序号。
x	屏幕坐标 X。
y	屏幕坐标 Y。
pressure	是否按下标志。 1：按下； 0：离开。
timeStamp	消息时间戳。
type	消息类型。

【注意事项】

无。

【相关数据类型及接口】

HIGV_TOUCH_E

HIGV_TOUCH_EVENT_S

【说明】

触摸事件结构。

【定义】

```
typedef struct hiHIGV_TOUCH_EVENT_S
{
    HIGV_TOUCH_POINT_S   last;/**<the last event*/
} HIGV_TOUCH_EVENT_S;
```

【成员】

成员名称	描述
last	最新的触摸事件。

【注意事项】

无

【相关数据类型及接口】

HIGV_TOUCH_POINT_S

HIGV_GESTURE_E

【说明】

手势动作类型枚举。

【定义】

```
typedef enum hiHIGV_GESTURE_E
{
    HIGV_GESTURE_TAP = 1,
    HIGV_GESTURE_LONGTAP,
    HIGV_GESTURE_FLING,
    HIGV_GESTURE_SCROLL,
    HIGV_GESTURE_PINCH,
    HIGV_GESTURE_BUTT
```



```
} HIGV_GESTURE_E;
```

【成员】

成员名称	描述
HIGV_GESTURE_TAP	轻触手势事件。
HIGV_GESTURE_LONGTAP	长按手势事件。
HIGV_GESTURE_FLING	轻扫手势事件。
HIGV_GESTURE_SCROLL	滑动手势事件。
HIGV_GESTURE_PINCH	捏合手势事件。

【注意事项】

HIGV_GESTURE_PINCH 事件暂时无用。

【相关数据类型及接口】

无。

HIGV_GESTURE_TAP_S

【说明】

轻触手势结构。

【定义】

```
typedef struct hiHIGV_GESTURE_TAP_S
{
    HIGV_TOUCH_POINT_S pointer;
} HIGV_GESTURE_TAP_S;
```

【成员】

成员名称	描述
pointer	轻触点信息。

【注意事项】

无。

【相关数据类型及接口】

HIGV_TOUCH_POINT_S

HIGV_GESTURE_LONGTAP_S

【说明】

长按手势结构。

【定义】

```
typedef struct hiHIGV_GESTURE_LONGTAP_S
{
    HIGV_TOUCH_POINT_S pointer;
} HIGV_GESTURE_LONGTAP_S;
```

【成员】

成员名称	描述
pointer	轻触点信息。

【注意事项】

无。

【相关数据类型及接口】

HIGV_TOUCH_POINT_S

HIGV_GESTURE_FLING_S

【说明】

轻扫手势结构。

【定义】

```
typedef struct hiHIGV_GESTURE_FLING_S
{
    HIGV_TOUCH_POINT_S start;
    HIGV_TOUCH_POINT_S end;
    HI_S32 velocityX;
```



```
    HI_S32 velocityY;
} HIGV_GESTURE_FLING_S;
```

【成员】

成员名称	描述
Start	轻扫起始点坐标消息。
End	轻扫终点坐标消息。
velocityX	X 轴方向速度。
velocityY	Y 轴方向速度。

【注意事项】

无。

【相关数据类型及接口】

HIGV_TOUCH_POINT_S

HIGV_GESTURE_SCROLL_S

【说明】

滑动手势结构。

【定义】

```
typedef struct hiHIGV_GESTURE_SCROLL_S
{
    HIGV_TOUCH_POINT_S start;
    HIGV_TOUCH_POINT_S end;
    HI_S32 distanceX;
    HI_S32 distanceY;
} HIGV_GESTURE_SCROLL_S;
```

【成员】

成员名称	描述
Start	滑动起点坐标信息。

成员名称	描述
End	滑动终点坐标信息。
distanceX	X 轴方向位移。
distanceY	Y 轴方向位移。

【注意事项】

无。

【相关数据类型及接口】

HIGV_TOUCH_POINT_S

HIGV_GESTURE_PINCH_S

【说明】

捏合手势结构。

【定义】

```
typedef struct hiHIGV_GESTURE_PINCH_S
{
    HIGV_TOUCH_POINT_S pointer1;
    HIGV_TOUCH_POINT_S pointer2;
    HI_S32 intervalX;
    HI_S32 intervalY;
} HIGV_GESTURE_PINCH_S;
```

【成员】

成员名称	描述
pointer1	第一个手指位置信息。
pointer2	第二个手指位置信息。
intervalX	两指 X 轴方向距离。
intervalY	两指 Y 轴方向距离。

【注意事项】

此事件暂时不可用。

【相关数据类型及接口】

HIGV_TOUCH_POINT_S

HIGV_GESTURE_U

【说明】

手势事件联合。

【定义】

```
typedef union hiHIGV_GESTURE_U
{
    HIGV_GESTURE_TAP_S tap;
    HIGV_GESTURE_LONGTAP_S longtap;
    HIGV_GESTURE_FLING_S fling;
    HIGV_GESTURE_SCROLL_S scroll;
    HIGV_GESTURE_PINCH_S pinch;
} HIGV_GESTURE_U;
```

【成员】

成员名称	描述
tap	轻触手势。
longtap	长按手势。
fling	轻扫手势。
scroll	滑动手势。
pinch	捏合手势。

【注意事项】

无。

【相关数据类型及接口】

- HIGV_GESTURE_TAP_S

- HIGV_GESTURE_LONGTAP_S
- HIGV_GESTURE_FLING_S
- HIGV_GESTURE_SCROLL_S
- HIGV_GESTURE_PINCH_S

HIGV_GESTURE_EVENT_S

【说明】

手势事件结构。

【定义】

```
typedef struct hiHIGV_GESTURE_EVENT_S
{
    HIGV_GESTURE_U gesture;
} HIGV_GESTURE_EVENT_S;
```

【成员】

成员名称	描述
gesture	手势事件联合。

【注意事项】

无。

【相关数据类型及接口】

HIGV_GESTURE_U

HIGV_TOUCH_INPUTINFO_S

【说明】

触摸消息事件结构体。

【定义】

```
typedef struct hiHIGV_TOUCH_INPUTINFO_S
{
    HI_S32   id;
    HI_S32    x;
    HI_S32    y;
```



```
    HI_U32    pressure;
    HI_U64    timeStamp;
} HIGV_TOUCH_INPUTINFO_S;
```

【成员】

成员名称	描述
id	ID 序号。
x	屏幕坐标 X。
y	屏幕坐标 Y。
pressure	是否按下标志。 1：按下； 0：离开。
timeStamp	消息时间戳。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_TOUCH_DEVICE_INPUT_S

【说明】

触摸设备注册事件结构体。

【定义】

```
typedef struct hiHIGV_TOUCH_DEVICE_INPUT_S
{
    HI_S32 (*OpenDevice)(HI_S32* fd);
    HI_S32 (*CloseDevie)(HI_VOID);
    HI_S32 (*ReadData)(HIGV_TOUCH_INPUTINFO_S* info);
} HIGV_TOUCH_DEVICE_INPUT_S;
```

【成员】

成员名称	描述
OpenDevice	注册打开设备函数。
CloseDevie	注册关闭设备函数。
ReadData	注册读取触摸消息事件函数。

【注意事项】

无。

【相关数据类型及接口】

无。

PTR_TOUCHHOOK_CallBack

【说明】

触摸钩子回调函数。

【定义】

```
typedef HI_S32 (*PTR_TOUCHHOOK_CallBack)(HIGV_TOUCH_POINT_S* touchEvent);
```

【成员】

成员名称	描述
touchEvent	触摸事件结构体。

【注意事项】

无。

【相关数据类型及接口】

无。

2.1.11 语言设置

2.1.11.1 接口描述

语言设置模块提供以下 API：

- HI_GV_Lan_Register：注册需要切换的语言信息。
- HI_GV_Lan_UnRegister：反注册已注册的语言信息。
- HI_GV_Lan_SetLocale：设置本地语言。
- HI_GV_Lan_FontRegister：注册多语言的字体信息。
- HI_GV_Lan_Change：切换到指定语言。
- HI_GV_Lan_GetCurLangID：找到当前语言 ID。
- HI_GV_Lan_GetLangString：根据字符串 ID，取得当前语言文本。
- HI_GV_Lan_GetLangIDDirection：获取指定语言方向。

HI_GV_Lan_Register

【描述】

注册需要切换的语言信息。

【语法】

```
HI_S32 HI_GV_Lan_Register(const HI_CHAR* resFile, HIGV_HANDLE fontHandle,
                          const HI_CHAR*    langId);
```

【参数】

参数名称	描述	输入/输出
resFile	界面中需要变换指定语言字符串指针。	输入
fontHandle	字体句柄。	输入
langId	语言标识。	输入

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_lan.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

- 语言注册函数放到加载 BIN 文件之后调用，在加载视图前调用。
- 当前语言为首次注册的语言，不是设置本地语言。

【举例】

无。

HI_GV_Lan_UnRegister

【描述】

反注册已注册的语言信息。

【语法】

```
HI_S32 HI_GV_Lan_UnRegister(const HI_CHAR*    langId);
```

【参数】

参数名称	描述	输入/输出
langId	语言标识指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_lan.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Lan_SetLocale

【描述】

设置本地语言。

【语法】

```
HI_S32 HI_GV_Lan_SetLocale(const HI_CHAR* locale);
```

【参数】

参数名称	描述	输入/输出
locale	语言标识指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_lan.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Lan_FontRegister

【描述】

注册多语言的字体信息。

【语法】

```
HI_S32 HI_GV_Lan_FontRegister(const HI_CHAR*    langId, HIGV_HANDLE fontHandle);
```

【参数】

参数名称	描述	输入/输出
langId	语言标识。	输入
fontHandle	字体句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_lan.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Lan_Change

【描述】

切换到指定语言，可改变语言的风格控件将收到 HIGV_MSG_LANCHNGE 消息。

【语法】

```
HI_S32 HI_GV_Lan_Change(const HI_CHAR*  langId);
```

【参数】

参数名称	描述	输入/输出
langId	语言标识。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_lan.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Lan_GetCurLangID

【描述】

找到当前语言 ID。

【语法】


```
HI_S32 HI_GV_Lan_GetCurLangID(HI_CHAR**  langId);
```

【参数】

参数名称	描述	输入/输出
langId	存入当前语言 ID 指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_lan.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Lan_GetLangString

【描述】

根据字符串 ID，取得当前语言文本。

【语法】

```
HI_S32 HI_GV_Lan_GetLangString(const HI_CHAR*  langId, const HI_U32 strId, HI_CHAR** str);
```

【参数】

参数名称	描述	输入/输出
langId	语言 ID 指针。	输入

参数名称	描述	输入/输出
strId	字符串 ID。	输入
str	字符串指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_lan.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Lan_GetLangIDDirection

【描述】

获取指定语言方向。

【语法】

```
HI_S32 HI_GV_Lan_GetLangIDDirection(const HI_CHAR* langId, HI_LAN_DIR* direction);
```

【参数】

参数名称	描述	输入/输出
langId	获取方向的语言指针。	输入
direction	存入当前语言方向指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_lan.h、hi_type.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.1.11.2 数据类型

相关数据类型、数据结构定义如下：

HI_LAN_DIR：语言的文本书写方向。

HI_LAN_DIR

【说明】

语言的文本书写方向。

【定义】

```
typedef enum
{
        HI_LAN_DIR_LTR = 0,
        HI_LAN_DIR_RTL = 1,
    HI_LAN_DIR_BUTT
} HI_LAN_DIR;
```

【成员】

成员名称	描述
HI_LAN_DIR_LTR	从左向右书写的语言。
HI_LAN_DIR_RTL	从右向左书写的语言。

【注意事项】

无。

【相关数据类型及接口】

无。

2.1.12 数据库管理

2.1.12.1 接口描述

数据库管理模块提供以下 API：

- HI_GV_DDB_Create：创建一个数据缓存实例。
- HI_GV_DDB_Destroy：删除一个数据缓存实例。
- HI_GV_DDB_Insert：向数据缓存中插入一个记录。
- HI_GV_DDB_Modify：修改数据缓存中的一个记录。
- HI_GV_DDB_Append：向数据缓存尾部添加一个记录。
- HI_GV_DDB_Clear：清除数据缓存中的记录。
- HI_GV_DDB_Del：根据 key 值删除一条记录。
- HI_GV_DDB_GetRow：根据行号获取一条记录。
- HI_GV_DDB_GetRows：从指定行出获取 n 行的数据。
- HI_GV_DDB_GetCellData：按行列号获取数据。
- HI_GV_DDB_SetCellData：按行列号设置单元格数据。
- HI_GV_DDB_GetRowCount：获取总行数。
- HI_GV_DDB_RegisterDataChange：注册数据变化通知函数。
- HI_GV_DDB_UnRegisterDataChange：取消数据变化通知。
- HI_GV_DDB_EnableDataChange：数据变化通知使能开关。

HI_GV_DDB_Create

【描述】

创建一个数据缓存实例。

【语法】

```
HI_S32 HI_GV_DDB_Create(HI_U32 fieldCount, const HIGV_FIELDATTR_S* fieldAttr,
HIGV_HANDLE* ddbHandle);
```

【参数】

参数名称	描述	输入/输出
fieldCount	字段个数。	输入
fieldAttr	字段属性指针。	输入
ddbHandle	数据缓存句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_ddb.h、hi_type.h、hi_gv_adm.h
- 库文件：libhigv.a、libhigv.so

【注意】

开发者调用 HI_GV_DDB_Create 创建的资源，使用结束后，由开发者主动调用 HI_GV_DDB_Destroy 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_DDB_Destroy

【描述】

删除一个数据缓存实例。

【语法】

```
HI_S32 HI_GV_DDB_Destroy(HIGV_HANDLE ddbHandle);
```

【参数】

参数名称	描述	输入/输出
ddbHandle	数据缓存句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_ddb.h、hi_type.h、hi_gv_adm.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_DDB_Insert

【描述】

向数据缓存中插入一个记录。

【语法】


```
HI_S32 HI_GV_DDB_Insert(HIGV_HANDLE ddbHandle, HI_U32 row, const HIGV_DBROW_S* data,
HIGV_DDBINSERT_E insert);
```

【参数】

参数名称	描述	输入/输出
ddbHandle	数据缓存句柄。	输入
row	标识插入数据的位置。	输入
data	记录的数据指针，注意 data->size 不能小于各列数据长度之和。	输入
insert	标识插入数据的位置。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_ddb.h、hi_type.h、hi_gv_adm.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_DDB_Modify

【描述】

修改数据缓存中的一个记录。

【语法】


```
HI_S32 HI_GV_DDB_Modify(HIGV_HANDLE ddbHandle, HI_U32 row, const HIGV_DBROW_S*
data);
```

【参数】

参数名称	描述	输入/输出
ddbHandle	数据缓存句柄。	输入
row	要修改的行号。	输入
data	记录的数据指针，注意 data->size 不能小于各列数据长度之和。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_ddb.h、hi_type.h、hi_gv_adm.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_DDB_Append

【描述】

向数据缓存尾部添加一个记录。

【语法】

```
HI_S32 HI_GV_DDB_Append(HIGV_HANDLE ddbHandle, const HIGV_DBROW_S* data);
```


【参数】

参数名称	描述	输入/输出
ddbHandle	数据缓存句柄。	输入
data	记录的数据指针，注意 data ->size 不能小于各列数据长度之和。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_ddb.h、hi_type.h、hi_gv_adm.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_DDB_Clear

【描述】

清除数据缓存中的记录。

【语法】

```
HI_S32 HI_GV_DDB_Clear(HIGV_HANDLE ddbHandle);
```

【参数】

参数名称	描述	输入/输出
ddbHandle	数据缓存句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_ddb.h、hi_type.h、hi_gv_adm.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_DDB_Del

【描述】

根据 key 值删除一条记录。

【语法】

```
HI_S32 HI_GV_DDB_Del(HIGV_HANDLE ddbHandle, HI_U32 row);
```

【参数】

参数名称	描述	输入/输出
ddbHandle	数据缓存句柄。	输入
row	要操作记录的行号。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_ddb.h、hi_type.h、hi_gv_adm.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_DDB_GetRow

【描述】

根据行号获取一条记录。

【语法】

```
HI_S32 HI_GV_DDB_GetRow(HIGV_HANDLE ddbHandle, HI_U32 row, const HIGV_DBROW_S* data);
```

【参数】

参数名称	描述	输入/输出
ddbHandle	数据缓存句柄。	输入
row	要操作记录的位置。	输入
data	记录的数据。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_ddb.h、hi_type.h、hi_gv_adm.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_DDB_GetRows

【描述】

从指定行出获取 n 行的数据。

【语法】

```
HI_S32 HI_GV_DDB_GetRows(HIGV_HANDLE ddbHandle, HI_U32 row, HI_U32 num, const HI_VOID* data, HI_U32* rowNum);
```

【参数】

参数名称	描述	输入/输出
ddbHandle	数据缓存句柄。	输入
row	要获取数据的起始行号。	输入
num	要获取的总行数。	输入
data	数据缓存指针。	输出
rowNum	实际获取的行数指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_ddb.h、hi_type.h、hi_gv_adm.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_DDB_GetCellData

【描述】

按行列号获取数据。

【语法】

```
HI_S32 HI_GV_DDB_GetCellData(HIGV_HANDLE ddbHandle, HI_U32 row, HI_U32 col, HI_VOID* fieldData, HI_U32 len);
```

【参数】

参数名称	描述	输入/输出
ddbHandle	数据缓存句柄。	输入
row	行号。	输入
col	列号。	输入
fieldData	字段 buffer 指针。	输出
len	字段 buffer 大小。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_ddb.h、hi_type.h、hi_gv_adm.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_DDB_SetCellData

【描述】

按行列号设置单元格数据。

【语法】

```
HI_S32 HI_GV_DDB_SetCellData(HIGV_HANDLE ddbHandle, HI_U32 row, HI_U32 col, const HI_VOID* fieldData, HI_U32 len);
```

【参数】

参数名称	描述	输入/输出
ddbHandle	数据缓存句柄。	输入
row	行号。	输入
col	列号。	输入
fieldData	字段 buffer。	输入

参数名称	描述	输入/输出
len	字段 buffer 大小。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_ddb.h、hi_type.h、hi_gv_adm.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_DDB_GetRowCount

【描述】

获取总行数。

【语法】

```
HI_S32 HI_GV_DDB_GetRowCount(HIGV_HANDLE ddbHandle, HI_U32* rowCount);
```

【参数】

参数名称	描述	输入/输出
ddbHandle	数据缓存句柄。	输入
rowCount	总行数指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_ddb.h、hi_type.h、hi_gv_adm.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_DDB_RegisterDataChange

【描述】

注册数据变化通知函数。

【语法】

```
HI_S32 HI_GV_DDB_RegisterDataChange(HIGV_HANDLE ddbHandle, HIGV_HANDLE admHandle);
```

【参数】

参数名称	描述	输入/输出
ddbHandle	数据缓存句柄。	输入
admHandle	需要通知的对象。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_ddb.h、hi_type.h、hi_gv_adm.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_DDB_UnRegisterDataChange

【描述】

取消数据变化通知。

【语法】

```
HI_S32 HI_GV_DDB_UnRegisterDataChange(HIGV_HANDLE ddbHandle, HIGV_HANDLE admHandle);
```

【参数】

参数名称	描述	输入/输出
ddbHandle	数据缓存句柄。	输入
admHandle	需要通知的对象。	输入

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_ddb.h、hi_type.h、hi_gv_adm.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_DDB_EnableDataChange

【描述】

数据变化通知使能开关。

【语法】

```
HI_S32 HI_GV_DDB_EnableDataChange(HIGV_HANDLE ddbHandle, HI_BOOL enable);
```

【参数】

参数名称	描述	输入/输出
ddbHandle	数据缓存句柄。	输入
enable	数据变化通知使能开关。 HI_TRUE：使能； HI_FALSE：去使能。	输入

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_ddb.h、hi_type.h、hi_gv_adm.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

2.1.12.2 数据类型

相关数据类型、数据结构定义如下：

HIGV_DDBINSERT_E：插入类型。

HIGV_DDBINSERT_E

【说明】

插入类型。

【定义】

```
typedef enum
{
    DDB_INSERT_PRE = 0,
    DDB_INSERT_NEXT,
    DDB_INSERT_BUTT
} HIGV_DDBINSERT_E;
```

【成员】

成员名称	描述
DDB_INSERT_PRE	插入该行之前。

成员名称	描述
DDB_INSERT_NEXT	插入该行之后。

【注意事项】

无。

【相关数据类型及接口】

无。

2.1.13 ADM 管理

2.1.13.1 接口描述

ADM 管理模块提供以下 API：

- HI_GV_ADM_Create：创建一个抽象数据模型实例。
- HI_GV_ADM_CreateByHandle：根据指定的 Handle 创建一个抽象数据模型实例，给 XML2Bin 使用。
- HI_GV_ADM_CreateDefault：根据默认的数据缓冲创建数据模型。
- HI_GV_ADM_CreateDefaultByHandle：通过指定的 Handle，根据默认的数据缓冲创建数据模型。
- HI_GV_ADM_Destroy：删除一个数据模型实例。
- HI_GV_ADM_Bind：将一个控件绑定到数据模型。
- HI_GV_ADM_UnBind：解除一个控件和数据模型的绑定。
- HI_GV_ADM_GetFieldAttr：获取指定字段的数据类型。
- HI_GV_ADM_GetRowCount：获取数据总行数。
- HI_GV_ADM_GetCellData：获取指定行指定列的数据。
- HI_GV_ADM_GetCellDataString：获取指定行指定列的数据，以字符串表示。
- HI_GV_ADM_SetDCCallBack：设置数据变化的回调函数。
- HI_GV_ADM_GetDDBHandle：根据 ADM 获取 DDB。
- HI_GV_ADM_GetColNum：获取列总数。
- HI_GV_ADM_Sync：ADM 与 DB 同步接口,调用此接口后以后获取的数据就是与数据库同步的。

- HI_GV_ADM_ClearAllData：清除所有 ADM 中的缓存数据。

HI_GV_ADM_Create

【描述】

创建一个抽象数据模型实例。

【语法】

```
HI_S32 HI_GV_ADM_Create(const HIGV_ADMOPT_S *dataSource, HIGV_HANDLE *admHandle);
```

【参数】

参数名称	描述	输入/输出
dataSource	数据源操作属性指针。	输入
admHandle	数据模型句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h
- 库文件：libhigv.a、libhigv.so

【注意】

开发者调用 HI_GV_ADM_Create 创建的资源，使用结束后，由开发者主动调用 HI_GV_ADM_Destroy 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_ADM_CreateByHandle

【描述】

根据指定的 Handle 创建一个抽象数据模型实例，给 XML2Bin 使用。

【语法】

```
HI_S32 HI_GV_ADM_CreateByHandle(const HIGV_ADMOPT_S *dataSource, HIGV_HANDLE admHandle);
```

【参数】

参数名称	描述	输入/输出
dataSource	数据源操作属性指针。	输入
admHandle	数据模型句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h
- 库文件：libhigv.a、libhigv.so

【注意】

开发者调用 HI_GV_ADM_CreateByHandle 创建的资源，使用结束后，由开发者主动调用 HI_GV_ADM_Destroy 来释放响应资源，否则会有内存泄露。

【举例】

无

HI_GV_ADM_CreateDefault

【描述】

根据默认的数据缓冲创建数据模型。

【语法】

```
HI_S32 HI_GV_ADM_CreateDefault(HIGV_HANDLE dbhandle, HI_U32 fieldCount,
HIGV_FIELDATTR_S *fieldAttr, HIGV_HANDLE   *admHandle);
```

【参数】

参数名称	描述	输入/输出
dbHandle	默认的数据缓冲句柄。	输入
fieldCount	字段个数。	输入
fieldAttr	字段属性指针。	输入
admHandle	数据模型句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h
- 库文件：libhigv.a、libhigv.so

【注意】

开发者调用 HI_GV_ADM_CreateDefault 创建的资源，使用结束后，由开发者主动调用 HI_GV_ADM_Destroy 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_ADM_CreateDefaultByHandle

【描述】

通过指定的 Handle，根据默认的数据缓冲创建数据模型。

【语法】

```
HI_S32 HI_GV_ADM_CreateDefaultByHandle(HIGV_HANDLE dbHandle, HI_U32 fieldCount, const HIGV_FIELDATTR_S *fieldAttr, HIGV_HANDLE admHandle);
```

【参数】

参数名称	描述	输入/输出
dbHandle	默认的数据缓冲句柄。	输入
fieldCount	字段个数。	输入
fieldAttr	字段属性指针。	输入
admHandle	数据模型句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h
- 库文件：libhigv.a、libhigv.so

【注意】

开发者调用 HI_GV_ADM_CreateDefaultByHandle 创建的资源，使用结束后，由开发者主动调用 HI_GV_ADM_Destroy 来释放响应资源，否则会有内存泄露。

【举例】

无

HI_GV_ADM_Destroy

【描述】

删除一个数据模型实例。

【语法】

```
HI_S32 HI_GV_ADM_Destroy(HIGV_HANDLE admHandle);
```

【参数】

参数名称	描述	输入/输出
admHandle	数据模型句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ADM_Bind

【描述】

将一个控件绑定到数据模型。

【语法】

```
HI_S32 HI_GV_ADM_Bind(HIGV_HANDLE widgetHandle, HIGV_HANDLE admHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
admHandle	数据模型句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_ADM_UnBind

【描述】

解除一个控件和数据模型的绑定。

【语法】

```
HI_S32 HI_GV_ADM_UnBind(HIGV_HANDLE widgetHandle, HIGV_HANDLE admHandle);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	控件句柄。	输入
admHandle	数据模型句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_ADM_GetFieldAttr

【描述】

获取指定字段的数据类型。

【语法】

```
HI_S32 HI_GV_ADM_GetFieldAttr(HIGV_HANDLE admHandle, HI_U32 col, HIGV_FIELDATTR_S *fieldAttr);
```

【参数】

参数名称	描述	输入/输出
admHandle	数据模型句柄。	输入
col	列号。	输入
fieldAttr	字段属性指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_ADM_GetRowCount

【描述】

获取数据总行数。

【语法】

```
HI_S32 HI_GV_ADM_GetRowCount(HIGV_HANDLE admHandle, HI_U32 *count);
```

【参数】

参数名称	描述	输入/输出
admHandle	数据模型句柄。	输入
count	数据总数指针。	输出

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_ADM_GetCellData

【描述】

获取指定行指定列的数据。

【语法】

```
HI_S32 HI_GV_ADM_GetCellData(HIGV_HANDLE admHandle, HI_U32 row, HI_U32 col, HI_VOID *data, HI_U32 len);
```

【参数】

参数名称	描述	输入/输出
admHandle	数据模型句柄。	输入
row	行号。	输入
col	列号。	输入
data	数据内容指针。	输出
len	所能容纳的字节数。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_ADM_GetCellDataString

【描述】

获取指定行指定列的数据，以字符串表示。

【语法】

```
HI_S32 HI_GV_ADM_GetCellDataString(HIGV_HANDLE admHandle, HI_U32 row, HI_U32 col, HI_CHAR *dataString, HI_U32 len);
```

【参数】

参数名称	描述	输入/输出
admHandle	数据模型句柄。	输入
row	行号。	输入
col	列号。	输入
dataString	数据内容指针。	输出
len	所能容纳的字节数。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_ADM_SetDCCallBack

【描述】

设置数据变化的回调函数。

【语法】

```
HI_S32 HI_GV_ADM_SetDCCallBack(HIGV_HANDLE admHandle, HIGV_MSG_PROC customProc, HIGV_PROCORDER_E procOrder);
```

【参数】

参数名称	描述	输入/输出
admHandle	数据模型句柄。	输入
customProc	数据变化通知回调函数。	输入
procOrder	回调的时间选择。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ADM_GetDDBHandle

【描述】

根据 ADM 获取 DDB。

【语法】

```
HI_S32 HI_GV_ADM_GetDDBHandle(HIGV_HANDLE admHandle, HIGV_HANDLE *ddbHandle);
```

【参数】

参数名称	描述	输入/输出
admHandle	数据模型句柄。	输入
ddbHandle	DDB 句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_ADM_GetColNum

【描述】

获取列总数。

【语法】

```
HI_S32 HI_GV_ADM_GetColNum(HIGV_HANDLE admHandle, HI_U32 *colNum);
```

【参数】

参数名称	描述	输入/输出
admHandle	数据模型句柄。	输入
colNum	列总数指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h

- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_ADM_Sync

【描述】

ADM 与 DB 同步接口,调用此接口后以后获取的数据就是与数据库同步的。

【语法】

```
HI_S32 HI_GV_ADM_Sync(HIGV_HANDLE admHandle);
```

【参数】

参数名称	描述	输入/输出
admHandle	数据模型句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_ADM_ClearAllData

【描述】

清除所有 ADM 中的缓存数据。

【语法】

```
HI_S32 HI_GV_ADM_ClearAllData(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_adm.h、hi_type.h、hi_gv_widget.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

2.1.13.2 数据类型

相关数据类型、数据结构定义如下：

- HIGV_DT_E：支持的数据类型定义。
- HIGV_DC_E：数据变化的类型。
- HIGV_DCINFO_S：数据变化信息。
- HIGV_FIELDATTR_S：字段属性。

- HIGV_DBROW_S：行数据存储。
- HIGV_ADMOPT_S：ADM 信息结构。

HIGV_DT_E

【说明】

支持的数据类型定义。

【定义】

```
typedef enum
{
    HIGV_DT_S8 = 0,
    HIGV_DT_U8,
    HIGV_DT_S16,
    HIGV_DT_U16,
    HIGV_DT_S32,
    HIGV_DT_U32,
    HIGV_DT_S64,
    HIGV_DT_U64,
    HIGV_DT_F32,
    HIGV_DT_D64,
    HIGV_DT_STRING,
    HIGV_DT_HIMAGE,
    HIGV_DT_STRID,
    HIGV_DT_BUTT
} HIGV_DT_E;
```

【成员】

成员名称	描述
HIGV_DT_S8	char 类型。
HIGV_DT_U8	unsigned char 类型。
HIGV_DT_S16	short 类型。
HIGV_DT_U16	unsigned short 类型。
HIGV_DT_S32	int 类型。

成员名称	描述
HIGV_DT_U32	unsigned int 类型。
HIGV_DT_S64	long long 类型。
HIGV_DT_U64	unsigned long long 类型。
HIGV_DT_F32	float 类型。
HIGV_DT_D64	double 类型。
HIGV_DT_STRING	char*类型。
HIGV_DT_HIMAGE	image handle 类型。
HIGV_DT_STRID	multi-langugae string ID 类型。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_DC_E

【说明】

数据变化的类型。

【定义】

```
typedef enum
{
    HIGV_DC_INSERT = 0,
    HIGV_DC_MODIFY,
    HIGV_DC_DELETE,
    HIGV_DC_UPDATE,
    HIGV_DC_CLEAR,
    HIGV_DC_DESTROY,
    HIGV_DC_BUTT
} HIGV_DC_E;
```


【成员】

成员名称	描述
HIGV_DC_INSERT	插入。
HIGV_DC_MODIFY	修改。
HIGV_DC_DELETE	删除。
HIGV_DC_UPDATE	数据变化，操作未知。
HIGV_DC_CLEAR	数据全部销毁。
HIGV_DC_DESTROY	数据源销毁。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_DCINFO_S

【说明】

数据变化信息。

【定义】

```
typedef struct
{
    HIGV_DC_E Action;
    HI_U16 StartRow;
    HI_U16 Rows;
} HIGV_DCINFO_S;
```

【成员】

成员名称	描述
Action	变化动作描述。

成员名称	描述
StartRow	变化的起始行。
Rows	变化的总行数。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_FIELDATTR_S

【说明】

字段属性。

【定义】

```
typedef struct hiHIGV_CELLATTR_S
{
    HIGV_DT_E eDataType;
    HI_U32 MaxSize;
} HIGV_FIELDATTR_S;
```

【成员】

成员名称	描述
eDataType	数据类型。
MaxSize	最大长度，单位：个数。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_DBROW_S

【说明】

行数据存储。

【定义】

```
typedef struct hiHIGV_DBROW_S
{
    HI_U32 Size;
    HI_VOID* pData;
} HIGV_DBROW_S;
```

【成员】

成员名称	描述
Size	数据长度，单位：字节。
pData	数据。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_ADMOPT_S

【说明】

ADM 信息结构。

【定义】

```
typedef struct hiADM_OPT_S
{
    HI_U32 DBSource;
    HI_U32 FieldCount;
    HIGV_FIELDATTR_S* pFieldAttr;
    HI_U32 BufferRows;
    GetCountFunc GetCount;
```



```
    GetRowValueFunc GetRowValue;
    RegisterDataChangeFunc RegisterDataChange;
    UnregisterDataChangeFunc UnregisterDataChange;
} HIGV_ADMOPT_S;
```

【成员】

成员名称	描述
DBSource	数据源标识。
FieldCount	数据源的字段总数。
pFieldAttr	数据源的每个字段属性。
BufferRows	希望的缓冲行数。
GetCount	获取总数接口。
GetRowValue	从指定行处获取 n 行数据。
RegisterDataChange	注册数据变化通知接口。
UnregisterDataChange	取消数据变化通知接口。

【注意事项】

暂时只支持普通输入事件

【相关数据类型及接口】

无。

2.1.14 输入设备

2.1.14.1 接口描述

输入设备模块提供以下 API：

HI_GV_SendInputEvent：送入输入事件，如果没有实现 GetInputEvent，则使用此接口送入输入事件。

HI_GV_SendInputEvent

【描述】

送入输入事件，如果没有实现 GetInputEvent，则使用此接口送入输入事件。

【语法】

```
HI_S32 HI_GV_SendInputEvent(const HIGV_INPUTEVENT_S* inputEvent);
```

【参数】

参数名称	描述	输入/输出
inputEvent	输入事件信息。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_input.h、hi_type.h、hi_go.h、hi_gv_gesture.h
- 库文件：libhigv.a、libhigv.so

【注意】

暂时只完整支持的按下和抬起事件。

【举例】

无

2.1.14.2 数据类型

相关数据类型、数据结构定义如下：

HIGV_INPUTEVENT_S：输入设备事件（鼠标/键盘）。

HIGV_INPUTEVENT_S

【说明】

输入设备事件（鼠标/键盘）。

【定义】

```
typedef struct
{
    HI_U32 msg;
    HI_U32 value;
    HI_S32 dx;
    HI_S32 dy;
    HI_S32 step;
} HIGV_INPUTEVENT_S;
```

【成员】

成员名称	描述
msg	鼠标按下，松开，滚动，双击等事件,或者键盘按下，松开等。
value	键值或者鼠标左，中，右键。
dx	鼠标移动偏移量 X，相对上次的值。
dy	鼠标移动偏移量 Y，相对上次的值。
step	鼠标滚动偏移量，相对上次的值。

【注意事项】

无。

【相关数据类型及接口】

无。

2.1.15 光标管理

2.1.15.1 接口描述

光标管理模块提供以下 API：

- HI_GV_CURSOR_Init：光标初始化。
- HI_GV_CURSOR_DeInit：光标去初始化。

- HI_GV_CURSOR_Show：显示光标。
- HI_GV_CURSOR_Hide：隐藏光标。
- HI_GV_CURSOR_SetMapScale：设置光标映射比例。
- HI_GV_CURSOR_SetThreshold：设置光标加速阈值参数。
- HI_GV_CURSOR_SetAcceleration：设置光标加速倍数。
- HI_GV_CURSOR_SetScreenPos：移动光标到屏幕的指定位置。
- HI_GV_CURSOR_GetScreenPos：获取光标的屏幕坐标。
- HI_GV_CURSOR_SetImage：设备光标的显示图片。
- HI_GV_CURSOR_GetImage：获取光标的图片显示信息。
- HI_GV_CURSOR_GetOutputDev：获取光标的显示输出设备。
- HI_GV_CURSOR_SetCursorRegion：设置光标的移动范围。
- HI_GV_CURSOR_SetDisplayScreen：设置光标物理输出屏幕的宽和高。
- HI_GV_CURSOR_AttchLayer：将光标绑定到图层上。
- HI_GV_CURSOR_DettchLayer：将光标从指定图层上解绑定。
- HI_GV_CURSOR_SetAllwaysShow：设置是否一直显示光标。

HI_GV_CURSOR_Init

【描述】

光标初始化。

【语法】

```
HI_S32 HI_GV_CURSOR_Init(HigvCursorType cursorType);
```

【参数】

参数名称	描述	输入/输出
cursorType	光标类型，硬光标或软光标。	输入

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

硬光标和软光标的区别在于光标的绘制是否在独立的图层上面，在支持多图层的芯片上可以配置一个图层为光标层，而单图层的芯片只能使用软光标模式。

【举例】

软光标模式开发参考：

```
HI_S32 ret;
HI_RESID retId;
HIGV_HANDLE imageHandle;
HigvCursorInfo cursorInfo = {0};

(HI_VOID)HI_GV_CURSOR_Init(HIGV_CURSOR_SW);

ret = HI_GV_Res_CreateID("./res/pic/mouse.png", HIGV_RESTYPE_IMG, &g_resId);
if (ret != HI_SUCCESS) {
printf("HI_GV_CURSOR_SetImage failed! Return: 0x%x\n", ret);
return ret;
}
ret = HI_GV_Res_GetResInfo(g_resId, &imageHandle);
if (ret != HI_SUCCESS) {
printf("HI_GV_CURSOR_SetImage failed! Return: 0x%x\n", ret);
return ret;
}
cursorInfo.imageHandle = imageHandle;

(HI_VOID)HI_GV_CURSOR_SetCursorRegion(0, 0, SCREEN_WIDTH_VGA, SCREEN_HEIGHT_VGA);

ret = HI_GV_CURSOR_SetImage(&cursorInfo);
```



```
if (ret != HI_SUCCESS) {
printf("HI_GV_CURSOR_SetImage failed! Return: 0x%x\n", ret);
return ret;
}
(HI_VOID)HI_GV_CURSOR_Show();
```

HI_GV_CURSOR_DeInit

【描述】

光标去初始化。

【语法】

```
HI_S32 HI_GV_CURSOR_DeInit(HI_VOID);
```

【参数】

无

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

无

【举例】

无

HI_GV_CURSOR_Show

【描述】

显示光标。

【语法】

```
HI_S32 HI_GV_CURSOR_Show(HI_VOID);
```

【参数】

无

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

无

【举例】

无

HI_GV_CURSOR_Hide

【描述】

隐藏光标。

【语法】

```
HI_S32 HI_GV_CURSOR_Hide(HI_VOID);
```

【参数】

无

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

无

【举例】

无

HI_GV_CURSOR_SetMapScale

【描述】

设置光标映射比例。

例如映射图层 1024*768 到 720*576，则 Scale_X = 720/1024, Scale_Y = 576/768

【语法】

```
HI_S32 HI_GV_CURSOR_SetMapScale(HI_FLOAT scaleX, HI_FLOAT scaleY);
```

【参数】

参数名称	描述	输入/输出
scaleX	X 轴映射比例。	输入
scaleY	Y 轴映射比例。	输入

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

仅硬光标模式支持。

【举例】

无

HI_GV_CURSOR_SetThreshold

【描述】

设置光标加速阀值参数，加速阀值必须大于等于 3，默认值 4。

【语法】

```
HI_S32 HI_GV_CURSOR_SetThreshold(HI_U16 threshold);
```

【参数】

参数名称	描述	输入/输出
threshold	光标加速阈值。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

无

【举例】

无

HI_GV_CURSOR_SetAcceleration

【描述】

设置光标加速倍数。

【语法】

```
HI_S32 HI_GV_CURSOR_SetAcceleration(HI_U16 acceleration);
```

【参数】

参数名称	描述	输入/输出
acceleration	光标加速倍数。必须大于 0，默认值：4。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

无

【举例】

无

HI_GV_CURSOR_SetScreenPos

【描述】

移动光标到指定屏幕指定位置。

【语法】

```
HI_S32 HI_GV_CURSOR_SetScreenPos(HIGV_CORD screenPosX, HIGV_CORD screenPosY);
```

【参数】

参数名称	描述	输入/输出
screenPosX	屏幕 X 坐标，以像素为单位。	输入
screenPosY	屏幕 Y 坐标，以像素为单位。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

无

【举例】

无

HI_GV_CURSOR_GetScreenPos

【描述】

获取光标的屏幕坐标

【语法】

```
HI_S32 HI_GV_CURSOR_GetScreenPos(HIGV_CORD * screenPosX, HIGV_CORD * screenPosY);
```

【参数】

参数名称	描述	输入/输出
screenPosX	屏幕 X 坐标指针，以像素为单位。	输出
screenPosY	屏幕 Y 坐标指针，以像素为单位。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

无

【举例】

无

HI_GV_CURSOR_SetImage

【描述】

设置光标的显示图片

【语法】

```
HI_S32 HI_GV_CURSOR_SetImage(const HigvCursorInfo *cursorInfo);
```

【参数】

参数名称	描述	输入/输出
cursorInfo	光标图片句柄指针，热点信息。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

热点坐标仅在硬光标模式下使用，软光标模式设置图片解码后句柄即可。

【举例】

无

HI_GV_CURSOR_GetImage

【描述】

获取光标的显示图片信息。

【语法】

```
HI_S32 HI_GV_CURSOR_GetImage(HigvCursorInfo *cursorInfo);
```

【参数】

参数名称	描述	输入/输出
cursorInfo	光标图片句柄指针，热点信息。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

无

【举例】

无

HI_GV_CURSOR_GetOutputDev

【描述】

获取光标显示输出设备。

【语法】

```
HI_S32 HI_GV_CURSOR_GetOutputDev(HI_U32 *devID);
```

【参数】

参数名称	描述	输入/输出
devID	输出设备 ID 指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

仅硬光标模式支持。

【举例】

无

HI_GV_CURSOR_SetCursorRegion

【描述】

设置光标移动的范围。

【语法】

```
HI_S32 HI_GV_CURSOR_SetCursorRegion(HI_S32 regionX, HI_S32 regionY, HI_S32 regionWidth, HI_S32 regionHeight);
```

【参数】

参数名称	描述	输入/输出
regionX	X 坐标。	输入
regionY	Y 坐标。	输入
regionWidth	宽度，单位：像素。	输入
regionHeight	高度，单位：像素。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

无

【举例】

无

HI_GV_CURSOR_SetDisplayScreen

【描述】

设置光标物理输出屏幕的宽和高。

【语法】

```
HI_S32 HI_GV_CURSOR_SetDisplayScreen(HI_U32 devID, HI_U32 screenWidth, HI_U32 screenHight);
```

【参数】

参数名称	描述	输入/输出
devID	光标设备 ID。	输入
screenWidth	光标实际输出屏幕的宽，用户可见像素数。	输入
screenHight	光标实际输出屏幕的高，用户可见像素数。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

仅硬光标模式支持，GUI 内部用于绘制的图层 surface 的宽度和高度与屏幕实际可见高宽可能并不相同。

例如:进行 PAL 和 NTSCL 转换时，GUI 内部图层 surface 是 720x576，输出时屏幕实际要求是 720x480，最终用户看到的是由 720x576 通过缩放处理成 720x480 的图像，光标的位置，也要根据比例进行相应的处理。

【举例】

无

HI_GV_CURSOR_AttchLayer

【描述】

将光标绑定到图层上。

【语法】

```
HI_S32 HI_GV_CURSOR_AttchLayer(HIGO_LAYER_E layerId);
```

【参数】

参数名称	描述	输入/输出
layerId	图层 id。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

- 仅硬光标模式支持。
- 光标在显示到图层之前，必须要先绑定，在绑定图层时，光标的活动显示范围基于最后那次绑定的图层绘制 buffer 的范围。

【举例】

无

HI_GV_CURSOR_DettchLayer

【描述】

将光标从指定图层上解绑定。

【语法】

```
HI_S32 HI_GV_CURSOR_DettchLayer(HIGO_LAYER_E layerId);
```

【参数】

参数名称	描述	输入/输出
layerId	图层 id。	输入

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

仅硬光标模式支持。

【举例】

无

HI_GV_CURSOR_SetAllwaysShow

【描述】

设置是否一直显示光标(即使光标未插入)。

【语法】

```
HI_VOID HI_GV_CURSOR_SetAllwaysShow(const HI_BOOL isCursorShow);
```

【参数】

参数名称	描述	输入/输出
isCursorShow	设置光标是否一直显示。如果为 HI_TRUE，则光标一直显示(即使光标未插入)，否则仅当光标插入时才显示光标。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_cursor.h、hi_type.h
- 库文件：libhigv.a

【注意】

仅硬光标模式支持。

【举例】

无

2.1.15.2 数据类型

相关数据类型、数据结构定义如下：

- HigvCursorInfo：光标信息结构。
- HigvCursorType：光标类型。

HigvCursorInfo

【说明】

光标信息结构。

【定义】

```
typedef struct {
    HIGV_HANDLE imageHandle;
    HI_U32 hotspotX;
    HI_U32 hotspotY;
} HigvCursorInfo;
```

【成员】

成员名称	描述
imageHandle	光标图片解码后的句柄。范围是[16*16，64*64]。
hotspotX	X 热点坐标。
hotspotY	Y 热点坐标。

【注意事项】

热点坐标仅在硬光标模式下使用，软光标模式可忽略。

【相关数据类型及接口】

无。

HigvCursorType

【说明】

光标模式。

【定义】

```
typedef enum {
    HIGV_CURSOR_HW = 0x1,
    HIGV_CURSOR_SW,
    HIGV_CURSOR_BUTT
} HigvCursorType;
```

【成员】

成员名称	描述
HIGV_CURSOR_HW	硬光标模式。
HIGV_CURSOR_SW	软光标模式。

【注意事项】

无

【相关数据类型及接口】

无

2.1.16 定时器

2.1.16.1 接口描述

定时器模块提供以下 API：

- HI_GV_Timer_Create：创建定时器。
- HI_GV_Timer_Destroy：销毁定时器。
- HI_GV_Timer_Start：启动定时器。
- HI_GV_Timer_Stop：停止定时器。
- HI_GV_Timer_Reset：复位定时器。
- HI_GV_Timer_SetSpeed：设置定时器速度。

HI_GV_Timer_Create

【描述】

创建定时器。

【语法】

```
HI_S32 HI_GV_Timer_Create(HIGV_HANDLE widgetHandle, HI_U32 timerId, HI_U32 speed);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	窗口或控件句柄。	输入
timerId	定时器 ID。	输入
speed	定时器间隔，以 ms 为单位。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_gv_timer.h
- 库文件：libhigv.a、libhigv.so

【注意】

开发者调用 HI_GV_Timer_Create 创建的资源，使用结束后，由开发者主动调用 HI_GV_Timer_Destroy 来释放响应资源，否则会有内存泄露。

【举例】

无

HI_GV_Timer_Destroy

【描述】

销毁定时器。

【语法】

```
HI_S32 HI_GV_Timer_Destroy(HIGV_HANDLE widgetHandle, HI_U32 timerId);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	窗口或控件句柄。	输入
timerId	定时器 ID。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_gv_timer.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_Timer_Start

【描述】

启动定时器。

【语法】

```
HI_S32 HI_GV_Timer_Start(HIGV_HANDLE widgetHandle, HI_U32 timerId);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	窗口或控件句柄。	输入
timerId	定时器 ID。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_gv_timer.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_Timer_Stop

【描述】

停止定时器。

【语法】

```
HI_S32 HI_GV_Timer_Stop(HIGV_HANDLE widgetHandle, HI_U32 timerId);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	窗口或控件句柄。	输入
timerId	定时器 ID。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_gv_timer.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_Timer_Reset

【描述】

复位定时器，定时器重现开始计时。

【语法】

```
HI_S32 HI_GV_Timer_Reset(HIGV_HANDLE widgetHandle, HI_U32 timerId);
```


【参数】

参数名称	描述	输入/输出
widgetHandle	窗口或控件句柄。	输入
timerId	定时器 ID。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_gv_timer.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Timer_SetSpeed

【描述】

设置定时器速度。

【语法】

```
HI_S32 HI_GV_Timer_SetSpeed(HIGV_HANDLE widgetHandle, HI_U32 timerId, HI_U32 speed);
```

【参数】

参数名称	描述	输入/输出
widgetHandle	窗口或控件句柄。	输入

参数名称	描述	输入/输出
timerId	定时器 ID。	输入
speed	定时器 Speed (ms)。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_gv_timer.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

2.1.16.2 数据类型

相关数据类型、数据结构定义如下：

TIMER_MAX_NUM：支持最多定时器数目。

TIMER_MAX_NUM

【说明】

支持最多定时器数目。

【定义】

```
#define TIMER_MAX_NUM 0x40
```

【成员】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

2.2 控件模块

2.2.1 按钮控件

2.2.1.1 接口描述

HiGV 按钮控件提供以下 API：

- HI_GV_Button_SetCheck：设置 button 的选中状态
- HI_GV_Button_GetCheck：获取 button 的选中状态
- HI_GV_Button_SetSwitch：设置 button 的开关状态
- HI_GV_Button_GetSwitch：获取 button 的开关状态
- HI_GV_Button_SetToggleColor：设置 toggle 颜色
- HI_GV_Button_SetSwitchTextByID：设置 switch/toggle 风格按钮的显示多语言文本 ID
- HI_GV_Button_SetSwitchLayout：设置 switch 风格按钮的文本布局

HI_GV_Button_SetCheck

【描述】

设置 Button 的选中状态，Radio Button 调用该接口后不可以再调用接口 HI_GV_Widget_Enable 去使能该 Button，去使能的 Button 无法改变状态，会影响其他 Radio Button 无法切换状态。

【语法】

```
HI_S32 HI_GV_Button_SetCheck(HIGV_HANDLE buttonHandle, HIGV_BUTTON_STATUS_E status);
```

【参数】

参数名称	描述	输入/输出
buttonHandle	Button 按钮句柄。	输入
status	Button 选中状态。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_gv_conf.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Button_GetCheck

【描述】

获取 Button 的选中状态。

【语法】

```
HI_S32 HI_GV_Button_GetCheck(HIGV_HANDLE buttonHandle, HIGV_BUTTON_STATUS_E *status);
```

【参数】

参数名称	描述	输入/输出
buttonHandle	Button 按钮句柄。	输入

参数名称	描述	输入/输出
status	Button 选中状态。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_gv_conf.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Button_SetSwitch

【描述】

设置 Button 的开关状态。

【语法】

```
HI_S32 HI_GV_Button_SetSwitch(HIGV_HANDLE buttonHandle, HIGV_BUTTON_SWITCH_E switchStatus);
```

【参数】

参数名称	描述	输入/输出
buttonHandle	Button 按钮句柄。	输入
switchStatus	Button 开关状态。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_gv_conf.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Button_GetSwitch

【描述】

获取 Button 的开关状态。

【语法】

```
HI_S32 HI_GV_Button_SetSwitch(HIGV_HANDLE buttonHandle, HIGV_BUTTON_SWITCH_E*
switchStatus);
```

【参数】

参数名称	描述	输入/输出
buttonHandle	Button 按钮句柄。	输入
switchStatus	Button 开关状态指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_gv_conf.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Button_SetToggleColor

【描述】

设置 toggle 颜色。

【语法】

```
HI_S32 HI_GV_Button_SetToggleColor(HIGV_HANDLE buttonHandle, HI_COLOR toggleOnColor, HI_COLOR toggleOffColor);
```

【参数】

参数名称	描述	输入/输出
buttonHandle	Button 按钮句柄。	输入
toggleOnColor	打开状态 toggle 颜色值。	输入
toggleOffColor	关闭状态 toggle 颜色值。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_gv_conf.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Button_SetSwitchTextByID

【描述】

设置 switch/toggle 风格按钮的显示多语言文本 ID

【语法】

```
HI_S32 HI_GV_Button_SetSwitchTextByID(HIGV_HANDLE buttonHandle, const HI_U32 onStrID,const HI_U32 offStrID);
```

【参数】

参数名称	描述	输入/输出
buttonHandle	Button 按钮句柄。	输入
onStrID	打开状态多语言字串 ID。	输入
offStrID	关闭状态多语言字串 ID。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_gv_conf.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Button_SetSwitchLayout

【描述】

设置 switch 风格按钮的文本布局。

【语法】

```
HI_S32 HI_GV_Button_SetSwitchLayout(HIGV_HANDLE buttonHandle,
HIGV_BUTTON_SWITCHLAYOUT_E textLayout)
```

【参数】

参数名称	描述	输入/输出
buttonHandle	Button 按钮句柄	输入
textLayout	文本布局方向	输入

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_resm.h、hi_gv_conf.h、hi_gv_button.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.2.1.2 数据类型

相关数据类型、数据结构定义如下：

- HIGV_BUTTON_STATUS_E：按钮的状态。
- HIGV_BUTTON_SWITCH_E：按钮的开关状态。
- HIGV_BUTTON_SWITCHLAYOUT_E：switch 风格文字布局。
- HIGV_HANDLE：句柄类型。

HIGV_BUTTON_STATUS_E

【说明】

按钮的状态。

【定义】

```
typedef enum
{
    BUTTON_STATUS_UNCHECKED = 0,
    BUTTON_STATUS_CHECKED,
    BUTTON_STATUS_BUTT
} HIGV_BUTTON_STATUS_E;
```

【成员】

成员名称	描述
BUTTON_STATUS_UNCHECKED	未选中状态。
BUTTON_STATUS_CHECKED	选中状态。
BUTTON_STATUS_BUTT	非法值。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_BUTTON_SWITCH_E

【说明】

按钮的开关状态。

【定义】

```
typedef enum
{
    BUTTON_SWITCH_OFF = 0,
    BUTTON_SWITCH_ON,
    BUTTON_SWITCH_BUTT
} HIGV_BUTTON_SWITCH_E;
```

【成员】

成员名称	描述
BUTTON_SWITCH_OFF	关闭开关。
BUTTON_SWITCH_ON	打开开关。
BUTTON_SWITCH_BUTT	非法值。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_BUTTON_SWITCHLAYOUT_E

【说明】

switch 风格文字布局。

【定义】

```
typedef enum
{
    BUTTON_SWITCHLAYOUT_OFF_ON = 0,
    BUTTON_SWITCHLAYOUT_ON_OFF,
    BUTTON_SWITCHLAYOUT_BUTT
} HIGV_BUTTON_STATUS_E;
```

【成员】

成员名称	描述
BUTTON_SWITCHLAYOUT_OFF_ON	off 在左边，on 在右边的文本布局。
BUTTON_SWITCHLAYOUT_ON_OFF	on 在左边，off 在右边的文本布局。
BUTTON_SWITCHLAYOUT_BUTT	非法值。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_HANDLE

【说明】

定义句柄类型。

【定义】

```
typedef unsigned int HIGV_HANDLE;
```

【成员】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

2.2.2 Win 窗口

2.2.2.1 接口描述

HiGV win 窗口提供 API：

- HI_GV_Win_SetOpacity：设置窗口可见度。
- HI_GV_Win_GetHilightChild：获取窗口第一个高亮的子句柄。
- HI_GV_Win_ResetHilightChild：复位窗口中的高亮子控件，将窗口第一个子控件高亮。
- HI_GV_Win_SyncShow：同步显示 window 并获取结束显示的控件句柄。
- HI_GV_Win_EndSyncShow：结束窗口同步显示。
- HI_GV_Win_GetWinLevel：获取窗口等级。
- HI_GV_Win_LoadSkin：加载窗口及其子控件的皮肤。

HI_GV_Win_SetOpacity

【描述】

设置窗口可见度。

【语法】

```
HI_S32 HI_GV_Win_SetOpacity(HIGV_HANDLE windowHandle, HI_U8 opacity)
```

【参数】

参数名称	描述	输入/输出
windowHandle	Window 窗口句柄。	输入
opacity	可见度值。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_win.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Win_GetHilightChild

【描述】

获取窗口第一个高亮子句柄。

【语法】

```
HI_S32 HI_GV_Win_GetHilightChild(HIGV_HANDLE windowHandle, HIGV_HANDLE*
handleHilghtChild);
```

【参数】

参数名称	描述	输入/输出
windowHandle	Window 窗口句柄。	输入

参数名称	描述	输入/输出
handleHilghtChild	高亮子控件。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_win.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Win_ResetHilightChild

【描述】

复位窗口中的高亮子控件，将窗口第一个子控件高亮。

【语法】

```
HI_S32 HI_GV_Win_ResetHilightChild(HIGV_HANDLE windowHandle);
```

【参数】

参数名称	描述	输入/输出
windowHandle	Window 窗口句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_win.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Win_SyncShow

【描述】

同步显示 window 并获取结束显示的控件句柄。

【语法】

```
HI_S32 HI_GV_Win_SyncShow(HIGV_HANDLE windowHandle);
```

【参数】

参数名称	描述	输入/输出
windowHandle	Window 窗口句柄。	输入

【返回值】

返回值	描述
有效的 hanlde	成功。
INVALID_HANDLE	失败。

【需求】

- 头文件：hi_gv_win.h
- 库文件：libhigv.a、libhigv.so

【注意】

死循环，需要使用 HI_GV_Win_EndSyncShow 接口来退出循环。

【举例】

无。

HI_GV_Win_EndSyncShow

【描述】

结束窗口同步显示。

【语法】

```
HI_S32 HI_GV_Win_EndSyncShow(HIGV_HANDLE windowHandle, HIGV_HANDLE handleWidget);
```

【参数】

参数名称	描述	输入/输出
windowHandle	Window 窗口句柄。	输入
handleWidget	结束同步显示时返回的控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_win.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Win_GetWinLevel

【描述】

获取窗口等级

【语法】

```
HI_S32 HI_GV_Win_GetWinLevel(HIGV_HANDLE windowHandle, HI_U32* level);
```

【参数】

参数名称	描述	输入/输出
windowHandle	Window 窗口句柄。	输入
level	窗口等级。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_win.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Win_LoadSkin

【描述】

加载皮肤。

【语法】

```
HI_S32 HI_GV_Win_LoadSkin(HIGV_HANDLE windowHandle);
```

【参数】

参数名称	描述	输入/输出
windowHandle	Window 窗口句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_win.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.2.2.2 数据类型

无

2.2.3 ScrollGrid 控件

2.2.3.1 接口描述

HiGV ScrollGrid 控件提供 API：

- HI_GV_ScrollGrid_Init：控件初始化。
- HI_GV_ScrollGrid_GetCellValue：获取控件数据格的内容。
- HI_GV_ScrollGrid_SetSelCell：设置控件当前单元。
- HI_GV_ScrollGrid_GetSelCell：获取控件当前单元。
- HI_GV_ScrollGrid_GetCellNum：获取控件当前数据总单元数。
- HI_GV_ScrollGrid_PageForward：向下或向右翻页。
- HI_GV_ScrollGrid_PageBackward：向上或向左翻页。
- HI_GV_ScrollGrid_MoveToOrigin：设置焦点移动到初始位置。
- HI_GV_ScrollGrid_MoveToLast：设置焦点移动到终点位置。
- HI_GV_ScrollGrid_SetGridding：设置网格线属性。
- HI_GV_ScrollGrid_GetGridding：获取网格线属性。
- HI_GV_ScrollGrid_GetTouchDiff：获取网格相对偏移距离。
- HI_GV_ScrollGrid_Adjust：设置控件内容偏移。
- HI_GV_ScrollGrid_EnableGesture：设置控件是否支持触摸。
- HI_GV_ScrollGrid_IsGestureEnable：获取控件是否支持触摸。
- HI_GV_ScrollGrid_RegisterWidget：注册控件。

HI_GV_ScrollGrid_Init

【描述】

控件初始化。

【语法】

```
HI_S32 HI_GV_ScrollGrid_Init(HIGV_HANDLE scrollGridPara, const HIGV_SCROLLGRID_STYLE_S* style, const HIGV_SCROLLGRID_COLATTR_S* colAttr, HI_U32 cellColNum);
```

【参数】

参数名称	描述	输入/输出
scrollGridPara	ScrollGrid 控件句柄。	输入
style	控件风格布局属性。	输入
colAttr	列属性数组。	输入
cellColNum	列属性数组包含元素个数。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollGrid_GetCellValue

【描述】

获取控件数据格的内容。

【语法】

```
HI_S32 HI_GV_ScrollGrid_GetCellValue(HIGV_HANDLE scrollGridPara, HI_U32 cell，HI_U32 cellCol, HI_CHAR* value, HI_U32 length);
```

【参数】

参数名称	描述	输入/输出
scrollGridPara	ScrollGrid 控件句柄。	输入
cell	单元格。	输入
cellCol	单元格的属性列。	输入
value	单元格属性列的内容。	输出
length	单元格属性列内容的长度，单位：像素。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

需要绑定数据模型使用。

【举例】

无

HI_GV_ScrollGrid_SetSelCell

【描述】

设置控件当前单元。

【语法】

```
HI_S32 HI_GV_ScrollGrid_SetSelCell(HIGV_HANDLE scrollGridPara, HI_U32 cell);
```

【参数】

参数名称	描述	输入/输出
scrollGridPara	ScrollGrid 控件句柄。	输入
cell	条目编号。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

- 此接口用于焦点单元格的跳转，即从一个单元格跳转到另外一个单元格，在跳转时，如果控件是半行显示的状态，跳转时会有滚动的效果，否则是直接跳转到另外一个单元格的效果。
- 如果期望控件在所有状态下都可以直接跳转，需要在设置此接口前先调用一次 HI_GV_ScrollGrid_MoveToOrigin 接口。
- 需要绑定数据模型使用。

【举例】

无。

HI_GV_ScrollGrid_GetSelCell

【描述】

获取控件当前单元。

【语法】

```
HI_S32 HI_GV_ScrollGrid_GetSelCell(HIGV_HANDLE scrollGridPara, HI_U32* cell);
```


【参数】

参数名称	描述	输入/输出
scrollGridPara	ScrollGrid 控件句柄。	输入
cell	条目编号。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

需要绑定数据模型使用。

【举例】

无。

HI_GV_ScrollGrid_GetCellNum

【描述】

获取控件当前数据总单元数。

【语法】

```
HI_S32 HI_GV_ScrollGrid_GetCellNum(HIGV_HANDLE scrollGridPara, HI_U32* cellNum);
```

【参数】

参数名称	描述	输入/输出
scrollGridPara	ScrollGrid 控件句柄。	输入

参数名称	描述	输入/输出
cellNum	总条目数。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

需要绑定数据模型使用。

【举例】

无。

HI_GV_ScrollGrid_PageForward

【描述】

向下或向右翻页。

【语法】

```
HI_S32 HI_GV_ScrollGrid_PageForward(HIGV_HANDLE scrollGridPara);
```

【参数】

参数名称	描述	输入/输出
scrollGridPara	ScrollGrid 控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollGrid_PageBackward

【描述】

向上或向左翻页

【语法】

```
HI_S32 HI_GV_ScrollGrid_PageBackward(HIGV_HANDLE scrollGridPara);
```

【参数】

参数名称	描述	输入/输出
scrollGridPara	ScrollGrid 控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollGrid_MoveToOrigin

【描述】

设置焦点移动到初始位置。

【语法】

```
HI_S32 HI_GV_ScrollGrid_MoveToOrigin(HIGV_HANDLE scrollGridPara);
```

【参数】

参数名称	描述	输入/输出
scrollGridPara	ScrollGrid 控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollGrid_MoveToLast

【描述】

设置焦点移动到终点位置。

【语法】

```
HI_S32 HI_GV_ScrollGrid_MoveToLast(HIGV_HANDLE scrollGridPara);
```

【参数】

参数名称	描述	输入/输出
scrollGridPara	ScrollGrid 控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollGrid_SetGridding

【描述】

设置网格线属性。

【语法】

```
HI_S32 HI_GV_ScrollGrid_SetGridding(HIGV_HANDLE scrollGridPara, HI_U32 horizontalLineHeight, HI_COLOR horizontalLineColor, HI_U32 verticalLineWidth, HI_COLOR verticalLineColor);
```

【参数】

参数名称	描述	输入/输出
scrollGridPara	ScrollGrid 控件句柄	输入
horizontalLineHeight	水平线高度	输入
horizontalLineColor	水平线颜色	输入
verticalLineWidth	垂直线宽度	输入
verticalLineColor	垂直线颜色	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollGrid_GetGridding

【描述】

获取网格线属性。

【语法】

```
HI_S32 HI_GV_ScrollGrid_GetGridding(HIGV_HANDLE scrollGridPara, HI_U32* horizontalLineHeight, HI_COLOR* horizontalLineColor, HI_U32* verticalLineWidth, HI_COLOR* verticalLineColor);
```

【参数】

参数名称	描述	输入/输出
scrollGridPara	ScrollGrid 控件句柄	输入
horizontalLineHeight	水平线高度指针	输出
horizontalLineColor	水平线颜色指针	输出
verticalLineWidth	垂直线宽度指针	输出
verticalLineColor	垂直线颜色指针	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollGrid_GetTouchDiff

【描述】

获取网格相对偏移距离。

【语法】

```
HI_S32 HI_GV_ScrollGrid_GetTouchDiff(HIGV_HANDLE scrollGridPara, HI_S32* diff);
```

【参数】

参数名称	描述	输入/输出
scrollGridPara	ScrollGrid 控件句柄。	输入
diff	偏移距离。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

网格相对偏移距离，当前网格相对与对齐位置偏移距离。控件内容向下或右偏移对齐为负，向上或左偏移对齐为正。

【举例】

无。

HI_GV_ScrollGrid_Adjust

【描述】

设置控件内容偏移。

【语法】

```
HI_S32 HI_GV_ScrollGrid_Adjust(HIGV_HANDLE scrollGridPara, HI_S32 diff);
```

【参数】

参数名称	描述	输入/输出
scrollGridPara	ScrollGrid 控件句柄。	输入
diff	背景偏移距离。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

设置偏移量是需要偏移的距离。

【举例】

无。

HI_GV_ScrollGrid_EnableGesture

【描述】

设置控件是否支持触摸。

【语法】

```
HI_S32 HI_GV_ScrollGrid_EnableGesture(HIGV_HANDLE scrollGridPara, HI_BOOL enable);
```

【参数】

参数名称	描述	输入/输出
scrollGridPara	ScrollGrid 控件句柄。	输入
enable	是否支持触摸(HI_TRUE 表示支持，HI_FALSE 表示禁止，系统默认为 HI_TRUE)。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollGrid_IsGestureEnable

【描述】

获取控件是否支持触摸。

【语法】

```
HI_S32 HI_GV_ScrollGrid_IsGestureEnable(HIGV_HANDLE scrollGridPara, HI_BOOL* enable);
```


【参数】

参数名称	描述	输入/输出
scrollGridPara	ScrollGrid 控件句柄。	输入
enable	是否支持触摸。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollGrid_RegisterWidget

【描述】

注册控件。

【语法】

```
HI_S32 HI_GV_ScrollGrid_RegisterWidget(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollgrid.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.2.3.2 数据类型

相关数据类型、数据结构定义如下：

- HIGV_SCROLLGRID_CONV_CB：字符串转换回调函数，将 DB 中保存的字符串转换为显示需要的字符串
- HIGV_SCROLLGRID_COLTYPE_E：scrollgrid 类型，包括文字或图片
- HIGV_SCROLLGRID_COLATTR_S：scrollgrid 控件列属性结构
- HIGV_SCROLLGRID_DIRECTION_E：scrollgrid 控件的滚动方向
- HIGV_SCROLLGRID_STYLE_S：scrollgrid 控件风格布局属性

HIGV_SCROLLGRID_CONV_CB

【说明】

字符串转换回调函数，将 DB 中保存的字符串转换为显示需要的字符串。

【定义】

```
Typedef HI_S32 (*HIGV_SCROLLGRID_CONV_CB)(HIGV_HANDLE scrollGridPara, HI_U32 cellCol, HI_U32 cell, const HI_CHAR*srcStr, HI_CHAR* dstStr, HI_U32 length);
```

【成员】

成员名称	描述
scrollGridPara	控件句柄。
cellCol	单元格列。
cell	单元格。
srcStr	源数据。
dstStr	目标数据。
length	目标数据长度，单位：字节。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_SCROLLGRID_COLTYPE_E

【说明】

scrollgrid 控件类型枚举。

【定义】

```
typedef enum
{
    SCROLLGRID_COLTYPE_TXT = 0,
    SCROLLGRID_COLTYPE_IMG,
    SCROLLGRID_COLTYPE_IMG_MEM,
    SCROLLGRID_COLTYPE_BUTT
} HIGV_SCROLLGRID_COLTYPE_E;
```

【成员】

成员名称	描述
SCROLLGRID_COLTYPE_TXT	文本类型。

成员名称	描述
SCROLLGRID_COLTYPE_IMG	图片类型。
SCROLLGRID_COLTYPE_IMG_MEM	图片类型.
SCROLLGRID_COLTYPE_BUTT	非法值。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_SCROLLGRID_COLATTR_S

【说明】

scrollgrid 控件列属性结构。

【定义】

```
typedef struct _HIGV_SCROLLGRID_COLATTR_S
{
    HIGV_SCROLLGRID_COLTYPE_E Type;
    HI_U32                    Top;
    HI_U32                    Left;
    HI_U32                    Width;
    HI_U32                    Height;
    HI_U32                    Align;
    HIGV_HANDLE                   hImage;
    HI_S32                    FieldColIdxInDb;
    HIGV_SCROLLGRID_CONV_CB   ConvString;
} HIGV_SCROLLGRID_COLATTR_S;
```

【成员】

成员名称	描述
Type	单元格的控件类型。

成员名称	描述
Top	相对 cell 的最上坐标。
Left	相对 cell 的最左坐标。
Width	Cell 的列宽，单位：像素。
Height	Cell 的列高，单位：像素。
Align	文本对齐方式。
hImage	需要显示图片的 surface 句柄。
FieldColIdxInDb	该列在数据库中的序号。
ConvString	字符串转换回调函数。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_SCROLLGRID_DIRECTION_E

【说明】

Scrollgrid 控件的滚动方向。

【定义】

```
typedef enum _SCROLLGRID_DIRECTION_E
{
    SCROLLGRID_DIRECTION_HORI,
    SCROLLGRID_DIRECTION_VERT,
    SCROLLGRID_DIRECTION _BUTT
} HIGV_SCROLLGRID_DIRECTION_E;
```

【成员】

成员名称	描述
SCROLLGRID_DIRECTION_HORI	水平方向延伸。
SCROLLGRID_DIRECTION _VERT	垂直方向。
SCROLLGRID_DIRECTION _BUTT	非法值。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_SCROLLGRID_STYLE_S

【说明】

scrollgrid 控件列属性结构。

【定义】

```
typedef struct _HIGV_SCROLLGRID_STYLE_S
{
    HI_U32                  RowNum;
    HI_U32                  ColNum;
    HI_U32                  LeftMargin;
    HI_U32                  RightMargin;
    HI_U32                  TopMargin;
    HI_U32                  BtmMargin;
    HI_U32                  RowSpace;
    HI_S32                  ColSpace;
    HI_U32                  HLHeight;
    HI_COLOR                HLColor;
    HI_U32                  VLWidth;
    HI_COLOR                VLColor;
    HI_U32                  FocusRectAnchor;
    HI_U32                  IsFocusAtBg;
   HI_U32                   ImgDecIndex;
   HI_U32                   IsDynamicDecImg;
```



```
    HIGV_SCROLLGRID_DIRECTION_E   Direction;
    HI_RESID                      FocusRectSkin;
    HI_RESID                      FocusRectNormalSkin;
    HI_RESID                      replaceImage;
} HIGV_SCROLLGRID_STYLE_S;
```

【成员】

成员名称	描述
RowNum	页显示行数。
ColNum	页显示列数。
LeftMargin	页左边距。
RightMargin	页右边距。
TopMargin	页顶边距。
BtmMargin	页底边距。
RowSpace	页行间距。
ColSpace	页列间距。
HLHeight	水平网格线宽度。
HLColor	水平网格线颜色。
VLWidth	垂直网格线宽度。
VLColor	垂直网格线颜色。
FocusRectAnchor	触发条目滚动时焦点框锚位置。
IsFocusAtBg	焦点框绘制在背景之上。
ImgDecIndex	图片解码索引。
IsDynamicDecImg	标记是否动态解码。
Direction	延伸方向。
FocusRectSkin	焦点框皮肤。
FocusRectNormalSkin	常态皮肤。

成员名称	描述
replaceImage	快滑时背景替换图片

【注意事项】

无。

【相关数据类型及接口】

无。

2.2.4 ScrollBar 控件

scrollbar 内部没有实现重绘动作，重绘时，需要重绘绑定 scrollbar 的控件。如调用 HI_GV_Widget_Paint 接口时，传入绑定 scrollbar 的控件句柄。

2.2.4.1 接口描述

HiGV ScrollBar 控件提供 API：

- HI_GV_ScrollBar_SetSlideRes：设置滑块信息。
- HI_GV_ScrollBar_SetPos：设置滑块位置。
- HI_GV_ScrollBar_GetPos：获取滑块位置。
- HI_GV_ScrollBar_SetScrollRange：设置滚动条滚动范围。
- HI_GV_ScrollBar_SetVisibleLen：设置可见区域长度。
- HI_GV_ScrollBar_SetContentLen：设置内容长度。
- HI_GV_ScrollBar_SetButtonImg：设置滚动条上下按钮正常和按下状态图片。
- HI_GV_ScrollBar_SetStatus：设置 ScrollBar 控件的绑定状态。

HI_GV_ScrollBar_SetSlideRes

【描述】

设置滑块信息。

【语法】

```
HI_S32 HI_GV_ScrollBar_SetSlideRes(HIGV_HANDLE scrollBarHandle, HI_RESID slideRes);
```

【参数】

参数名称	描述	输入/输出
scrollBarHandle	ScrollBar 控件句柄。	输入
slideRes	资源信息。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollBar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollBar_SetPos

【描述】

设置滑块位置。

【语法】

```
HI_S32 HI_GV_ScrollBar_SetPos(HIGV_HANDLE scrollBarHandle, HI_U32 pos);
```

【参数】

参数名称	描述	输入/输出
scrollBarHandle	ScrollBar 控件句柄。	输入
pos	滑块位置。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollBar.h
- 库文件：libhigv.a、libhigv.so

【注意】

pos 小于等于滚动条滚动范围。

【举例】

无。

HI_GV_ScrollBar_GetPos

【描述】

获取滑块位置。

【语法】

```
HI_S32 HI_GV_ScrollBar_GetPos(HIGV_HANDLE scrollBarHandle, HI_U32* pos);
```

【参数】

参数名称	描述	输入/输出
scrollBarHandle	ScrollBar 控件句柄。	输入
pos	滑块位置。	输出

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollBar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollBar_SetScrollRange

【描述】

设置滚动条滚动范围。

【语法】

```
HI_S32 HI_GV_ScrollBar_SetScrollRange(HIGV_HANDLE scrollBarHandle, HI_U32 scrollRange);
```

【参数】

参数名称	描述	输入/输出
scrollBarHandle	ScrollBar 控件句柄。	输入
scrollRange	滚动条滚动范围。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollBar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollBar_SetVisibleLen

【描述】

设置可见区域长度。

【语法】

```
HI_S32 HI_GV_ScrollBar_SetVisibleLen(HIGV_HANDLE scrollBarHandle, HI_U32 len);
```

【参数】

参数名称	描述	输入/输出
scrollBarHandle	ScrollBar 控件句柄。	输入
len	控件可见区域的长度，单位：像素。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollBar.h

- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollBar_SetContentLen

【描述】

设置内容长度

【语法】

```
HI_S32 HI_GV_ScrollBar_SetContentLen(HIGV_HANDLE scrollBarHandle, HI_U32 len;
```

【参数】

参数名称	描述	输入/输出
scrollBarHandle	ScrollBar 控件句柄。	输入
len	控件有效内容的长度，单位：像素。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollBar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollBar_SetButtonImg

【描述】

设置滚动条上下按钮正常和按下状态图片。

【语法】

```
HI_S32 HI_GV_ScrollBar_SetButtonImg(HIGV_HANDLE scrollBarHandle, HI_RESID upNormalImg, HI_RESID upPressImg, HI_RESID downNormalImg, HI_RESID downPressImg);
```

【参数】

参数名称	描述	输入/输出
scrollBarHandle	ScrollBar 控件句柄。	输入
upNormalImg	上按钮正常图片。	输入
upPressImg	上按钮按下图片。	输入
downNormalImg	下按钮正常图片。	输入
downPressImg	下按钮按下图片。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollBar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollBar_SetStatus

【描述】

设置 ScrollBar 控件的绑定状态。

【语法】

```
HI_S32 HI_GV_ScrollBar_SetStatus(HIGV_HANDLE scrollBarHandle, HI_BOOL isSuspending);
```

【参数】

参数名称	描述	输入/输出
scrollBarHandle	ScrollBar 控件句柄。	输入
isSuspending	滑块是否悬浮。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollBar.h
- 库文件：libhigv.a、libhigv.so

【注意】

当前 scrollbar 为悬浮状态并且绑定的控件的内容的长度小于等于控件的长度时，不绘制 scrollbar 控件。

【举例】

无。

2.2.4.2 数据类型

无

2.2.5 ScrollView 控件

ScrollView 子控件焦点切换需要用户在回调实现。

2.2.5.1 接口描述

HiGV ScrollView 控件提供 API：

- HI_GV_ScrollView_Init：初始化控件。
- HI_GV_ScrollView_SetContentSize：设置内容的固定大小。
- HI_GV_ScrollView_MoveToCenter：将视口移至中央。
- HI_GV_ScrollView_GetContentRect：获取内容的矩形。
- HI_GV_ScrollView_GetViewCoordinate：获取视口相对内容的坐标。
- HI_GV_ScrollView_SetViewCoordinate：设置视口相对内容的坐标。
- HI_GV_ScrollView_SetStep：设置滚动步长。
- HI_GV_ScrollView_SetInterval：设置滚动间隔。
- HI_GV_ScrollView_BindScrollBar：绑定滚动条。
- HI_GV_ScrollView_CheckFocusPos：检查确保焦点控件在视口内。
- HI_GV_ScrollView_SetScrollParam：设置滑动操作的灵敏度系数。
- HI_GV_ScrollView_SetFlingParam：设置轻扫操作的灵敏度系数。

HI_GV_ScrollView_Init

【描述】

初始化控件。

【语法】

```
HI_S32 HI_GV_ScrollView_Init (HIGV_HANDLE scrollViewHandle, const
HIGV_SCROLLVIEW_Init_S* initAttr);
```

【参数】

参数名称	描述	输入/输出
scrollViewHandle	ScrollView 控件句柄。	输入
initAttr	控件创建信息。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollView.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollView_SetContentSize

【描述】

设置内容固定大小。

【语法】

```
HI_S32 HI_GV_ScrollView_SetContentSize（HIGV_HANDLE scrollViewHandle, HI_U32 width,
HI_U32 height);
```

【参数】

参数名称	描述	输入/输出
scrollViewHandle	ScrollView 控件句柄。	输入

参数名称	描述	输入/输出
width	宽度，单位：像素。	输入
height	高度，单位：像素。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollView.h
- 库文件：libhigv.a、libhigv.so

【注意】

- Width 的取值范围需大于或等于 Scrollview 控件的宽度。
- Height 的取值范围需大于或等于 Scrollview 控件的高度。

【举例】

无。

HI_GV_ScrollView_MoveToCenter

【描述】

将视口移至中央。

【语法】

```
HI_S32 HI_GV_ScrollView_MoveToCenter（HIGV_HANDLE scrollViewHandle);
```

【参数】

参数名称	描述	输入/输出
scrollViewHandle	ScrollView 控件句柄。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollView.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollView_GetContentRect

【描述】

获取内容矩形。

【语法】

```
HI_S32 HI_GV_ScrollView_GetContentRect (HIGV_HANDLE scrollViewHandle, HI_RECT* rect);
```

【参数】

参数名称	描述	输入/输出
scrollViewHandle	ScrollView 控件句柄。	输入
rect	内容矩形。	输出

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollView.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollView_GetViewCoordinate

【描述】

获取视口相对内容的坐标。

【语法】

```
HI_S32 HI_GV_ScrollView_GetViewCoordinate（HIGV_HANDLE scrollViewHandle, HIGV_CORD* cordX, HIGV_CORD* cordY);
```

【参数】

参数名称	描述	输入/输出
scrollViewHandle	ScrollView 控件句柄。	输入
cordX	X 坐标。	输出
cordY	Y 坐标。	输出

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollView.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollView_SetViewCoordinate

【描述】

设置视口相对内容的坐标。

【语法】

```
HI_S32 HI_GV_ScrollView_SetViewCoordinate（HIGV_HANDLE scrollViewHandle, HIGV_CORD cordX， HIGV_CORD cordY);
```

【参数】

参数名称	描述	输入/输出
scrollViewHandle	ScrollView 控件句柄。	输入
cordX	X 坐标。	输入
cordY	Y 坐标。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollView.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollView_SetStep

【描述】

设置滚动步长。

【语法】

```
HI_S32 HI_GV_ScrollView_SetStep（HIGV_HANDLE scrollViewHandle, HI_U32 step）；
```

【参数】

参数名称	描述	输入/输出
scrollViewHandle	ScrollView 控件句柄。	输入
step	滚动步长，单位为 pixel。	输入

【返回值】

返回值	描述
0	成功

返回值	描述
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollView.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollView_SetInterval

【描述】

设置滚动间隔。

【语法】

```
HI_S32 HI_GV_ScrollView_SetInterval（HIGV_HANDLE scrollViewHandle, HI_U32 interval);
```

【参数】

参数名称	描述	输入/输出
scrollViewHandle	ScrollView 控件句柄。	输入
interval	间隔。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollView.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollView_BindScrollBar

【描述】

绑定滚动条。

【语法】

```
HI_S32 HI_GV_ScrollView_BindScrollBar（HIGV_HANDLE scrollViewHandle, HIGV_HANDLE handleVertical, HIGV_HANDLE handleHorizontal);
```

【参数】

参数名称	描述	输入/输出
scrollViewHandle	ScrollView 控件句柄。	输入
handleVertical	垂直滚动条。	输入
handleHorizontal	水平滚动条。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollView.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollView_CheckFocusPos

【描述】

检查确保焦点控件在视口内。

【语法】

```
HI_S32 HI_GV_ScrollView_CheckFocusPos（HIGV_HANDLE scrollViewHandle);
```

【参数】

参数名称	描述	输入/输出
scrollViewHandle	ScrollView 控件句柄。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollView.h
- 库文件：libhigv.a、libhigv.so

【注意】

该接口是把焦点子控件移至视口内，请保证在 scrollview 为焦点控件的情况下使用此接口。

【举例】

无

HI_GV_ScrollView_SetScrollParam

【描述】

设置滑动操作的灵敏度系数，默认是 1.0。

【语法】

```
HI_S32 HI_GV_ScrollView_SetScrollParam(HIGV_HANDLE scrollViewHandle, HI_FLOAT scrollParam);
```

【参数】

参数名称	描述	输入/输出
scrollViewHandle	ScrollView 控件句柄。	输入
scrollParam	滑动灵敏度系数。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollView.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollView_SetFlingParam

【描述】

设置轻扫操作的灵敏度系数，默认是 1.0。

【语法】

```
HI_S32 HI_GV_ScrollView_SetFlingParam(HIGV_HANDLE scrollViewHandle, HI_FLOAT flingParam);
```

【参数】

参数名称	描述	输入/输出
scrollViewHandle	ScrollView 控件句柄。	输入
flingParam	轻扫灵敏度系数。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_scrollView.h
- 库文件：libhigv.a、libhigv.so

【注意】

灵敏度系数控制的是滑动的距离。

【举例】

无。

2.2.5.2 数据类型

相关数据类型、数据结构定义如下：

- HIGV_SCROLLVIEW_DIRECTION_E：scrollview 控件的滚动方向。
- HIGV_SCROLLVIEW_Init_S：scrollview 控件初始化信息。
- HigvScrollViewCord：视口相对内容偏移坐标。

HIGV_SCROLLVIEW_DIRECTION_E

【说明】

Scrollview 控件的滚动方向。

【定义】

```
typedef enum
{
    SCROLLVIEW_DIRECTION_HORI = 0,
    SCROLLVIEW_DIRECTION_VERT,
    SCROLLVIEW_DIRECTION_BUTT
} HIGV_SCROLLVIEW_DIRECTION_E;
```

【成员】

成员名称	描述
SCROLLVIEW_DIRECTION_HORI	水平方向延伸。
SCROLLVIEW_DIRECTION_VERT	垂直方向延伸。
SCROLLVIEW_DIRECTION_BUTT	非法值。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_SCROLLVIEW_Init_S

【说明】

Scrollview 控件初始化信息。

【定义】

```
typedef struct hiHIGV_SCROLLVIEW_Init_S
{
    HIGV_HANDLE                 hVerticalScrollBar;
    HIGV_HANDLE                 hHorizontalScrollBar;
    HI_U32                      LeftMargin;
    HI_U32                      RightMargin;
    HI_U32                      TopMargin;
    HI_U32                      BottomMargin;
   HI_U32                        Step;
   HI_U32                        Interval;
   HI_U32                        ScrollContentWidth;
   HI_U32                        ScrollContentHeight;
   HI_U32                        ScrollViewStyle;
   HIGV_SCROLLVIEW_DIRECTION_E Direction;
} HIGV_SCROLLVIEW_Init_S;
```

【成员】

成员名称	描述
hVerticalScrollBar	绑定的垂直滚动条句柄。
hHorizontalScrollBar	绑定的水平滚动条句柄。
LeftMargin	子控件在背景的左边距。
RightMargin	子控件在背景的右边距。
TopMargin	子控件在背景的顶边距。
BottomMargin	子控件在背景的底边距。
Step	滚动步长。
Interval	滚动间隔。

成员名称	描述
ScrollContentWidth	滚动内容的固定宽度，单位：像素。 为 0 时根据子控件布局自动调整，有效值须大于或等于控件宽度。 注意：设置为 0 时，高度也会自动调整。
ScrollContentHeight	滚动内容的固定高度，单位：像素。为 0 时根据子控件布局自动调整，有效值须大于或等于控件高度。 注意：设置为 0 时，宽度也会自动调整。
ScrollViewStyle	私有风格。
Direction	滚动方向。

【注意事项】

无。

【相关数据类型及接口】

无。

HigvScrollViewCord

【说明】

Scrollview 控件视口相对内容偏移坐标。

【定义】

```
typedef struct {
    HI_S32 cordX;
    HI_S32 cordY;
} HigvScrollViewCord;
```

【成员】

成员名称	描述
cordX	视口相对内容偏移 x 坐标。

成员名称	描述
cordY	视口相对内容偏移 y 坐标。

【注意事项】

无。

【相关数据类型及接口】

无。

2.2.6 ListBox 控件

2.2.6.1 接口描述

HiGV ListBox 控件提供 API：

- HI_GV_List_Init：控件初始化。
- HI_GV_List_InitEx：控件扩展初始化。
- HI_GV_List_GetCell：获取 ListBox 控件数据格的内容。
- HI_GV_List_SetRowSkin：设置焦点行的皮肤。
- HI_GV_List_SetSelItem：设置 listbox 当前条目。
- HI_GV_List_GetSelItem：获取 listbox 当前条目。
- HI_GV_List_GetItemNum：获取 listbox 当前数据总条数。
- HI_GV_List_IsListBoxType：是否为 listbox 控件类型。
- HI_GV_List_SetGridding：设置网络属性。
- HI_GV_List_GetGridding：获取网络属性。
- HI_GV_List_ChangeImage：改变显示的图片。
- HI_GV_List_GetStartItem：获取当前显示页面首项的序号。
- HI_GV_List_SetStartItem：设置当前显示页面首项的序号。
- HI_GV_List_GetEndItem：获取当前显示页面末项的序号。
- HI_GV_List_SetSelCell：设置 listbox 当前焦点单元格。
- HI_GV_List_GetSelCell：获取 listbox 当前焦点单元格。
- HI_GV_List_GetCellWidth：单元格模式下获取单元格宽度。

- HI_GV_List_SetStep：设置文本的滚动步长。
- HI_GV_List_SetDirection：设置滚动方向。
- HI_GV_List_SetScroll：设置控件状态。
- HI_GV_List_SetColFgIdx：设置列前景颜色。
- HI_GV_List_SetNoFrame：设置是否绘制网格外框。
- HI_GV_List_SetCyc：设置是否循环焦点。
- HI_GV_List_SetScrollParam：设置滑动灵敏度系数。
- HI_GV_List_SetFlingParam：设置轻扫灵敏度系数。

HI_GV_List_Init

【描述】

控件初始化。

【语法】

```
HI_S32 HI_GV_List_Init（HIGV_HANDLE listHandle，HI_U32 rowNum, HI_U32 colNum， const HIGV_LIST_COLATTR_S* colAttr);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
rowNum	显示行数。	输入
colNum	列数。	输入
colAttr	列属性数组。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_InitEx

【描述】

控件扩展初始化

【语法】

```
HI_S32 HI_GV_List_InitEx（HIGV_HANDLE listHandle, const HIGV_LIST_ATTRIBUTE_S* listAttr);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
listAttr	初始化列表属性框。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_GetCell

【描述】

获取 listbox 控件数据格的内容。

【语法】

```
HI_S32 HI_GV_List_GetCell (HIGV_HANDLE listHandle,  HI_U32 item, HI_U32 col, HI_CHAR* value, HI_U32 length);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
item	行。	输入
col	列。	输入
value	单元格内容。	输出
length	单元格内容 pValue 字符串存储长度。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h

- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_SetRowSkin

【描述】

设置焦点行的皮肤。

【语法】

```
HI_S32 HI_GV_List_SetRowSkin (HIGV_HANDLE listHandle, HIGV_HANDLE selSkin,
HIGV_HANDLE normSkin);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
selSkin	控件处于焦点模式下选中行皮肤。	输入
normSkin	控件处于非焦点模式下选中行皮肤。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_SetSelItem

【描述】

设置 listbox 当前条目。

【语法】

```
HI_S32 HI_GV_List_SetSelItem (HIGV_HANDLE listHandle, HI_U32 item);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
item	条目编号。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_GetSelItem

【描述】

获取 listbox 当前条目。

【语法】

```
HI_S32 HI_GV_List_GetSelItem (HIGV_HANDLE listHandle, HI_U32* item);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
item	条目编号。	输出

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_GetItemNum

【描述】

获取 listbox 当前数据总条数。

【语法】

```
HI_S32 HI_GV_List_GetItemNum (HIGV_HANDLE listHandle,  HI_U32* itemNum);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
itemNum	数据总条目数。	输出

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_IsListBoxType

【描述】

是否为 listbox 控件类型。

【语法】

```
HI_BOOL HI_GV_List_IsListBoxType（HIGV_HANDLE listHandle）；
```


【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入

【返回值】

返回值	描述
HI_TRUE	是 ListBox 控件
HI_FALSE	不是 ListBox 控件

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_SetGridding

【描述】

设置网格属性。

【语法】

```
HI_S32 HI_GV_List_SetGridding (HIGV_HANDLE listHandle, HI_U32 horizontalLineHeight,
HI_U32 horizontalLineColor, HI_U32 verticalLineWidth, HI_U32 verticalLineColor);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入

参数名称	描述	输入/输出
horizontalLineHeight	水平线高度，单位：像素。	输入
horizontalLineColor	水平线颜色。	输入
verticalLineWidth	垂直线宽度，单位：像素。	输入
verticalLineColor	垂直线颜色。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_GetGridding

【描述】

获取网格属性。

【语法】

```
HI_S32 HI_GV_List_GetGridding (HIGV_HANDLE listHandle, HI_U32* horizontalLineHeight,
HI_COLOR* horizontalLineColor, HI_U32* verticalLineWidth, HI_COLOR* verticalLineColor);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
horizontalLineHeight	水平线高度，单位：像素。	输出
horizontalLineColor	水平线颜色，单位：像素。	输出
verticalLineWidth	垂直线宽度。	输出
verticalLineColor	垂直线颜色。	输出

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_ChangeImage

【描述】

改变显示的图片。

【语法】

```
HI_S32 HI_GV_List_ChangeImage (HIGV_HANDLE listHandle, HI_U32 col, HIGV_HANDLE imageHandle);
```


【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
col	列。	输入
imageHandle	图片句柄。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_GetStartItem

【描述】

获取当前显示页面首项的序号。

【语法】

```
HI_S32 HI_GV_List_GetStartItem (HIGV_HANDLE listHandle, HI_U32* item);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
item	行。	输出

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_SetStartItem

【描述】

设置当前显示页面首项的序号。

【语法】

```
HI_S32 HI_GV_List_SetStartItem (HIGV_HANDLE listHandle, HI_U32 item);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
item	行号。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_GetEndItem

【描述】

获取当前显示页面末项的序号。

【语法】

```
HI_S32 HI_GV_List_GetEndItem (HIGV_HANDLE listHandle, HI_U32* item);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
item	行。	输出

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

此接口需要 Listbox 控件完成绘制后才生效。

【举例】

无。

HI_GV_List_SetSelCell

【描述】

设置 listbox 当前焦点单元格。

【语法】

```
HI_S32 HI_GV_List_SetSelCell (HIGV_HANDLE listHandle,  HI_U32 item, HI_U32 col);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
item	行。	输入
col	列。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_GetSelCell

【描述】

获取 listbox 当前焦点单元格。

【语法】

```
HI_S32 HI_GV_List_GetSelCell (HIGV_HANDLE listHandle, HI_U32* item, HI_U32* col);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
item	行。	输出
col	列。	输出

【返回值】

返回值	描述
0	成功

返回值	描述
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_GetCellWidth

【描述】

单元格焦点模式下获取单元格宽度。

【语法】

```
HI_S32 HI_GV_List_GetCellWidth(HIGV_HANDLE listHandle, HI_U32 item, HI_U32 col, HI_U32* width);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
item	行。	输入
col	列。	输入
width	宽度，单位：像素。	输出

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_SetStep

【描述】

设置文本的滚动步长。

【语法】

```
HI_S32 HI_GV_List_SetStep (HIGV_HANDLE listHandle, HI_U32 step);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
step	滚动步长，单位为 pixel。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_SetDirection

【描述】

设置滚动方向。

【语法】

```
HI_S32 HI_GV_List_SetDirection (HIGV_HANDLE listHandle, HI_BOOL fromLeft);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
fromLeft	由左向右或由右向左。 HI_FALSE：由右向左； HI_TRUE：由左向右。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_SetScroll

【描述】

设置控制状态。

【语法】

```
HI_S32 HI_GV_List_SetScroll (HIGV_HANDLE listHandle, HI_BOOL scroll);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄	输入
scroll	滚动状态。 HI_TRUE：滚动； HI_FALSE：停止滚动。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_SetColFgIdx

【描述】

设置列前景颜色。

【语法】

```
HI_S32 HI_GV_List_SetColFgIdx (HIGV_HANDLE listHandle, HI_U32 col, HI_COLOR index);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
col	列。	输入
index	颜色值。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_SetNoFrame

【描述】

设置是否绘制网格外框。

【语法】

```
HI_S32 HI_GV_List_SetNoFrame（HIGV_HANDLE listHandle, HI_BOOL noFrame）；
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
noFrame	是否绘制。 HI_FALSE：绘制； HI_TRUE：不绘制。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_SetCyc

【描述】

设置是否循环焦点。

【语法】

```
HI_S32 HI_GV_List_SetCyc（HIGV_HANDLE listHandle， HI_BOOL cycle）；
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
cycle	循环。 HI_FALSE：不循环； HI_TRUE：循环。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_SetScrollParam

【描述】

设置滑动灵敏度系数(默认为 1.0，大于等于 1.0)。

【语法】

```
HI_S32 HI_GV_List_SetScrollParam(HIGV_HANDLE listHandle, HI_FLOAT scrollParam);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
scrollParam	滑动灵敏度系数。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_List_SetFlingParam

【描述】

设置轻扫灵敏度系数(默认为 2，大于等于 1)。

【语法】

```
HI_S32 HI_GV_List_SetFlingParam(HIGV_HANDLE listHandle, HI_S32 flingParam);
```

【参数】

参数名称	描述	输入/输出
listHandle	控件句柄。	输入
flingParam	轻扫灵敏度系数。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_listbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

灵敏度系数控制的是滑动的响应速度。

【举例】

无。

2.2.6.2 数据类型

相关数据类型、数据结构定义如下：

- HIGV_LIST_COLTYPE_E：listbox 控件列类型。
- HIGV_LIST_CONV_CB：字符串转换函数：将 DB 中保存的字符串转换为显示时需要的字符串。
- HIGV_GET_WIDTH_CB：单元格焦点模式下，控件用来获取单元格宽度回调函数。

- HIGV_LIST_COLATTR_S：listbox 控件列属性结构。
- HIGV_LIST_ATTRIBUTE_S：listbox 控件属性结构。

HIGV_LIST_COLTYPE_E

【说明】

Listbox 控件列类型。

【定义】

```
typedef enum
{
    LIST_COLTYPE_TXT = 0,
    LIST_COLTYPE_IMG,
    LIST_COLTYPE_TXTICONLEFT,
    LIST_COLTYPE_TXTICONRIGHT,
    LIST_COLTYPE_BUTT
 }
HIGV_LIST_COLTYPE_E;
```

【成员】

成员名称	描述
LIST_COLTYPE_TXT	文本。
LIST_COLTYPE_IMG,	图片。
LIST_COLTYPE_TXTICONLEFT	文本和图标，图标在左。
LIST_COLTYPE_TXTICONRIGHT	文本和图标，图标在右。
LIST_COLTYPE_BUTT	非法值。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_LIST_CONV_CB

【说明】

字符串转换回调函数，将 DB 中保存的字符串转换为显示时需要的字符串。

【定义】

```
typedef HI_S32 (*HIGV_LIST_CONV_CB)(HIGV_HANDLE listHandle, HI_U32 col, HI_U32 item,
const HI_CHAR* srcStr, HI_CHAR* dstStr, HI_U32 length);
```

【成员】

成员名称	描述
listHandle	控件句柄。
col	列。
item	项。
srcStr	源数据。
dstStr	目标数据。
length	目标数据长度，单位：字节。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_GET_WIDTH_CB

【说明】

单元格焦点模式下，控件用来获取单元格宽度回调函数。

【定义】

```
typedef HI_S32 (* HIGV_GET_WIDTH_CB)(HIGV_HANDLE listHandle, HI_U32 item, HI_U32 col);
```

【成员】

成员名称	描述
listHandle	控件句柄。
item	项。
col	列。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_LIST_COLATTR_S

【说明】

Listbox 控件列属性结构。

【定义】

```
typedef struct
{
    HIGV_LIST_COLTYPE_E    Type;
    HI_U32                 Width;
    HI_U32                 Align;
        HI_COLOR               Fgidx;
        HIGV_HANDLE            hImage;
        HI_S32                 FieldColIdxInDb;
        HIGV_LIST_CONV_CB      ConvString;
 }
HIGV_LIST_COLATTR_S;
```

【成员】

成员名称	描述
Type	列属性类型。
Width	列宽，单位：像素。

成员名称	描述
Align	文本对齐方式。
Fgidx	列文本字体颜色。
hImage	图片句柄。
FieldColIdxInDb	该列在数据库中的列号。
ConvString	字符串转换函数。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_LIST_ATTRIBUTE_S

【说明】

Listbox 控件属性结构。

【定义】

```
typedef struct
{
    HI_U32                  RowNum;
    HI_U32                  ColNum;
    HI_BOOL                 NoFrame;
        HI_BOOL                 Cyc;
        HI_BOOL                 IsCellActive;
        HI_BOOL                 AutoSwitchCell;
        HI_BOOL                 Scroll;
        HI_BOOL                 Fromleft;
        HI_U32                  Timeinterval;
        HI_U32                  Step;
        HI_U32                  ScrollCol;
        HIGV_GET_WIDTH_CB   GetWidthCb;
        HIGV_LIST_COLATTR_S*  pColAttr;
```



```
}
HIGV_LIST_ATTRIBUTE_S;
```

【成员】

成员名称	描述
RowNum	行数目。
ColNum	列数目。
NoFrame	是否无边框。
Cyc	是否循环滚动。
IsCellActive	当前单元格是否是焦点。
AutoSwitchCell	单元格焦点是否自动切换。
Scroll	是否滚动单元格显示。
Fromleft	是否从左边开始滚动。
Timeinterval	滚动时间间隔。 参数值为 0 时，设为默认值 300。
Step	滚动单元格的滚动步长。 参数值为 0 时，设为默认值 10。
ScrollCol	条目焦点时可滚动列。 参数 ScrollCol 的值为 HIGV_LISTBOX_COL_MAX(20)时，设置为不滚动。
GetWidthCb	控件内部回调用户设置单元格宽度函数。
pColAttr	列表框控件列属性结构数组，有 ColNum 个元素。

【注意事项】

无。

【相关数据类型及接口】

无。

2.2.7 Clock 控件

2.2.7.1 接口描述

HiGV Clock 控件提供 API：

- HI_GV_Clock_Init：初始化控件。
- HI_GV_Clock_SetFlash：设置时间控件低位的冒号是否闪烁。
- HI_GV_Clock_SetTimeUnit：设置时间控件的最小单元。
- HI_GV_Clock_Run：控制时间控件开始或结束计时。
- HI_GV_Clock_ClearContent：清除时钟控件的内容。
- HI_GV_Clock_SetUTC：设置 Clock 控件 UTC 时间。
- HI_GV_Clock_GetUTC：获取 Clock 控件 UTC 时间。
- HI_GV_Clock_SetSwitchItemStep：设置条目切换步长。

HI_GV_Clock_Init

【描述】

初始化控件。

【语法】

```
HI_S32 HI_GV_Clock_Init (HIGV_HANDLE clockHandle, const HIGV_CLOCK_STYLE_S* style);
```

【参数】

参数名称	描述	输入/输出
clockHandle	控件句柄。	输入
style	控件创建信息。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_clock.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Clock_SetFlash

【描述】

设置时间控件低位的冒号是否闪烁。

【语法】

```
HI_S32 HI_GV_Clock_SetFlash(HIGV_HANDLE clockHandle, HI_BOOL flash);
```

【参数】

参数名称	描述	输入/输出
clockHandle	控件句柄。	输入
flash	是否闪烁。 HI_TRUE：闪烁； HI_FALSE：不闪烁。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_clock.h
- 库文件：libhigv.a、libhigv.so

【注意】

目前只支持 text 模式下的冒号闪烁。

【举例】

无。

HI_GV_Clock_SetTimeUnit

【描述】

设置时间控件的最小单元。

【语法】

```
HI_S32 HI_GV_Clock_SetTimeUnit(HIGV_HANDLE clockHandle, const HI_CHAR* unit);
```

【参数】

参数名称	描述	输入/输出
clockHandle	控件句柄。	输入
unit	字符串的最小单元，有“second”， “minute” ，“hour”， “day”，“week”， “month”，“year”。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_clock.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Clock_Run

【描述】

控制时间控件开始或结束计时。

【语法】

```
HI_S32 HI_GV_Clock_Run (HIGV_HANDLE clockHandle, HI_BOOL run);
```

【参数】

参数名称	描述	输入/输出
clockHandle	控件句柄。	输入
run	时间控件运行或停止。 HI_TRUE：运行； HI_FALSE：停止。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_clock.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Clock_ClearContent

【描述】

清除时钟控件的内容。

【语法】

```
HI_S32 HI_GV_Clock_ClearContent (HIGV_HANDLE clockHandle);
```

【参数】

参数名称	描述	输入/输出
clockHandle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_clock.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Clock_SetUTC

【描述】

设置 clock 控件 UTC 时间。

【语法】

```
HI_S32 HI_GV_Clock_SetUTC（HIGV_HANDLE clockHandle, time_t t）；
```

【参数】

参数名称	描述	输入/输出
clockHandle	控件句柄。	输入
t	UTC 的时间指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_clock.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Clock_GetUTC

【描述】

获取 clock 控件 UTC 时间。

【语法】

```
HI_S32 HI_GV_Clock_GetUTC（HIGV_HANDLE clockHandle, time_t* t）；
```


【参数】

参数名称	描述	输入/输出
clockHandle	控件句柄。	输入
t	UTC 的时间指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_clock.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Clock_SetSwitchItemStep

【描述】

设置条目切换步长。

【语法】

```
HI_S32 HI_GV_Clock_SetSwitchItemStep (HIGV_HANDLE clockHandle, HI_U32 step，HI_U32 interval);
```

【参数】

参数名称	描述	输入/输出
clockHandle	控件句柄。	输入
step	切换步长。	输入
interval	时间间隔，单位：ms。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_clock.h
- 库文件：libhigv.a、libhigv.so

【注意】

- 仅对 YEAR 条目有效。
- 年份最小值为 1970.

【举例】

两次设置时间命令的间隔	设置的步长	设置的时间间隔	年份增加值
<2S	5	2S	1(默认值)
>2S	5	2S	5(设置值)

由于第一次设置命令没有时间对比，按默认值增加。

2.2.7.2 数据类型

相关数据类型、数据结构定义如下：

- HIGV_TIME_S：时间信息结构。

- HIGV_CURSORRES_TYPE_E：光标资源类型。
- HIGV_CLOCK_MODE_E：时间模式。
- HIGV_CLOCK_STYLE_S：时间风格结构。

HIGV_TIME_S

【说明】

时间信息结构。

【定义】

```
typedef struct hiHIGV_TIME_S
{
    HI_U32          year;
        HI_U32          month;
        HI_U32          day;
        HI_U32          week;
        HI_U32          hour;
        HI_U32          minute;
        HI_U32          second;
 }
HIGV_TIME_S;
```

【成员】

成员名称	描述
year	年。
month	月。
day	日。
week	周。
hour	时。
minute	分。
second	秒。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_CURSORRES_TYPE_E

【说明】

光标资源类型。

【定义】

```
typedef enum
{
    HIGV_CURSORRES_TYPE_COLOR=0,
            HIGV_CURSORRES_TYPE_IMAGE=1,
            HIGV_CURSORRES_TYPE_BUTT,
 }
HIGV_CURSORRES_TYPE_E
```

【成员】

成员名称	描述
HIGV_CURSORRES_TYPE_COLOR	颜色光标。
HIGV_CURSORRES_TYPE_IMAGE	图片光标。
HIGV_CURSORRES_TYPE_BUTT	非法值。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_CLOCK_MODE_E

【说明】

时间模式。

【定义】

```
typedef enum
{
    HIGV_CLOCK_MODE_TEXT=0,
        HIGV_CLOCK_MODE_FORMAT
        HIGV_CLOCK_MODE_BUTT,
 }
HIGV_CURSORRES_TYPE_E
```

【成员】

成员名称	描述
HIGV_CLOCK_MODE_TEXT	通过文本设置时间。
HIGV_CLOCK_MODE_FORMAT	通过时间格式方式来显示时间。
HIGV_CLOCK_MODE_BUTT	非法值。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_CLOCK_STYLE_S

【说明】

时间模式。

【定义】

```
typedef struct
{
    HIGV_CLOCK_MODE_E               DispMode;
        HIGV_CURSORRES_TYPE_E           CursorResType;
        HI_U32                          CursorRes;
 }
```


HIGV_CLOCK_STYLE_S

【成员】

成员名称	描述
DispMode	Clock 控件是否具有显示和编辑功能。
CursorResType	编辑焦点的资源类型。
CursorRes	可编辑窗口的资源 id，图片或颜色。

【注意事项】

无。

【相关数据类型及接口】

无。

2.2.8 Trackbar 控件

2.2.8.1 接口描述

HiGV Trackbar 控件提供 API：

- HI_GV_Track_SetSliImage：设置 trackbar 滑竿的皮肤。
- HI_GV_Track_SetTraImage：设置 trackbar 滑块的皮肤。
- HI_GV_Track_SetCurVal：设置 trackbar 当前值。
- HI_GV_Track_GetCurVal：获取 trackbar 当前值。
- HI_GV_Track_SetRange：设置游标的范围值。
- HI_GV_Track_GetRange：获取游标的范围值。
- HI_GV_Track_EnableGesture：设置控件是否支持触摸。
- HI_GV_Track_IsGestureEnable：获取控件是否支持触摸。

HI_GV_Track_SetSliImage

【描述】

设置 trackbar 滑杆的皮肤。

【语法】


```
HI_S32 HI_GV_Track_SetSliImage（HIGV_HANDLE trackBarHandle, HI_U32 handleSkinIndex,
HI_RESID handleRes）；
```

【参数】

参数名称	描述	输入/输出
trackBarHandle	控件句柄。	输入
handleSkinIndex	Track 皮肤状态。	输入
handleRes	Track 图片资源 id。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_trackbar.h
- 库文件：libhigv.a、libhigv.so

【注意】

滑竿目前不存在状态，handleSkinIndex 暂时未用到，做扩展保留。

【举例】

无。

HI_GV_Track_SetTraImage

【描述】

设置 trackbar 滑块的皮肤。

【语法】

```
HI_S32 HI_GV_Track_SetTraImage（HIGV_HANDLE trackBarHandle, HI_U32 handleSkinIndex,
```



```
HI_RESID handleRes）；
```

【参数】

参数名称	描述	输入/输出
trackBarHandle	控件句柄。	输入
handleSkinIndex	Track 皮肤状态。	输入
handleRes	Track 图片资源 id。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_trackbar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Track_SetCurVal

【描述】

设置 trackbar 当前值。

【语法】

```
HI_S32 HI_GV_Track_SetCurVal（HIGV_HANDLE trackBarHandle, HI_U32 value）；
```

【参数】

参数名称	描述	输入/输出
trackBarHandle	控件句柄。	输入
value	游标的当前值。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_trackbar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Track_GetCurVal

【描述】

获取 trackbar 当前值。

【语法】

```
HI_S32 HI_GV_Track_GetCurVal（HIGV_HANDLE trackBarHandle, HI_U32* value）；
```

【参数】

参数名称	描述	输入/输出
trackBarHandle	控件句柄。	输入

参数名称	描述	输入/输出
value	游标的当前值。	输出

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_trackbar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Track_SetRange

【描述】

设置游标的范围值。

【语法】

```
HI_S32 HI_GV_Track_SetRange（HIGV_HANDLE trackBarHandle, HI_U32 minValue, HI_U32 maxValue）；
```

【参数】

参数名称	描述	输入/输出
trackBarHandle	控件句柄。	输入
minValue	游标最小值。	输入

参数名称	描述	输入/输出
maxValue	游标最大值	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_trackbar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Track_GetRange

【描述】

获取游标的范围值。

【语法】

```
HI_S32 HI_GV_Track_GetRange (HIGV_HANDLE trackBarHandle, HI_U32* minValue, HI_U32* maxValue);
```

【参数】

参数名称	描述	输入/输出
trackBarHandle	控件句柄。	输入
minValue	游标最小值。	输出

参数名称	描述	输入/输出
maxValue	游标最大值。	输出

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_trackbar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Track_EnableGesture

【描述】

设置控件是否支持触摸。

【语法】

```
HI_S32 HI_GV_Track_EnableGesture(HIGV_HANDLE trackBarHandle, HI_BOOL isEnable);
```

【参数】

参数名称	描述	输入/输出
trackBarHandle	控件句柄。	输入
isEnable	是否支持触摸(HI_TRUE 表示支持，HI_FALSE 表示禁止)。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_trackbar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Track_IsGestureEnable

【描述】

获取控件是否支持触摸。

【语法】

```
HI_S32 HI_GV_Track_IsGestureEnable(HIGV_HANDLE trackBarHandle, HI_BOOL* isEnable);
```

【参数】

参数名称	描述	输入/输出
trackBarHandle	控件句柄。	输入
isEnable	是否支持触摸(HI_TRUE 表示支持，HI_FALSE 表示禁止，系统默认为 HI_TRUE)。	输出

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_trackbar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.2.8.2 数据类型

无。

2.2.9 Image 控件

2.2.9.1 接口描述

HiGV Image 控件提供 API：

- HI_GV_Image_SetImage：设置图片资源
- HI_GV_Image_DrawMemImage：绘制内存图片
- HI_GV_Image_DrawSurface：绘制 surface 图片
- HI_GV_Image_FreeMemSurface：释放内存图片的 surface
- HI_GV_Image_SetBlitOpt：设置图片的搬移混合操作运算
- HI_GV_Image_GetBlitOpt：获取图片的搬移混合操作运算

HI_GV_Image_SetImage

【描述】

设置图片资源。

【语法】

```
HI_S32 HI_GV_Image_SetImage(HIGV_HANDLE imageHandle, HI_RESID image);
```

【参数】

参数名称	描述	输入/输出
imageHandle	控件句柄。	输入
image	图片资源 ID。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_image.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Image_DrawMemImage

【描述】

绘制内存图片。

【语法】

```
HI_S32 HI_GV_Image_DrawMemImage(HIGV_HANDLE imageHandle, HI_GV_MemInfo*
memInfo, HI_U32 imageHeight, HI_U32 imageWidth, const HI_RECT* srcRect, const HI_RECT *
dstRect, const HIGO_BLTOPT_S * blitOpt，HI_BOOL transparent);
```


【参数】

参数名称	描述	输入/输出
imageHandle	控件句柄。	输入
memInfo	内存数据。	输入
imageHeight	解码图片高度，单位：像素。	输入
imageWidth	解码图片宽度，单位：像素。	输入
srcRect	源 surface 矩形区域。	输入
dstRect	目标 surface 的矩形区域。	输入
blitOpt	搬移混合操作运算属性。 详情请参考《HiGO API 参考文档》。	输入
transparent	是否透明皮肤。 HI_TRUE：透明； HI_FALSE：非透明。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_image.h
- 库文件：libhigv.a、libhigv.so

【注意】

需要和 HI_GV_Image_FreeMemSurface 配合使用。

【举例】

无。

HI_GV_Image_DrawSurface

【描述】

绘制 surface 图片。

【语法】

```
HI_S32 HI_GV_Image_DrawSurface(HIGV_HANDLE imageHandle, HIGV_HANDLE handleSrcSurface, HI_RECT* srcRect, HI_RECT * dstRect, HIGO_BLTOPT_S * blitOpt，HI_BOOL transparent);
```

【参数】

参数名称	描述	输入/输出
imageHandle	控件句柄。	输入
handleSrcSurface	Surface 句柄。	输入
srcRect	源 surface 矩形区域。	输入
dstRect	目标 surface 的矩形区域。	输入
blitOpt	搬移混合操作运算属性。 详情请参考《HiGO API 参考文档》。	输入
transparent	是否透明皮肤。 HI_TRUE：透明； HI_FALSE：非透明。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_image.h

- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Image_FreeMemSurface

【描述】

释放内存图片的 surface。

【语法】

```
HI_S32 HI_GV_Image_FreeMemSurface(HIGV_HANDLE imageHandle);
```

【参数】

参数名称	描述	输入/输出
imageHandle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_image.h
- 库文件：libhigv.a、libhigv.so

【注意】

需要和 HI_GV_Image_DrawMemImage 配合使用。

【举例】

无。

HI_GV_Image_SetBlitOpt

【描述】

设置图片的搬移混合操作运算。

【语法】

```
HI_S32 HI_GV_Image_SetBlitOpt(HIGV_HANDLE imageHandle, const HIGO_BLTOPT_S* blitOpt);
```

【参数】

参数名称	描述	输入/输出
imageHandle	控件句柄。	输入
blitOpt	搬移混合操作运算属性。 详情请参考《HiGO API 参考文档》。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_image.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_Image_GetBlitOpt

【描述】

获取图片的搬移混合操作运算。

【语法】

```
HI_S32 HI_GV_Image_GetBlitOpt(HIGV_HANDLE imageHandle, HIGO_BLTOPT_S* blitOpt）;
```

【参数】

参数名称	描述	输入/输出
imageHandle	控件句柄。	输入
blitOpt	搬移混合操作运算属性指针。 详情请参考《HiGO API 参考文档》。	输出

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_image.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.2.9.2 数据类型

相关数据类型、数据结构定义如下：

- HI_GV_MemInfo：图片内存属性。
- HIGO_BLTOPT_S：位块搬移。

HI_GV_MemInfo

【说明】

图片内存属性。

【定义】

```
typedef struct
{
    HI_CHAR*            pAddr;
    HI_U32              Length;
 }
HI_GV_MemInfo
```

【成员】

成员名称	描述
pAddr	内存地址。
Length	内存地址长度。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGO_BLTOPT_S

【说明】

位块搬移。

【定义】

```
typedef struct {
    HI_BOOL EnableGlobalAlpha;
    HI_BOOL EnablePixelAlpha;
```



```
    HIGO_COMPOPT_E PixelAlphaComp;
    HIGO_CKEY_E ColorKeyFrom;
    HI_BOOL EnableRop;
    HIGO_ROP_E Rop;
    HIGO_ROP_E RopAlpha;
    HI_BOOL EnableScale
    HIGO_ROTATE_E RotateType;
    HIGO_MIRROR_E MirrorType;
} HIGO_BLTOPT_S;
```

【成员】

成员名称	描述
EnableGlobalAlpha	全局 alpha 使能标志
EnablePixelAlpha	像素 alpha 使能标志
PixelAlphaComp	像素 alpha 操作
ColorKeyFrom	ColorKey 操作
EnableRop	启用 ROP2 操作
Rop	ROP2 操作类型
RopAlpha	ROP alpha 操作类型
EnableScale	启用缩放
RotateType	旋转方式
MirrorType	镜像方式

【注意事项】

位块搬移在每次搬移中存在以下约束关系：

- 色彩空间转换仅支持 YUV 到 RGB 的转换。
- 目标位图像素格式：RGB444、ARGB4444、RGB555、RGB565、ARGB1555、RGB888、ARGB8888。
- 不可同时设置 Alpha 混和与 ROP 运算。

- 不可同时设置缩放、旋转、镜像三者中的任意 2 种。
- 不可同时设置缩放、旋转、镜像与 Alpha 混和。
- 不可同时设置缩放、旋转、镜像与 ROP 运算。
- 不可同时设置缩放、旋转、镜像与 ColorKey。
- 缩放、旋转、镜像仅适用于源位图和目标位图像素格式都是 32 位和 16 位的情况。

【相关数据类型及接口】

无。

2.2.10 Imageex 控件

2.2.10.1 接口描述

HiGV Imageex 控件提供 API：

- HI_GV_ImageEx_SetImage：设置图片内容。
- HI_GV_ImageEx_SetPos：设置图片在控件中的显示位置。
- HI_GV_ImageEx_SetInterval：设置图片的显示时间间隔。
- HI_GV_ImageEx_SetRepeatCount：设置图片显示循环次数。

HI_GV_ImageEx_SetImage

【描述】

设置图片内容。

【语法】

```
HI_S32 HI_GV_ImageEx_SetImage(HIGV_HANDLE imageExHandle, const HI_CHAR* imgFile);
```

【参数】

参数名称	描述	输入/输出
imageExHandle	控件句柄。	输入
imgFile	图片文件路径。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_imageex.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ImageEx_SetPos

【描述】

设置图片在控件中的显示位置。

【语法】

```
HI_S32 HI_GV_ImageEx_SetPos(HIGV_HANDLE imageExHandle, HIGV_IMGPOS_E posType）；
```

【参数】

参数名称	描述	输入/输出
imageExHandle	控件句柄。	输入
posType	图片位置类型。	输入

【返回值】

返回值	描述
0	成功

返回值	描述
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_imageex.h
- 库文件：libhigv.a、libhigv.so

【注意】

此接口仅支持静态图片类型设置，GIF 格式图片不支持缩放。

【举例】

无。

HI_GV_ImageEx_SetInterval

【描述】

设置图片显示时间间隔。

【语法】

```
HI_S32 HI_GV_ImageEx_SetInterval(HIGV_HANDLE imageExHandle, HI_U32 interval);
```

【参数】

参数名称	描述	输入/输出
imageExHandle	控件句柄。	输入
interval	图片显示的时间间隔，单位为 ms。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_imageex.h
- 库文件：libhigv.a、libhigv.so

【注意】

需要 GIF 格式图片本身没有显示间隔才能生效。

【举例】

无。

HI_GV_ImageEx_SetRepeatCount

【描述】

设置图片显示循环次数。如：设置 gif 图片动画播放循环次数。

【语法】

```
HI_S32 HI_GV_ImageEx_SetRepeatCount(HIGV_HANDLE imageExHandle, HI_S32 repeatCount);
```

【参数】

参数名称	描述	输入/输出
imageExHandle	控件句柄。	输入
repeatCount	循环次数，-1 为无限次循环。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_imageex.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.2.10.2 数据类型

相关数据类型、数据结构定义如下：

HIGV_IMGPOS_E：图片的对齐方式。

HIGV_IMGPOS_E

【说明】

图片的对齐方式。

【定义】

```
typedef enum
{
    HIGV_IMGPOS_CENTER,
    HIGV_IMGPOS_STRETCH,
    HIGV_IMGPOS_AUTOSTRETCH,
    HIGV_IMGPOS_BUTT
}
HIGV_IMGPOS_E
```

【成员】

成员名称	描述
HIGV_IMGPOS_CENTER	自动居中。
HIGV_IMGPOS_STRETCH	自动拉伸。
HIGV_IMGPOS_AUTOSTRETCH	自动按比例拉伸。
HIGV_IMGPOS_BUTT	非法值。

【注意事项】

无。

【相关数据类型及接口】

无。

2.2.11 Progressbar 控件

2.2.11.1 接口描述

HiGV Progressbar 控件提供 API：

- HI_GV_ProgressBar_SetRange：设置进度条范围。
- HI_GV_ProgressBar_SetStep：设置步长。
- HI_GV_ProgressBar_SetPos：设置进度条当前位置。
- HI_GV_ProgressBar_GetPos：获取进度条当前位置。
- HI_GV_ProgressBar_SetFg：设置进度条的前景风格。

HI_GV_ProgressBar_SetRange

【描述】

设置进度条范围。

【语法】

```
HI_S32 HI_GV_ProgressBar_SetRange(HIGV_HANDLE progressBarHandle, HI_U32 min, HI_U32 max);
```

【参数】

参数名称	描述	输入/输出
progressBarHandle	控件句柄。	输入
min	最小值。	输入
max	最大值。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_progressbar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ProgressBar_SetStep

【描述】

设置步长。

【语法】

```
HI_S32 HI_GV_ProgressBar_SetStep(HIGV_HANDLE progressBarHandle, HI_U32 step);
```

【参数】

参数名称	描述	输入/输出
progressBarHandle	控件句柄。	输入
step	步长。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_progressbar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ProgressBar_SetPos

【描述】

设置进度条当前位置。

【语法】

```
HI_S32 HI_GV_ProgressBar_SetPos(HIGV_HANDLE progressBarHandle, HI_U32 pos);
```

【参数】

参数名称	描述	输入/输出
progressBarHandle	控件句柄。	输入
pos	位置。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_progressbar.h

- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ProgressBar_GetPos

【描述】

获取进度条当前位置。

【语法】

```
HI_S32 HI_GV_ProgressBar_GetPos(HIGV_HANDLE progressBarHandle, HI_U32* pos);
```

【参数】

参数名称	描述	输入/输出
progressBarHandle	控件句柄。	输入
pos	位置。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_progressbar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ProgressBar_SetFg

【描述】

设置进度条的前景风格。

【语法】

```
HI_S32 HI_GV_ProgressBar_SetFg(HIGV_HANDLE progressBarHandle, const HI_RECT* fgRect,
HI_RESID fgStyle);
```

【参数】

参数名称	描述	输入/输出
progressBarHandle	控件句柄。	输入
fgRect	进度条前景显示的最大范围。	输入
fgStyle	风格句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_resm.h、hi_gv_progressbar.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.2.11.2 数据类型

无。

2.2.12 WheelView 控件

2.2.12.1 接口描述

HiGV WheelView 控件提供 API：

- HI_GV_WheelView_Create：WheelView 控件初始化。
- HI_GV_WheelView_SetSelItem：设置 WheelView 当前条目。
- HI_GV_WheelView_GetSelItem：获取 WheelView 当前条目。
- HI_GV_WheelView_GetItemNum：获取 WheelView 当前数据总条数。
- HI_GV_WheelView_SetUpImage：设置上遮挡皮肤资源。
- HI_GV_WheelView_SetDownImage：设置下遮挡皮肤资源。
- HI_GV_WheelView_Enable：设置允许触摸功能。
- HI_GV_WheelView_IsEnable：获取是否允许触摸状态。
- HI_GV_WheelView_SetScrollParam：设置滑动灵敏度系数。
- HI_GV_WheelView_SetFlingParam：设置轻扫灵敏度系数。

HI_GV_WheelView_Create

【描述】

WheelView 控件初始化。

【语法】

```
HI_S32 HI_GV_WheelView_Create(const HIGV_WCREATE_S* creatInfo, HIGV_HANDLE*
wheelViewHandle);
```

【参数】

参数名称	描述	输入/输出
creatInfo	控件风格布局属性指针。	输入
wheelViewHandle	控件句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_wheelview.h
- 库文件：libhigv.a、libhigv.so

【注意】

开发者调用 HI_GV_WheelView_Create 创建的资源，使用结束后，由开发者主动调用 HI_GV_Widget_Destroy 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_WheelView_SetSelItem

【描述】

设置 WheelView 当前条目。

【语法】

```
HI_S32    HI_GV_WheelView_SetSelItem(HIGV_HANDLE wheelViewHandle, HI_U32 item);
```

【参数】

参数名称	描述	输入/输出
wheelViewHandle	控件句柄。	输入
item	条目编号。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_wheelview.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WheelView_GetSelItem

【描述】

获取 WheelView 当前条目。

【语法】

```
HI_S32    HI_GV_WheelView_GetSelItem(HIGV_HANDLE wheelViewHandle, HI_U32* item);
```

【参数】

参数名称	描述	输入/输出
wheelViewHandle	控件句柄。	输入
item	条目编号。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_wheelview.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WheelView_GetItemNum

【描述】

获取 WheelView 当前数据总条数。

【语法】

```
HI_S32    HI_GV_WheelView_GetItemNum(HIGV_HANDLE wheelViewHandle, HI_U32* itemNum);
```

【参数】

参数名称	描述	输入/输出
wheelViewHandle	控件句柄。	输入
itemNum	总条目数。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件： hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_wheelview.h

- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WheelView_SetUpImage

【描述】

设置上遮挡皮肤资源。

【语法】

```
HI_S32 HI_GV_WheelView_SetUpImage(HIGV_HANDLE wheelViewHandle, HI_RESID handleRes);
```

【参数】

参数名称	描述	输入/输出
wheelViewHandle	控件句柄。	输入
handleRes	图片资源句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_wheelview.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WheelView_SetDownImage

【描述】

设置下遮挡皮肤资源。

【语法】

```
HI_S32 HI_GV_WheelView_SetDownImage(HIGV_HANDLE wheelViewHandle, HI_RESID handleRes);
```

【参数】

参数名称	描述	输入/输出
wheelViewHandle	控件句柄。	输入
handleRes	图片资源句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_wheelview.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WheelView_Enable

【描述】

设置允许触摸功能。

【语法】

```
HI_S32 HI_GV_WheelView_Enable(HIGV_HANDLE wheelViewHandle, HI_BOOL enable);
```

【参数】

参数名称	描述	输入/输出
wheelViewHandle	控件句柄。	输入
enable	使能标志。 HI_TRUE：使能； HI_FALSE：去使能。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_wheelview.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WheelView_IsEnable

【描述】

获取是否允许触摸状态。

【语法】

```
HI_BOOL HI_GV_WheelView_IsEnable(HIGV_HANDLE wheelViewHandle);
```

【参数】

参数名称	描述	输入/输出
wheelViewHandle	控件句柄。	输入

【返回值】

返回值	描述
HI_TRUE	触摸可用。
HI_FALSE	触摸禁用。

【需求】

- 头文件：hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_wheelview.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WheelView_SetScrollParam

【描述】

设置滑动灵敏度系数(默认为 1.0，大于等于 1.0)。

【语法】


```
HI_S32 HI_GV_WheelView_SetScrollParam(HIGV_HANDLE wheelViewHandle, HI_FLOAT scrollParam);
```

【参数】

参数名称	描述	输入/输出
wheelViewHandle	控件句柄。	输入
scrollParam	滑动灵敏度系数。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_wheelview.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_WheelView_SetFlingParam

【描述】

设置轻扫灵敏度系数(默认为 1.0，大于等于 1.0)。

【语法】

```
HI_S32 HI_GV_WheelView_SetFlingParam(HIGV_HANDLE wheelViewHandle, HI_FLOAT flingParam);
```

【参数】

参数名称	描述	输入/输出
wheelViewHandle	控件句柄。	输入
flingParam	轻扫灵敏度系数。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_wheelview.h
- 库文件：libhigv.a、libhigv.so

【注意】

灵敏度系数控制的是滑动的响应速度。

【举例】

无。

2.2.12.2 数据类型

相关数据类型、数据结构定义如下：

HigvWheelViewStyle：WheelView 控件属性。

HigvWheelViewStyle

【说明】

WheelView 控件属性。

【定义】

```
typedef struct
{
    HI_U32      RowNum;
```



```
    HIGV_HANDLE DataModel;
    HIGV_HANDLE UpCoverSkin;
    HIGV_HANDLE DownCoverSkin;
    HI_BOOL isCircleScroll;
    HI_FLOAT   ScrollParam;
    HI_FLOAT   FlingParam;
    HI_U32 LeftMargin;
    HI_U32 RightMargin;
    HI_U32 TopMargin;
    HI_U32 BtmMargin;
} HigvWheelViewStyle;
```

【成员】

成员名称	描述
RowNum	屏幕上显示的行数。
DataModel	数据模型句柄。
UpCoverSkin	上覆盖皮肤句柄。
DownCoverSkin	下覆盖皮肤句柄。
isCircleScroll	是否循环滚动。 HI_TRUE:循环滚动; HI_FALSE:非循环滚动。
ScrollParam	滑动灵敏度系数。
FlingParam	轻扫灵敏度系数。
LeftMargin	左边距。
RightMargin	右边距。
TopMargin	上边距。
BtmMargin	下边距。

【注意事项】

无。

【相关数据类型及接口】

无。

2.2.13 SlideUnlock 控件

2.2.13.1 接口描述

HiGV SlideUnlock 控件提供 API：

- HI_GV_SlideUnlock_SetSliImage：设置 SlideUnlock 滑杆的皮肤。
- HI_GV_SlideUnlock_SetTraImage：设置 SlideUnlock 滑块的皮肤。
- HI_GV_SlideUnlock_SetStatus：设置 SlideUnlock 业务的使能状态。
- HI_GV_SlideUnlock_GetStatus：获取 SlideUnlock 业务的使能状态。
- HI_GV_SlideUnlock_ReSet：恢复到初始状态。

HI_GV_SlideUnlock_SetSliImage

【描述】

设置 SlideUnlock 滑杆的皮肤。

【语法】

```
HI_S32 HI_GV_SlideUnlock_SetSliImage(HIGV_HANDLE handleSlideUnlock, HI_RESID handleRes);
```

【参数】

参数名称	描述	输入/输出
handleSlideUnlock	控件句柄。	输入
handleRes	图片资源 ID。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件： hi_go.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_slideunlock.h
- 库文件：libhigv.a、libhigv.so

【注意】

hRes 为图片资源 ID，使用前需调用 HI_GV_Res_CreateID 接口创建。

【举例】

无。

HI_GV_SlideUnlock_SetTraImage

【描述】

设置 SlideUnlock 滑块的皮肤。

【语法】

```
HI_S32 HI_GV_SlideUnlock_SetTraImage(HIGV_HANDLE handleSlideUnlock, HI_U32 handleSkinIndex, HI_RESID handleRes);
```

【参数】

参数名称	描述	输入/输出
handleSlideUnlock	控件句柄。	输入
handleSkinIndex	皮肤状态。	输入
handleRes	图片资源 ID。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_go.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_slideunlock.h
- 库文件：libhigv.a、libhigv.so

【注意】

- hSkinIndex 为皮肤状态，类型为 HIGV_SLIDEUNLOCK_E。
- hRes 为图片资源 ID，使用前需调用 HI_GV_Res_CreateID 接口创建。

【举例】

无。

HI_GV_SlideUnlock_SetStatus

【描述】

设置 SlideUnlock 业务的使能状态。

【语法】

```
HI_S32 HI_GV_SlideUnlock_SetStatus(HIGV_HANDLE handleSlideUnlock, HI_BOOL isEnable);
```

【参数】

参数名称	描述	输入/输出
handleSlideUnlock	控件句柄。	输入
isEnable	使能标志。 HI_TRUE：使能； HI_FALSE：禁用。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_go.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_slideunlock.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_SlideUnlock_GetStatus

【描述】

获取 SlideUnlock 业务的使能状态。

【语法】

```
HI_S32 HI_GV_SlideUnlock_GetStatus(HIGV_HANDLE handleSlideUnlock, HI_U32* isEnable);
```

【参数】

参数名称	描述	输入/输出
handleSlideUnlock	控件句柄。	输入
isEnable	使能状态。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_go.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_slideunlock.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_SlideUnlock_ReSet

【描述】

重置滑块位置，恢复到 normal 状态皮肤。

【语法】

```
HI_S32 HI_GV_SlideUnlock_ReSet(HIGV_HANDLE handleSlideUnlock);
```

【参数】

参数名称	描述	输入/输出
handleSlideUnlock	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_go.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_slideunlock.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.2.13.2 数据类型

相关数据类型、数据结构定义如下：

- SLIDEUNLOCK_STYLE：控件风格方向。
- HIGV_SLIDEUNLOCK_E：滑块控件种类。

SLIDEUNLOCK_STYLE

【说明】

控件风格方向，水平或垂直。

【定义】

```
typedef enum HI_SLIDEUNLOCK_STYLE
{
    SLIDEUNLOCK_H = 0,
    SLIDEUNLOCK_V,
    SLIDEUNLOCK_BUT
} SLIDEUNLOCK_STYLE;
```

【成员】

成员名称	描述
SLIDEUNLOCK_H	水平方向。
SLIDEUNLOCK_V	垂直方向。

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_SLIDEUNLOCK_E

【说明】

滑块控件种类。

【定义】

```
typedef enum hiHIGV_SLIDEUNLOCK_E
{
    HIGV_SLIDEUNLOCK_NORMAL = 0,
    HIGV_SLIDEUNLOCK_TOUCH,
    HIGV_SLIDEUNLOCK_DONE,
    HIGV_SLIDEUNLOCK_BUTT
} HIGV_SLIDEUNLOCK_E;
```

【成员】

成员名称	描述
HIGV_SLIDEUNLOCK_NORMAL	滑块处于初始状态。
HIGV_SLIDEUNLOCK_TOUCH	滑块处于移动状态。
HIGV_SLIDEUNLOCK_DONE	滑块处于解锁完成状态。

【注意事项】

无。

【相关数据类型及接口】

无。

2.2.14 ScaleView 控件

2.2.14.1 接口描述

HiGV ScaleView 控件提供 API：

- HI_GV_SCALEVIEW_Create：ScaleView 控件初始化。
- HI_GV_SCALEVIEW_SetSelItem：设置 ScaleView 初始中间行条目号。
- HI_GV_SCALEVIEW_GetSelItem：获取 ScaleView 初始中间行条目号。
- HI_GV_SCALEVIEW_GetItemNum：获取 ScaleView 数据源总条目数。
- HI_GV_SCALEVIEW_GetTapItem：获取 ScaleView 控件 Tap 操作后焦点条目号。

- HI_GV_SCALEVIEW_GetFoucItem：获取 ScaleView 滑动条目变化时的中间行条目号。

HI_GV_SCALEVIEW_Create

【描述】

ScaleView 控件初始化。

【语法】

```
HI_S32 HI_GV_SCALEVIEW_Create(const HIGV_WCREATE_S *creatInfo, HIGV_HANDLE *handleScaleView);
```

【参数】

参数名称	描述	输入/输出
creatInfo	控件风格布局属性指针。	输入
handleScaleView	控件句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_scaleview.h
- 库文件：libhigv.a、libhigv.so

【注意】

开发者调用 HI_GV_WheelView_Create 创建的资源，使用结束后，由开发者主动调用 HI_GV_Widget_Destroy 来释放响应资源，否则会有内存泄露。

【举例】

无。

HI_GV_SCALEVIEW_SetSelItem

【描述】

设置 scaleView 初始中间行条目号。

【语法】

```
HI_S32 HI_GV_SCALEVIEW_SetSelItem(HIGV_HANDLE handleScaleView, HI_U32 item);
```

【参数】

参数名称	描述	输入/输出
handleScaleView	控件句柄。	输入
item	条目编号。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_scaleview.h
- 库文件：libhigv.a、libhigv.so

【注意】

条目号从 0 开始计数，初始中间行默认为焦点行，tap 手势选中某一行后，该行变为焦点行。

【举例】

无。

HI_GV_SCALEVIEW_GetSelItem

【描述】

获取 ScaleView 初始中间行条目号。

【语法】

```
HI_S32 HI_GV_SCALEVIEW_GetSelItem(HIGV_HANDLE handleScaleView, HI_U32 *selItem);
```

【参数】

参数名称	描述	输入/输出
handleScaleView	控件句柄。	输入
selItem	条目编号。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_scaleview.h
- 库文件：libhigv.a、libhigv.so

【注意】

条目号从 0 开始计数，计数包含控件上下空白行行数，空白行行数为（n - 1）/ 2，n 为页面显示的行数。初始中间行默认为焦点行，tap 手势选中某一行后，该行变为焦点行。

【举例】

无。

HI_GV_SCALEVIEW_GetItemNum

【描述】

获取 ScaleView 数据源总条目数。

【语法】


```
HI_S32 HI_GV_SCALEVIEW_GetItemNum(HIGV_HANDLE handleScaleView, HI_U32 *itemNum);
```

【参数】

参数名称	描述	输入/输出
handleScaleView	控件句柄。	输入
itemNum	总条目数。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_scaleview.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_SCALEVIEW_GetTapItem

【描述】

获取 ScaleView 控件 Tap 操作后焦点条目号。

【语法】

```
HI_S32 HI_GV_SCALEVIEW_GetTapItem(HIGV_HANDLE handleScaleView, HI_U32 *tapItem);
```

【参数】

参数名称	描述	输入/输出
handleScaleView	控件句柄。	输入
tapItem	点击的条目的编号。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_scaleview.h
- 库文件：libhigv.a、libhigv.so

【注意】

条目号从 0 开始计数，计数包含控件上下空白行行数，空白行行数为（n - 1）/ 2，n 为页面显示的行数。

【举例】

无。

HI_GV_SCALEVIEW_GetFoucItem

【描述】

获取 ScaleView 滑动条目变化时的中间行条目号。

【语法】

```
HI_S32 HI_GV_SCALEVIEW_GetFoucItem(HIGV_HANDLE handleScaleView, HI_U32 *itemNum);
```

【参数】

参数名称	描述	输入/输出
handleScaleView	控件句柄。	输入

参数名称	描述	输入/输出
itemNum	焦点的条目编号。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv.h、hi_gv_conf.h、hi_gv_resm.h、hi_gv_scaleview.h
- 库文件：libhigv.a、libhigv.so

【注意】

条目号从 0 开始计数，计数包含控件上下空白行行数，空白行行数为（n - 1）/ 2，n 为页面显示的行数。

【举例】

无。

2.2.14.2 数据类型

相关数据类型、数据结构定义如下：

HIGV_SCALEVIEW_STYLE_S：ScaleView 控件属性。

HIGV_SCALEVIEW_STYLE_S

【说明】

ScaleView 控件属性。

【定义】

```
typedef struct HIGV_SCALEVIEW_STYLE_S {
    HIGV_HANDLE dataModel;
    HI_U32 rowNum;
```



```
    HI_U32 leftMargin;
    HI_U32 rightMargin;
    HI_U32 topMargin;
    HI_U32 bottomMargin;
    HI_U32 imgDecIndex;
    HI_U32 lineHeight;
    HI_U32 lineWidth;
    HI_U32 lineDistance;
    HI_U32 hlineColor;
    HI_U32 type;
    HIGV_HANDLE scaleFont;
    HI_COLOR tapTxtColor;
    HI_DOUBLE sizeGrain;
    HI_BOOL isNeedTransform;
    HI_BOOL tapAutoMove;
    HI_DOUBLE imageSize;
    HI_U32 imageSizeGrain;
} HIGV_SCALEVIEW_STYLE;
```

【成员】

成员名称	描述
dataModel	数据模型句柄。
rowNum	屏幕上显示的行数。
leftMargin	左边距。
rightMargin	右边距。
topMargin	上边距。
bottomMargin	下边距。
imgDecIndex	图片解码索引号。
lineHeight	条目内容下水平网格线高度。
lineWidth	条目内容下水平网格线宽度。
lineDistance	条目内容下水平网格线到条目内容的距离。

成员名称	描述
hlineColor	条目内容下水平网格线颜色。
type	条目内容的类型。
scaleFont	字体大小。
tapTxtColor	点击后获得焦点的条目字体颜色。
sizeGrain	字体大小变化粒度。
isNeedTransform	被点击条目后是否需要转换。
tapAutoMove	被点击条目后是否需要自动搬移。
imageSize	单元格图片高度。
imageSizeGrain	图片大小变化粒度。

【注意事项】

无。

【相关数据类型及接口】

无。

2.2.15 输入法窗口控件

2.2.15.1 接口描述

HiGV 输入法控件提供 API：

- HI_GV_IMEWINDOW_Create：创建输入法窗口。
- HI_GV_IMEWINDOW_Destroy：销毁输入法窗口。
- HI_GV_IMEWINDOW_Change：切换输入法。
- HI_GV_IMEWINDOW_Enable：打开/关闭指定输入法。
- HI_GV_IMEWINDOW_SetSwitchIMWin：初始化时设置输入法切换窗口句柄。
- HI_GV_IMEWINDOW_SetSoftKB：初始化时绑定软键盘到输入法窗口。
- HI_GV_IMEWINDOW_IsShowIM：获取当前输入法窗口的显示状态。
- HI_GV_IMEWINDOW_IsShowSoftKB：获取当前软键盘窗口软键盘的显示状态。

- HI_GV_IMEWINDOW_SetDispSoftKB：设置软键盘窗口软键盘的显示状态。
- HI_GV_IMEWINDOW_GetIMEActiveWidget：获取当前激活输入法的控件。
- HI_GV_IMEWINDOW_GetCurrentIME：获取当前输入法。
- HI_GV_IMEWINDOW_SetKeyValue：设置软键盘窗口的键值映射。
- HI_GV_IMEWINDOW_PinSoftKB：设置软键盘显示在屏幕的固定位置。
- HI_GV_IMEWINDOW_UnPinSoftKB：不固定显示软键盘显示位置。
- HI_GV_IMEWINDOW_FlexibleSoftKB：灵活设置输入法窗口和软键盘窗口的位置。

HI_GV_IMEWINDOW_Create

【描述】

创建输入法窗口。

【语法】

```
HI_S32 HI_GV_IMEWINDOW_Create(const HIGV_WCREATE_S * creatInfo, HigvIMEWindowInit * initData, HIGV_HANDLE * handle);
```

【参数】

参数名称	描述	输入/输出
creatInfo	输入法窗口初始化信息指针。	输入
initData	输入法窗口初始化私有信息指针。	输入
handle	创建的输入法窗口句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_imewindow.h
- 库文件：libhigv.a、libhigv.so

【注意】

输入法的窗口宽度和位置会根据软键盘的位置和大小自适应，输入法的高度根据字体、logo 及箭头的大小自适应。

【举例】

无。

HI_GV_IMEWINDOW_Destroy

【描述】

销毁输入法窗口。

【语法】

```
HI_S32 HI_GV_IMEWINDOW_Destroy(HIGV_HANDLE handle);
```

【参数】

参数名称	描述	输入/输出
handle	输入法窗口控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_imewindow.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_IMEWINDOW_Change

【描述】

切换输入法。

【语法】

```
HI_S32 HI_GV_IMEWINDOW_Change(HIGV_HANDLE handle, HigvIMEWindowType imeType);
```

【参数】

参数名称	描述	输入/输出
handle	输入法窗口控件句柄。	输入
imeType	输入法类型。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_imewindow.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_IMEWINDOW_Enable

【描述】

打开/关闭指定输入法。

【语法】

```
HI_S32 HI_GV_IMEWINDOW_Enable(HigvIMEWindowType imeType, HI_BOOL isEnable);
```

【参数】

参数名称	描述	输入/输出
imeType	输入法类型。	输入
isEnable	HI_TRUE 表示使能；HI_FALSE 表示禁用。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_imewindow.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_IMEWINDOW_SetSwitchIMWin

【描述】

初始化时设置输入法切换窗口句柄。

【语法】

```
HI_S32 HI_GV_IMEWINDOW_SetSwitchIMWin(HIGV_HANDLE handle, HIGV_HANDLE
switchWindowHandle);
```

【参数】

参数名称	描述	输入/输出
handle	输入法窗口控件句柄。	输入
switchWindowHandle	输入法切换窗口句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_imewindow.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_IMEWINDOW_SetSoftKB

【描述】

初始化时绑定软键盘到输入法窗口。

【语法】

```
HI_S32 HI_GV_IMEWINDOW_SetSoftKB(HIGV_HANDLE handle, const HigvIMEWindowSoftKB
*softKBArray, HI_U32 imeNum);
```


【参数】

参数名称	描述	输入/输出
handle	输入法窗口控件句柄。	输入
softKBArray	需要绑定的软键盘指针。	输入
imeNum	绑定输入法数目。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_imewindow.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_IMEWINDOW_IsShowIM

【描述】

获取当前输入法窗口的显示状态。

【语法】

```
HI_BOOL HI_GV_IMEWINDOW_IsShowIM(HIGV_HANDLE handle);
```

【参数】

参数名称	描述	输入/输出
handle	输入法窗口控件句柄。	输入

【返回值】

返回值	描述
HI_TRUE	显示。
HI_FALSE	隐藏。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_imewindow.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_IMEWINDOW_IsShowSoftKB

【描述】

获取当前软键盘窗口软键盘的显示状态。

【语法】

```
HI_BOOL HI_GV_IMEWINDOW_IsShowSoftKB(HIGV_HANDLE handle);
```

【参数】

参数名称	描述	输入/输出
handle	输入法窗口控件句柄。	输入

【返回值】

返回值	描述
HI_TRUE	显示。
HI_FALSE	隐藏。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_imewindow.h
- 库文件：libhigv.a、libhigv.so

【注意】

无

【举例】

无

HI_GV_IMEWINDOW_SetDispSoftKB

【描述】

设置软键盘窗口软键盘的显示状态。

【语法】

```
HI_S32 HI_GV_IMEWINDOW_SetDispSoftKB(HIGV_HANDLE handle, HIGV_HANDLE softKBHandle, HI_BOOL isShow);
```

【参数】

参数名称	描述	输入/输出
handle	输入法窗口控件句柄。	输入
softKBHandle	软键盘窗口句柄。	输入
isShow	显示状态。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_imewindow.h
- 库文件：libhigv.a、libhigv.so

【注意】

不支持动态设置，需要在输入法窗口 show 之前设置软键盘的显示状态。

【举例】

无。

HI_GV_IMEWINDOW_GetIMEActiveWidget

【描述】

获取当前激活输入法的控件。

【语法】

```
HI_S32 HI_GV_IMEWINDOW_GetIMEActiveWidget(HIGV_HANDLE handle, HIGV_HANDLE *activeWidget);
```

【参数】

参数名称	描述	输入/输出
handle	输入法窗口控件句柄。	输入
activeWidget	和输入法窗口绑定的控件句柄指针。	输出

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_imewindow.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_IMEWINDOW_GetCurrentIME

【描述】

获取当前输入法类型。

【语法】

```
HI_S32 HI_GV_IMEWINDOW_GetCurrentIME(HIGV_HANDLE handle, HigvIMEWindowType *imeType);
```

【参数】

参数名称	描述	输入/输出
handle	输入法窗口控件句柄。	输入
imeType	输入法类型指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_imewindow.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_IMEWINDOW_SetKeyValue

【描述】

设置软键盘窗口的键值映射。

【语法】

```
HI_S32 HI_GV_IMEWINDOW_SetKeyValue(HigvIMEWindowType imeType, const HI_U32 *keyValue, HI_U32 num);
```

【参数】

参数名称	描述	输入/输出
imeType	重设的输入法。	输入
keyValue	映射指针。	输入
num	重设的数目。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_imewindow.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无

HI_GV_IMEWINDOW_PinSoftKB

【描述】

设置软键盘显示在屏幕的固定位置。

【语法】

```
HI_S32 HI_GV_IMEWINDOW_PinSoftKB(HIGV_HANDLE handle, HI_S32 x, HI_S32 y);
```

【参数】

参数名称	描述	输入/输出
handle	输入法窗口控件句柄。	输入
x	屏幕坐标 x。	输入
y	屏幕坐标 y	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_imewindow.h

- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_IMEWINDOW_UnPinSoftKB

【描述】

不固定显示软键盘显示位置。

【语法】

```
HI_S32 HI_GV_IMEWINDOW_UnPinSoftKB(HIGV_HANDLE handle);
```

【参数】

参数名称	描述	输入/输出
handle	输入法窗口控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_imewindow.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_IMEWINDOW_FlexibleSoftKB

灵活设置输入法窗口和软键盘窗口的位置。

【语法】

```
HI_S32 HI_GV_IMEWINDOW_FlexibleSoftKB(HIGV_HANDLE handle, HI_BOOL isFlexible);
```

【参数】

参数名称	描述	输入/输出
handle	输入法窗口控件句柄。	输入
isFlexible	灵活设置输入法窗口和软键盘窗口的位置的开关	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_imewindow.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.2.15.2 数据类型

相关数据类型、数据结构定义如下：

- HigvIMEWindowInit：输入法窗口初始化私有信息。

- HigvIMEWindowType：输入法类型。
- HigvIMEWindowKeyValuePair：控件键值映射。
- HigvIMEWindowSoftKB：软键盘控制信息。

HigvIMEWindowInit

【说明】

输入法窗口初始化私有信息。

【定义】

```
typedef struct {
    HIGO_PF_E pixelFormat;
    HI_PARAM englishLogoIndex; /* english input method logo */
    HI_PARAM capEnglishLogoIndex; /* Caps english input method logo */
    HI_PARAM symbolLogoIndex; /* Symbol input method logo */
    HI_PARAM numberLogoIndex; /* Number input method LOGO */
    HI_PARAM pinyinLogoIndex; /* Pinyin input method LOGO */
    HI_PARAM standardALogoIndex; /* Standard input method A logo */
    HI_PARAM standardBLogoIndex; /* Standard input method B logo */
    HI_PARAM leftArrowPicIndex; /* Left arrow picture */
    HI_PARAM rightArrowPicIndex; /* Right arrow picture */
    HI_PARAM unActiveLeftArrowPicIndex; /* Unactive status of left arrow */
    HI_PARAM unActiveRightArrowPicIndex; /* Unactive status of right arrow */
    HI_PARAM pinYinTablePath; /* path of pinyin code table */
    HI_PARAM pinYinTablePathLen; /* path length of pinyin code table */
    HigvIMEWindowInitEx *exData; /* Change input method size */
} HigvIMEWindowInit;
```

【成员】

成员名称	描述
pixelFormat	像素格式。 详情请参考《HIGO API 参考》
englishLogoIndex	英文输入法 logo 索引。
capEnglishLogoInde	大写英文输入法 logo 索引。

成员名称	描述
symbolLogoIndex	符号输入法 logo 索引。
numberLogoIndex	数字输入法 logo 索引。
pinyinLogoIndex	拼音输入法 logo 索引。
standardALogoIndex	标准输入法 A logo 索引
standardBLogoInde	标准输入法 B logo 索引
leftArrowPicIndex	左箭头图片资源索引
rightArrowPicIndex	右箭头图片资源索引
unActiveLeftArrowPicIndex	左箭头非活动状态图片索引
unActiveRightArrowPicIndex	右箭头非活动状态图片资源索引
pinYinTablePath	拼音码表路径
pinYinTablePathLen	拼音码表路径长度
exData	更改输入法大小

【注意事项】

无。

【相关数据类型及接口】

无。

HigvIMEWindowType

【说明】

输入法类型。

【定义】

```
typedef enum {
    HI_GV_IMEWindow_ENGLISH = 0,
    HI_GV_IMEWindow_CAPENGLISH,
    HI_GV_IMEWindow_NUMBER,
```



```
    HI_GV_IMEWindow_PINYIN,
    HI_GV_IMEWindow_SYBMOL,
    HI_GV_IMEWindow_STANDARD_A,
    HI_GV_IMEWindow_STANDARD_B,
    HI_GV_IMEWindow_BUTT
} HigvIMEWindowType;
```

【成员】

成员名称	描述
HI_GV_IMEWindow_ENGLISH	小写英文输入。
HI_GV_IMEWindow_CAPENGLISH	大写英文输入。
HI_GV_IMEWindow_NUMBER	数字输入。
HI_GV_IMEWindow_PINYIN	拼音输入。
HI_GV_IMEWindow_SYBMOL	符号输入。
HI_GV_IMEWindow_STANDARD_A	标准键盘 A 类型输入。
HI_GV_IMEWindow_STANDARD_B	标准键盘 B 类型输入。

【注意事项】

无。

【相关数据类型及接口】

无。

HigvIMEWindowKeyValuePair

【说明】

控件键值映射。

【定义】

```
typedef struct {
    HIGV_HANDLE keyWidget;
    HI_U32 keyValue;
} HigvIMEWindowKeyValuePair;
```


【成员】

成员名称	描述
keyWidget	绑定的 button 控件句柄。
keyValue	键值。

【注意事项】

无。

【相关数据类型及接口】

无。

HigvIMEWindowSoftKB

【说明】

软键盘控制信息。

【定义】

```
typedef struct {
    HI_BOOL isNoDispIMW;
    HIGV_HANDLE softKB;
    HI_BOOL isDispSoftKB;
    HigvIMEWindowKeyValuePair *keyValueMap;
    HI_U32 keyNum;
} HigvIMEWindowSoftKB;
```

【成员】

成员名称	描述
isNoDispIMW	是否显示输入法窗口。
softKB	软键盘窗口句柄。
isDispSoftKB	是否显示软键盘。
keyValueMap	控件键值映射数组指针。
keyNum	按键数目。

【注意事项】

无。

【相关数据类型及接口】

无。

2.2.16 Edit 控件

2.2.16.1 接口描述

HiGV Edit 控件提供 API：

- HI_GV_EDIT_SetCountLimit：设置可编辑字符个数。
- HI_GV_EDIT_GetCountLimit：获取可编辑字符个数。
- HI_GV_EDIT_SetCursorIndex：设置光标在文本内容中的字节索引。
- HI_GV_EDIT_GetCursorIndex：获取光标在文本内容中的字节索引。
- HI_GV_EDIT_SetCursorPos：设置光标位置。
- HI_GV_EDIT_GetCursorPos：获取光标位置。
- HI_GV_EDIT_SetContent：设置 Edit 控件当前显示内容。
- HI_GV_EDIT_GetContent：获取 Edit 控件当前显示内容。
- HI_GV_EDIT_SetType：设置控件的风格,只能设置一次。
- HI_GV_EDIT_GetType：获取控件的风格。
- HI_GV_EDIT_ShowReplace：设置替换风格是否显示内容。
- HI_GV_EDIT_CleanContent：清除编辑框内内容。

HI_GV_EDIT_SetCountLimit

【描述】

设置可编辑字符个数。

【语法】

```
HI_S32 HI_GV_EDIT_SetCountLimit(HIGV_HANDLE handle, HI_U32 countLimit);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄。	输入
countLimit	可编辑字符个数。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_edit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_EDIT_GetCountLimit

【描述】

获取可编辑字符个数。

【语法】

```
HI_S32 HI_GV_EDIT_GetCountLimit(HIGV_HANDLE handle, HI_U32 *count);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄。	输入
count	Edit 控件可编辑字符个数指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_edit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_EDIT_SetCursorIndex

【描述】

设置光标在文本内容中的字节索引。

【语法】

```
HI_S32 HI_GV_EDIT_SetCursorIndex(HIGV_HANDLE handle, HI_U32 cursorIndex);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄。	输入
cursorIndex	光标索引。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_edit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_EDIT_GetCursorIndex

【描述】

获取光标在文本内容中的字节索引。

【语法】

```
HI_S32 HI_GV_EDIT_GetCursorIndex(HIGV_HANDLE handle, HI_U32 *cursorIndex);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄。	输入
cursorIndex	光标索引指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_edit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_EDIT_SetCursorPos

【描述】

设置光标位置。

【语法】

```
HI_S32 HI_GV_EDIT_SetCursorPos(HIGV_HANDLE handle, HI_U32 cursorPos);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄。	输入
cursorPos	光标位置。	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_edit.h

- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_EDIT_GetCursorPos

【描述】

获取光标位置。

【语法】

```
HI_S32 HI_GV_EDIT_GetCursorPos(HIGV_HANDLE handle, HI_U32 *cursorPos);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄。	输入
cursorPos	光标位置指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_edit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_EDIT_SetContent

【描述】

设置 Edit 控件当前显示内容。

【语法】

```
HI_S32 HI_GV_EDIT_SetContent(HIGV_HANDLE handle, const HI_CHAR *content);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄。	输入
content	显示内容指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_edit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_EDIT_GetContent

【描述】

获取 Edit 控件当前显示内容。

【语法】

```
HI_S32 HI_GV_EDIT_GetContent(HIGV_HANDLE handle, HI_CHAR *content, HI_U32 length);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄。	输入
content	显示内容信息指针。	输出
length	获取内容的长度。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_edit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_EDIT_SetType

【描述】

设置控件的风格，只能设置一次。

【语法】

```
HI_S32 HI_GV_EDIT_SetType(HIGV_HANDLE handle, HI_U32 type);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄。	输入
type	控件风格，如 HIGV_EDITTYPE_REPLACE	输入

【返回值】

返回值	描述
0	成功
非 0	参见错误码

【需求】

- 头文件：hi_gv_conf.h、hi_gv_edit.h
- 库文件：libhigv.a、libhigv.so

【注意】

当前支持的控件风格类型有 HIGV_EDITTYPE_REPLACE 和 HIGV_EDITTYPE_RDONLY，可参考 hi_gv_edit.h 头文件中相关描述。

【举例】

无。

HI_GV_EDIT_GetType

【描述】

获取控件的风格。

【语法】

```
HI_S32 HI_GV_EDIT_GetType(HIGV_HANDLE handle, HI_U32 *type);
```


【参数】

参数名称	描述	输入/输出
handle	控件句柄。	输入
type	控件风格指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_edit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_EDIT_ShowReplace

【描述】

设置替换风格是否显示内容。

【语法】

```
HI_S32 HI_GV_EDIT_ShowReplace(HIGV_HANDLE handle, HI_BOOL isShow);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄。	输入

参数名称	描述	输入/输出
isShow	显示内容标志位，HI_TRUE 表示显示，HI_FALSE 表示隐藏。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_edit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_EDIT_CleanContent

【描述】

清除编辑框里内容。

【语法】

```
HI_S32 HI_GV_EDIT_CleanContent(HIGV_HANDLE handle);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_edit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.2.16.2 数据类型

相关数据类型、数据结构定义如下：

HigvEditType：单编辑器风格。

HigvEditType

【说明】

输入法类型。

【定义】

```
typedef enum {
    EDIT_TYPE_DEFAULT    = 0x0,
    EDIT_TYPE_REPLACE    = 0x4,
    EDIT_TYPE_RDONLY     = 0x10,
} HigvEditType;
```

【成员】

成员名称	描述
EDIT_TYPE_DEFAULT	默认风格。
EDIT_TYPE_REPLACE	替换风格。
EDIT_TYPE_RDONLY	只读风格。

【注意事项】

无。

【相关数据类型及接口】

无。

2.2.17 MsgBox 控件

2.2.17.1 接口描述

HiGV MsgBox 控件提供 API：

- HI_GV_MSGBOX_Init：初始化消息框。
- HI_GV_MSGBOX_SetTitleText：设置 title 显示文本。
- HI_GV_MSGBOX_GetTitleText：获取 title 文本。
- HI_GV_MSGBOX_SetTitleTextByID：通过字符串 ID 设置 title 文本。
- HI_GV_MSGBOX_GetTitleTextID：获取 title 文本 ID。
- HI_GV_MSGBOX_SetIcon：设置图片资源。
- HI_GV_MSGBOX_ButtonLayout：设置 button 布局。
- HI_GV_MSGBOX_SetInitFocusButton：设置初始焦点。
- HI_GV_MSGBOX_GetButtonHandle：获取子按钮句柄。
- HI_GV_MSGBOX_Show：显示 MsgBox 控件，同步获取按钮索引。

HI_GV_MSGBOX_Init

【描述】

初始化消息框。

【语法】


```
HI_S32 HI_GV_MSGBOX_Init(HIGV_HANDLE handle, const HigvMsgBoxCreateInfo* createInfo);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入
createInfo	初始化 MsgBox 信息指针	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_msgbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_MSGBOX_SetTitleText

【描述】

设置 title 显示文本。

【语法】

```
HI_S32 HI_GV_MSGBOX_SetTitleText(HIGV_HANDLE handle, const HI_CHAR* text);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入
text	标题栏文本内容指针	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_msgbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_MSGBOX_GetTitleText

【描述】

获取 title 文本。

【语法】

```
HI_S32    HI_GV_MSGBOX_GetTitleText(HIGV_HANDLE handle, HI_CHAR* buf, HI_U32 bufLen);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入
buf	输出 buf 缓存指针	输出

参数名称	描述	输入/输出
bufLen	输出内容长度	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_msgbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_MSGBOX_SetTitleTextByID

【描述】

通过字符串 ID 设置 title 文本。

【语法】

```
HI_S32 HI_GV_MSGBOX_SetTitleTextByID(HIGV_HANDLE handle, const HI_U32 strID);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入
strID	控件文本 ID	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_msgbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_MSGBOX_GetTitleTextID

【描述】

获取 title 文本 ID。

【语法】

```
HI_S32   HI_GV_MSGBOX_GetTitleTextID(HIGV_HANDLE handle, HI_U32* strID);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入
strID	标题文本 ID 指针	输出

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_msgbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_MSGBOX_SetIcon

【描述】

设置图片资源。

【语法】

```
HI_S32 HI_GV_MSGBOX_SetIcon(HIGV_HANDLE handle, HI_RESID icon);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入
icon	Icon 的资源 ID	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_msgbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_MSGBOX_ButtonLayout

【描述】

设置 button 布局。

【语法】

```
HI_S32 HI_GV_MSGBOX_ButtonLayout(HIGV_HANDLE handle, const HI_S32 buttonID[MAXBUTTONNUM], HI_S32 length);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入
buttonID	按钮 ID 数组	输入
length	按钮 ID 数组 buttonID 长度	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_msgbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_MSGBOX_SetInitFocusButton

【描述】

设置初始焦点。

【语法】

```
HI_S32 HI_GV_MSGBOX_SetInitFocusButton(HIGV_HANDLE handle, HI_S32 buttonID);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入
buttonID	初始焦点 button	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_msgbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_MSGBOX_GetButtonHandle

【描述】

获取子按钮句柄。

【语法】

```
HI_S32 HI_GV_MSGBOX_GetButtonHandle(HIGV_HANDLE handle, HIGV_HANDLE buttonArray[MAXBUTTONNUM], HI_S32 length);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入
buttonArray	按钮句柄获取参数	输出
length	按钮数组 buttonArray 的长度	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_msgbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_MSGBOX_Show

【描述】

显示 MsgBox 控件，同步获取按钮索引。

【语法】

```
HI_S32 HI_GV_MSGBOX_Show(HIGV_HANDLE handle);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入

【返回值】

返回值	描述
非 0	返回的按钮索引。
MSG_BUTTONID_INVALID（0）	失败。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_msgbox.h
- 库文件：libhigv.a、libhigv.so

【注意】

MsgBox 提供了阻塞式接口调用的功能，当接口 HI_GV_MSGBOX_Show 被调用时程序运行会阻塞在此接口而不继续执行其他处理，UI 界面的操作运行并不会阻塞。当按下 MsgBox 控件中的按钮时触发控件隐藏，此时接口 HI_GV_MSGBOX_Show 返回被点击按钮的 ID，同时停止阻塞并继续执行后续的程序。

【举例】

无。

2.2.17.2 数据类型

相关数据类型、数据结构定义如下：

- HigvMsgBoxStyle：按钮类型风格。
- HigvMsgBoxTitle：MsgBox 标题栏数据结构。
- HigvMsgBoxButton：MsgBox 按钮框数据结构。
- HigvMsgBoxCreateInfo：MsgBox 创建信息结构。

HigvMsgBoxStyle

【说明】

按钮类型风格。

【定义】

```
typedef enum {
    MSGBOX_STYLE_NONE = 0,
    MSGBOX_STYLE_OK = 1,
    MSGBOX_STYLE_OKCANCEL = 2,
    MSGBOX_STYLE_RETRYCANCEL = 3,
    MSGBOX_STYLE_ABORTRETRYIGNORE = 4,
    MSGBOX_STYLE_YESNOCANCEL = 5,
    MSGBOX_STYLE_USERDEF = 6
} HigvMsgBoxStyle;
```

【成员】

成员名称	描述
MSGBOX_STYLE_NONE	None button
MSGBOX_STYLE_OK	Only OK 风格
MSGBOX_STYLE_OKCANCEL	OK+cancel 风格
MSGBOX_STYLE_RETRYCANCEL	Retry+cancel 风格
MSGBOX_STYLE_ABORTRETRYIGNORE	Abort+retry+ignore 风格
MSGBOX_STYLE_YESNOCANCEL	Yes+no+cancel 风格
MSGBOX_STYLE_USERDEF	User-Defined 风格

【注意事项】

无。

【相关数据类型及接口】

无。

HigvMsgBoxTitle

【说明】

MsgBox 标题栏数据结构。

【定义】

```
typedef struct {
    HI_U32    titleHeight;
    HI_U32    titleAlignment;
    HI_RESID titleSkin;
    HI_HANDLE titleFont;
} HigvMsgBoxTitle;
```

【成员】

成员名称	描述
titleHeight	标题栏高度
titleAlignment	标题栏对齐方式
titleSkin	标题栏皮肤类型
titleFont	标题栏字体格式

【注意事项】

无。

【相关数据类型及接口】

无。

HigvMsgBoxButton

【说明】

MsgBox 按钮框数据结构。

【定义】

```
typedef struct {
    HI_U32    buttonAreaHeight;
    HI_U32    buttonCount;
    HI_U32    buttonHeight;
    HI_U32    buttonWidth;
    HI_RESID buttonNormalSkin;
    HI_RESID buttonActiveSkin;
    HI_RESID buttonHighlightSkin;
    HI_RESID buttonMouseDownSkin;
    HI_RESID buttonAreaSkin;
    HI_HANDLE buttonFont;
} HigvMsgBoxButton;
```

【成员】

成员名称	描述
buttonAreaHeight	按钮框区域高度
buttonCount	按钮数量
buttonHeight	按钮高度
buttonWidth	按钮宽度
buttonNormalSkin	按钮普通风格皮肤
buttonActiveSkin	按钮激活风格皮肤
buttonHighlightSkin	按钮高亮风格皮肤
buttonMouseDownSkin	按钮鼠标按下皮肤
buttonAreaSkin	按钮区域皮肤
buttonFont	按钮文本字体风格

【注意事项】

无。

【相关数据类型及接口】

无。

HigvMsgBoxCreateInfo

【说明】

MsgBox 创建信息结构。

【定义】

```
typedef struct {
    HigvMsgBoxStyle msgBoxStyle;
    HigvMsgBoxTitle titleInfo;
    HigvMsgBoxButton buttonInfo;
    HI_U32 showTimes;
    HI_RESID icon;
} HigvMsgBoxCreateInfo;
```

【成员】

成员名称	描述
msgBoxStyle	msgBox 风格类型
titleInfo	标题栏信息
buttonInfo	按钮框区域信息
showTimes	MsgBox 显示时间
icon	MsgBox 标题栏图标

【注意事项】

无。

【相关数据类型及接口】

无。

2.2.18 Charts 控件

2.2.18.1 接口描述

HiGV Charts 控件提供 API：

- HI_GV_CHARTS_Init：初始化图表控件。
- HI_GV_CHARTS_BindLineData：绑定折线各点数据。
- HI_GV_CHARTS_SetLineTitle：设置在图例处显示折线的名称。
- HI_GV_CHARTS_BindBarData：绑定单图例模式下的柱状图数据。
- HI_GV_CHARTS_SetAxisXBarTitle：在单柱状图或条目簇图模式下，设置 X 轴方向上，各柱状图下方的标签内容。
- HI_GV_CHARTS_SetAxisXReplace：设置 X 轴标签替换风格。
- HI_GV_CHARTS_SetAxisXReplaceTitle：X 轴标签替换风格下，X 轴标签内容。

HI_GV_CHARTS_Init

【描述】

初始化图表控件。

【语法】

```
HI_S32 HI_GV_CHARTS_Init(HIGV_HANDLE handle, const HIGV_CHARTS_STYLE *style,
HIGV_CHARTS_BAR_S *barAttr, HI_U32 barCnt);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入
style	初始化图表信息结构体指针	输入
barAttr	初始化条目簇类型时，子条目属性指针	输入
barCnt	子条目数量	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_charts.h
- 库文件：libhigv.a、libhigv.so

【注意】

目前不支持折线图规格。

【举例】

无。

HI_GV_CHARTS_BindLineData

【描述】

绑定折线各点数据。

【语法】

```
HI_S32 HI_GV_CHARTS_BindLineData(HIGV_HANDLE handle, HI_U32 pos, HI_COLOR color,
const HIGV_LINECHARTS_POINT_S *lineChartPoint, HI_U32 length);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入
pos	第几条折线，序号从 0 开始	输入
color	绘制时，折线的颜色	输入
lineChartPoint	一条折线上点集合的指针	输入
length	折线上共多少点	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_charts.h
- 库文件：libhigv.a、libhigv.so

【注意】

目前不支持折线图规格。

【举例】

请见 HI_GV_CHARTS_SetLineTitle 举例。

HI_GV_CHARTS_SetLineTitle

【描述】

设置用于图例显示的折线标题。

【语法】

```
HI_S32 HI_GV_CHARTS_SetLineTitle(HIGV_HANDLE handle, const HI_CHAR *name, HI_U32 length, HI_U32 pos);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入
name	传入的内容指针	输入
length	传入字符串长度	输入
pos	对应的第几条折线的标题	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_charts.h
- 库文件：libhigv.a、libhigv.so

【注意】

在调用 HI_GV_Widget_Show 前，需调用此接口将折线标题与图表控件绑定，否则图例显示不正常。

【举例】

```
/* 折线图示例数据 */
const static HI_U32 POINTCNT = 6;
static HIGV_LINECHARTS_POINT_S pointSave [POINTCNT] = {
    {15, 23}, {30, 40}, {50, 90},
    {75, 2}, {86, 40}, {92, 80}
    };
/* 绑定折线数据 */
(HI_VOID)HI_GV_CHARTS_BindLineData(chartHandle, 0, 0xFFFF0000, pointSave, POINTCNT);
/* 设置用于图例显示的折线标题 */
(HI_VOID)HI_GV_CHARTS_SetLineTitle(chartHandle, "Line_Sam", 9, 0);
HI_GV_PARSER_LoadViewById(view);
HI_GV_Widget_Show(view);
```

HI_GV_CHARTS_BindBarData

【描述】

设置绑定单柱状图数据。

【语法】

```
HI_S32 HI_GV_CHARTS_BindBarData(HIGV_HANDLE handleBarCharts, const
```



```
HIGV_BARCHARTS_VALUE_S *barChartValue, HI_U32 length);
```

【参数】

参数名称	描述	输入/输出
handleBarCharts	控件句柄	输入
barChartValue	传入的柱状图结构体指针	输入
length	传入柱状图数量	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_charts.h
- 库文件：libhigv.a、libhigv.so

【注意】

在调用 HI_GV_Widget_Show 前，需调用此接口将单柱状图与图表控件绑定，否则图表无法显示柱状图；条目簇类型的图表不需要调用此接口。

【举例】

请见 HI_GV_CHARTS_SetAxisXBarTitle 举例。

HI_GV_CHARTS_SetAxisXBarTitle

【描述】

设置柱状图、条目簇图的 X 轴方向标签内容。

【语法】

```
HI_S32 HI_GV_CHARTS_SetAxisXBarTitle(HIGV_HANDLE handle, const HI_CHAR *name, HI_U32 length, HI_U32 pos);
```


【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入
name	传入的标签内容指针	输入
length	字符串长度	输入
pos	对应第 pos 位的柱状图，pos 序号从 0 开始	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_charts.h
- 库文件：libhigv.a、libhigv.so

【注意】

在调用 HI_GV_Widget_Show 前，需调用此接口将柱状图（条目簇图）与图表控件绑定，否则柱状（条目簇）图表 x 轴下方标签无法正常显示。

【举例】

```
/* 柱状图 示例数据 */
const static HI_U32 TICKNUM = 5;
const static HI_U32 MAXLEN = 5;
/* 柱状图 数据 */
static HIGV_BARCHARTS_VALUE_S barSave[TICKNUM] = {
    {80, 0xffff0000}, {40, 0xFFFF8000}, {68, 0xFFFF00FF},
    {95, 0xFF00FF00}, {20,0xFF804000}
    };
/* X轴坐标上各柱状图对应的标签数据 */
```

```
static char* barLabel[MAXLEN] = {"Data1", "Data2", "Data3", "Data4", "Data5"};
/* 绑定柱状图数据 */
(HI_VOID)HI_GV_CHARTS_BindBarData(barHandle,barSave, TICKNUM);
/* 绑定X轴坐标上各柱状图对应的标签数据 */
for (HI_U32 i = 0; i < MAXLEN; i++) {
    (HI_VOID)HI_GV_CHARTS_SetAxisXBarTitle(barHandle,barLabel[i], strlen(barLabel[i]), i);
}
HI_GV_PARSER_LoadViewById(view);
HI_GV_Widget_Show(view);
```

HI_GV_CHARTS_SetAxisXReplace

【描述】

设置 X 轴标签替换风格。

【语法】

```
HI_S32 HI_GV_CHARTS_SetAxisXReplace(HIGV_HANDLE handle, HI_BOOL replace);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入
replace	设置是否为替换风格	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_charts.h
- 库文件：libhigv.a、libhigv.so

【注意】

仅在折线图类型中使用。

【举例】

无

HI_GV_CHARTS_SetAxisXReplaceTitle

【描述】

X 轴标签替换风格下，X 轴标签内容。

【语法】

```
HI_S32 HI_GV_CHARTS_SetAxisXReplaceTitle(HIGV_HANDLE handle, const HI_CHAR *name, HI_U32 length, HI_U32 pos);
```

【参数】

参数名称	描述	输入/输出
handle	控件句柄	输入
name	传入字符串指针	输入
length	传入字符串长度	输入
pos	替换 X 轴刻度标签的位置	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_charts.h
- 库文件：libhigv.a、libhigv.so

【注意】

- 仅在折线图类型中使用；
- 使用此接口时，需先调用 HI_GV_CHARTS_SetAxisXReplace 接口，或在布局文件(*.xml)中将 isreplacexlabel 属性设置为”yes”。

【举例】

无

2.2.18.2 数据类型

相关数据类型、数据结构定义如下：

- HIGV_CHARTS_STYLE：图表控件结构体。
- HIGV_CHARTS_COMM_S：图表控件公有属性结构体。
- HIGV_CHARTS_AXIS_S：图表控件坐标轴属性结构体。
- HIGV_CHARTS_STLINE_S：图表控件标准线结构体。
- HIGV_CHARTS_BAR_S：图表控件控件中子条目簇结构体。
- HIGV_LINECHARTS_POINT_S：图表控件折线点坐标结构体。
- HIGV_BARCHARTS_VALUE_S：图表控件单柱状图模式下柱状图数据结构体。
- MAX_LINE_LABEL_TEXT：数组大小长度。

HIGV_CHARTS_STYLE

【说明】

图表控件结构体。

【定义】

```
typedef struct {
    HIGV_CHARTS_COMM_S comAttr;
    HIGV_CHARTS_AXIS_S axisAttr;
    HIGV_CHARTS_STLINE_S stLineAttr;
    HI_BOOL isDrawStLine;
    HI_BOOL gridLine;
    HI_COLOR gridColor;
    HI_BOOL textOnBar;
    HIGV_HANDLE textOnBarFont;
    HI_COLOR textOnBarColor;
```



```
    HI_U32 axisXIndentation;
    HI_U32 axisYIndentation;
    HI_BOOL isReplaceXLabel;
} HIGV_CHARTS_STYLE;
```

【成员】

成员名称	描述
comAttr	控件公有属性结构体
axisAttr	控件坐标轴属性结构体
stLineAttr	控件标准线属性结构体
isDrawStLine	是否绘制标准线
gridLine	是否绘制网格线
gridColor	网格线颜色
textOnBar	是否在柱状图上方绘制相应数值
textOnBarFont	柱状图上方绘制相应数值的字体句柄
textOnBarColor	柱状图上方绘制相应数值的字体颜色
axisXIndentation	X 轴方向的缩进大小，用于显示字体等
axisYIndentation	Y 轴方向的缩进大小，用于显示字体等
isReplaceXLabel	用于代替 X 轴上刻度值，比如用日期显示等

【注意事项】

无。

【相关数据类型及接口】

HI_GV_CHARTS_Init

HIGV_CHARTS_COMM_S

【说明】

图表控件公有属性结构体。

【定义】

```
typedef struct {
    HI_U32 chartType;
    HI_U32 barWidth;
    HI_U32 subBarWidth;
    HI_U32 subBarDist;
    HI_U32 subBarCnt;
    HI_U32 lineCnt;
    HI_U32 lineWidth;
    HI_COLOR borderColor;
    HI_U32 borderWidth;
    HIGV_HANDLE titleFont;
    HI_COLOR titleColor;
    HIGV_HANDLE legendFont;
    HI_COLOR legendColor;
    HI_BOOL drawBorder;
    HI_CHAR   titleContent[MAX_LINE_LABEL_TEXT];
} HIGV_CHARTS_COMM_S
```

【成员】

成员名称	描述
chartType	控件类型
barWidth	单柱状图下，单个柱状宽度
subBarWidth	条目簇模式下，单个子柱状宽度
subBarDist	条目簇模式下，子柱状之间的间距
subBarCnt	条目簇模式下，子柱状个数
lineCnt	折线个数
lineWidth	折线宽度
borderColor	边框颜色
borderWidth	边框宽度
titleFont	控件标题字体句柄

成员名称	描述
titleColor	控件标题字体颜色
legendFont	控件图例字体句柄
legendColor	控件图例字体颜色
drawBorder	是否绘制边框
titleContent	控件标题内容

【注意事项】

无。

【相关数据类型及接口】

HI_GV_CHARTS_Init

HIGV_CHARTS_AXIS_S

【说明】

图表控件坐标轴属性结构体。

【定义】

```
typedef struct {
    HI_U32 axisLeftSpace;
    HI_U32 axisRightSpace;
    HI_U32 axisTopSpace;
    HI_U32 axisBottonSpace;
    HI_U32 axisWidth;
    HI_COLOR axisColor;
    HI_U32 axisArrowLength;
    HIGV_HANDLE axisLabelFont;
    HI_COLOR axisLabelColor;
    HI_U32 axisXMinValue;
    HI_U32 axisXMaxValue;
    HI_U32 axisYMinValue;
    HI_U32 axisYMaxValue;
    HI_U32 axisXTickNum;
```



```
    HI_U32 axisYTickNum;
    HI_U32 axisTickLength;
    HI_CHAR axisXLabelContent[MAX_LINE_LABEL_TEXT];
    HI_CHAR axisYLabelContent[MAX_LINE_LABEL_TEXT];
} HIGV_CHARTS_AXIS_S;
```

【成员】

成员名称	描述
axisLeftSpace	坐标轴与边框的左边距
axisRightSpace	坐标轴与边框的右边距
axisTopSpace	坐坐标轴与边框的上边距
axisBottonSpace	坐标轴与边框的下边距
axisWidth	坐标轴宽度
axisColor	坐标轴颜色
axisArrowLength	坐标轴箭头长度
axisLabelFont	坐标轴标签字体句柄
axisLabelColor	坐标轴标签颜色
axisXMinValue	X 轴取值最小值
axisXMaxValue	X 轴取值最大值
axisYMinValue	Y 轴取值最小值
axisYMaxValue	Y 轴取值最大值
axisXTickNum	X 轴等分数
axisYTickNum	Y 轴等分数
axisTickLength	坐标轴刻度长度
axisXLabelContent	X 轴坐标标签内容
axisYLabelContent	Y 轴坐标标签内容

【注意事项】

无。

【相关数据类型及接口】

HI_GV_CHARTS_Init

HIGV_CHARTS_STLINE_S

【说明】

图表控件标准线结构体。

【定义】

```
typedef struct {
    HI_U32 stLineMinValue;
    HI_COLOR stLineMinColor;
    HI_U32 stLineMaxValue;
    HI_COLOR stLineMaxColor;
} HIGV_CHARTS_STLINE_S;
```

【成员】

成员名称	描述
stLineMinValue	下标准线值
stLineMinColor	下标准线颜色
stLineMaxValue	上标准线值
stLineMaxColor	上标准线颜色

【注意事项】

无。

【相关数据类型及接口】

HI_GV_CHARTS_Init

HIGV_CHARTS_BAR_S

【说明】

图表控件控件中子条目簇结构体。

【定义】

```
typedef struct  {
    HI_COLOR barColor;
    HI_U32 subBarTitle;
    HI_U32 indexDB;
} HIGV_CHARTS_BAR_S;
```

【成员】

成员名称	描述
barColor	子柱状图填充颜色
subBarTitle	子柱状图标题，用于在图例处显示
indexDB	对应数据模型中的列号

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_LINECHARTS_POINT_S

【说明】

图表控件折线点坐标结构体。

【定义】

```
typedef struct {
    HI_U32 x;
    HI_U32 y;
} HIGV_LINECHARTS_POINT_S;
```

【成员】

成员名称	描述
x	X 轴坐标点
y	Y 轴坐标点

【注意事项】

无。

【相关数据类型及接口】

无。

HIGV_BARCHARTS_VALUE_S

【说明】

图表控件单柱状图模式下柱状图数据结构体。

【定义】

```
typedef struct {
    HI_U32 value;
    HI_COLOR color;
} HIGV_BARCHARTS_VALUE_S;
```

【成员】

成员名称	描述
value	对应纵坐标的数据
color	对应的颜色值

【注意事项】

无。

【相关数据类型及接口】

无。

MAX_LINE_LABEL_TEXT

【说明】

数组大小长度。

【定义】

```
#define   MAX_LINE_LABEL_TEXT       128
```

【注意事项】

无。

【相关数据类型及接口】

无。

2.2.19 MultiEdit 控件

2.2.19.1 接口描述

HiGV MultiEdit 控件提供 API：

- HI_GV_MULTIEDIT_Init：初始化 multiedit 控件。
- HI_GV_MULTIEDIT_SetContent：设置 Edit 控件当前显示内容。
- HI_GV_MULTIEDIT_GetContent：获取 Edit 控件当前显示内容。
- HI_GV_MULTIEDIT_ClearContent：清除 Edit 控件当前显示内容。
- HI_GV_MULTIEDIT_GetSelectString：获取选中的字串。
- HI_GV_MULTIEDIT_PageUp：向上翻页。
- HI_GV_MULTIEDIT_PageDown：向下翻页。
- HI_GV_MULTIEDIT_SetCursorColor：设置光标颜色。

HI_GV_MULTIEDIT_Init

【描述】

初始化 multiedit 控件。

【语法】

```
HI_S32 HI_GV_MULTIEDIT_Init(HIGV_HANDLE editHandle, const HigvMultiEditStyle *style);
```

【参数】

参数名称	描述	输入/输出
editHandle	控件句柄。	输入
style	控件创建信息指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_multiedit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_MULTIEDIT_SetContent

【描述】

设置 Edit 控件当前显示内容。

【语法】

```
HI_S32 HI_GV_MULTIEDIT_SetContent(HIGV_HANDLE editHandle, const HI_CHAR *content);
```

【参数】

参数名称	描述	输入/输出
editHandle	控件句柄。	输入
content	当前显示内容指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_multiedit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_MULTIEDIT_GetContent

【描述】

获取 Edit 控件当前显示内容。

【语法】

```
HI_S32 HI_GV_MULTIEDIT_GetContent(HIGV_HANDLE editHandle, HI_CHAR *content, HI_32 *length);
```

【参数】

参数名称	描述	输入/输出
editHandle	控件句柄。	输入
content	当前显示内容指针。	输出

参数名称	描述	输入/输出
length	• 当 content 不为空指针时，作为输入参数，表示要获取内容长度； • 当 content 传入空指针时，作为输出参数，表示实际内容长度。	输入/输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_multiedit.h
- 库文件：libhigv.a、libhigv.so

【注意】

此接口一般先获取内容长度，再申请内存获取显示内容，否则 buf 长度小于实际内容长度时，获取内容会被截断。

【举例】

无。

HI_GV_MULTIEDIT_ClearContent

【描述】

清除 Edit 控件当前显示内容。

【语法】

```
HI_S32 HI_GV_MULTIEDIT_ClearContent(HIGV_HANDLE editHandle);
```

【参数】

参数名称	描述	输入/输出
editHandle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_multiedit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_MULTIEDIT_GetSelectString

【描述】

获取选中的字串。

【语法】

```
HI_S32 HI_GV_MULTIEDIT_GetSelectString(HIGV_HANDLE editHandle, HI_CHAR *str, HI_U32 *len);
```

【参数】

参数名称	描述	输入/输出
editHandle	控件句柄。	输入
str	输出的 UTF-8 字符串指针。	输出

参数名称	描述	输入/输出
len	• 当 str 不为空指针时，作为输入参数，表示要获取得字串长度； • 当 strt 传入空指针时，作为输出参数，表示实际字串长度。	输入/输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_multiedit.h
- 库文件：libhigv.a、libhigv.so

【注意】

此接口一般先获取内容长度，再申请内存获取显示内容，否则 buf 长度小于实际内容长度时，获取内容会被截断。

【举例】

无。

HI_GV_MULTIEDIT_PageUp

【描述】

向上翻页。

【语法】

```
HI_S32 HI_GV_MULTIEDIT_PageUp(HIGV_HANDLE editHandle);
```

【参数】

参数名称	描述	输入/输出
editHandle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_multiedit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_MULTIEDIT_PageDown

【描述】

向下翻页。

【语法】

```
HI_S32 HI_GV_MULTIEDIT_PageDown(HIGV_HANDLE editHandle);
```

【参数】

参数名称	描述	输入/输出
editHandle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_multiedit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_MULTIEDIT_SetCursorColor

【描述】

设置光标颜色。

【语法】

```
HI_S32 HI_GV_MULTIEDIT_SetCursorColor(HIGV_HANDLE editHandle, HI_COLOR color);
```

【参数】

参数名称	描述	输入/输出
editHandle	控件句柄。	输入
color	光标颜色值。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_multiedit.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.2.19.2 数据类型

相关数据类型、数据结构定义如下：

- HigvMultiEditStyle：多行编辑器属性。
- HigvMultiEditSlide：多行编辑器边界。

HigvMultiEditStyle

【说明】

多行编辑器属性。

【定义】

```
t typedef struct {
    HI_U32 charNum;
    HI_BOOL isReadOnly;
    HI_BOOL isSingleLine;
    HIGV_HANDLE font;
    HI_COLOR selectBgColor;
    HI_COLOR selectFgColor;
    HI_COLOR cursorColor;
    HI_U32 lineSpace;
    HI_U32 charSpace;
} HigvMultiEditStyle;
```

【成员】

成员名称	描述
charNum	字符限制，为零不做限制
isReadOnly	是否只读
isSingleLine	是否单行。 自动换行会根据很多因素多判断，不同种类的语言，符号，空格等等
font	字体句柄
selectBgColor	反选背景颜色
selectFgColor	反选前景颜色
cursorColor	光标颜色
lineSpace	行间距
charSpace	字符间距

【注意事项】

无。

【相关数据类型及接口】

无。

HigvMultiEditSlide

【说明】

多行编辑器边界。

【定义】

```
typedef enum {
    HIGV_MULTIEDIT_SIDE_LEFT = 0,
    HIGV_MULTIEDIT_SIDE_RIGHT = 1,
    HIGV_MULTIEDIT_SIDE_TOP = 2,
    HIGV_MULTIEDIT_SIDE_BOTTOM = 3,
} HigvMultiEditSlide;
```


【成员】

成员名称	描述
HIGV_MULTIEDIT_SIDE_LEFT	左边界
HIGV_MULTIEDIT_SIDE_RIGHT	右边界
HIGV_MULTIEDIT_SIDE_TOP	上边界
HIGV_MULTIEDIT_SIDE_BOTTOM	下边界

【注意事项】

无。

【相关数据类型及接口】

无。

2.2.20 ScrollText 控件

2.2.20.1 接口描述

HiGV ScrollText 控件提供 API：

- HI_GV_ScrollText_SetContent：设置滚动字幕控件中的内容。
- HI_GV_ScrollText_SetContentByID：根据资源 ID 设置滚动字幕控件中的内容。
- HI_GV_ScrollText_GetContent：获取滚动字幕控件的内容。
- HI_GV_ScrollText_SetImagePos：设置图片的起始位置。
- HI_GV_ScrollText_SetTextPos：设置文本的起始位置。
- HI_GV_ScrollText_SetTimeInterval：设置控件滚动的时间间隔。
- HI_GV_ScrollText_SetStep：设置控件滚动步长。
- HI_GV_ScrollText_SetDirection：设置控件滚动方向。
- HI_GV_ScrollText_SetStatus：设置控件状态。
- HI_GV_ScrollText_GetCurPos：获取控件内容当前所在的位置。
- HI_GV_ScrollText_ResetPos：复位控件内容的位置。

HI_GV_ScrollText_SetContent

【描述】

设置滚动字幕控件中的内容。

【语法】

```
HI_S32 HI_GV_ScrollText_SetContent(HIGV_HANDLE scrollTextHandle, HI_RESID image, const HI_CHAR *string);
```

【参数】

参数名称	描述	输入/输出
scrollTextHandle	控件句柄。	输入
image	图片资源 ID，INVALID_RESID 表示无图片。	输入
string	文本字符串	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_scrolltext.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollText_SetContentByID

【描述】

根据资源 ID 设置滚动字幕控件中的内容。

【语法】

```
HI_S32 HI_GV_ScrollText_SetContentByID(HIGV_HANDLE scrollTextHandle, HI_RESID image,
HI_U32 strID);
```

【参数】

参数名称	描述	输入/输出
scrollTextHandle	控件句柄。	输入
image	图片资源 ID，INVALID_RESID 表示无图片。	输入
strID	文本字符串 ID	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_scrolltext.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollText_GetContent

【描述】

获取滚动字幕控件的内容。

【语法】

```
HI_S32 HI_GV_ScrollText_GetContent(HIGV_HANDLE scrollTextHandle, HI_RESID *image,
HI_CHAR *, HI_U32 textBufLen);
```

【参数】

参数名称	描述	输入/输出
scrollTextHandle	控件句柄。	输入
image	图片资源 ID，INVALID_RESID 表示无图片。	输出
string	文本字符串。	输出
textBufLen	保存文本字串的 buf 长度。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_scrolltext.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollText_SetImagePos

【描述】

设置图片的起始位置。

【语法】

```
HI_S32 HI_GV_ScrollText_SetImagePos(HIGV_HANDLE scrollTextHandle, HI_S32 x, HI_S32 y);
```

【参数】

参数名称	描述	输入/输出
scrollTextHandle	控件句柄。	输入
x	图片在控件中的 x 坐标(相对于控件原点，默认相对控件居中)。	输入
y	图片在控件中的 y 坐标(相对于控件原点，默认相对控件居中)	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_scrolltext.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollText_SetTextPos

【描述】

设置文本的起始位置。

【语法】

```
HI_S32 HI_GV_ScrollText_SetTextPos(HIGV_HANDLE scrollTextHandle, HI_S32 x, HI_S32 y);
```

【参数】

参数名称	描述	输入/输出
scrollTextHandle	控件句柄。	输入
x	文本在控件中的 x 坐标(相对于控件原点，默认相对控件居中)。	输入
y	文本在控件中的 y 坐标(相对于控件原点，默认相对控件居中)	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_scrolltext.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollText_SetTimeInterval

【描述】

设置控件滚动的时间间隔。

【语法】

```
HI_S32 HI_GV_ScrollText_SetTimeInterval(HIGV_HANDLE scrollTextHandle, HI_U32 timeInterval);
```

【参数】

参数名称	描述	输入/输出
scrollTextHandle	控件句柄。	输入
timeInterval	时间间隔(以 ms 为单位，初始值为 100ms)。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_scrolltext.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollText_SetStep

【描述】

设置控件滚动步长。

【语法】

```
HI_S32 (HIGV_HANDLE scrollTextHandle, HI_U32 step);
```

【参数】

参数名称	描述	输入/输出
scrollTextHandle	控件句柄。	输入
step	滚动步长(以 pixel 为单位，默认为 5pixel)。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_scrolltext.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollText_SetDirection

【描述】

设置控件滚动方向。

【语法】

```
HI_S32 HI_GV_ScrollText_SetDirection(HIGV_HANDLE scrollTextHandle, HIGV_DIRECTION_E direction);
```


【参数】

参数名称	描述	输入/输出
scrollTextHandle	控件句柄。	输入
direction	滚动方向(默认自右向左)。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_scrolltext.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollText_SetStatus

【描述】

设置控件状态。

【语法】

```
HI_S32 HI_GV_ScrollText_SetStatus(HIGV_HANDLE scrollTextHandle, HI_BOOL status);
```

【参数】

参数名称	描述	输入/输出
scrollTextHandle	控件句柄。	输入

参数名称	描述	输入/输出
status	控件状态(TRUE 为滚动，FALSE 为停止滚动，默认为 TRUE)	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_scrolltext.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollText_GetCurPos

【描述】

获取控件内容当前所在的位置。

【语法】

```
HI_S32 HI_GV_ScrollText_GetCurPos(HIGV_HANDLE scrollTextHandle, HI_S32 *x, HI_S32 *y);
```

【参数】

参数名称	描述	输入/输出
scrollTextHandle	控件句柄。	输入
x	X 坐标	输出

参数名称	描述	输入/输出
y	Y 坐标	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_scrolltext.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_ScrollText_ResetPos

【描述】

复位控件内容的位置。

【语法】

```
HI_S32 HI_GV_ScrollText_ResetPos(HIGV_HANDLE scrollTextHandle);
```

【参数】

参数名称	描述	输入/输出
scrollTextHandle	控件句柄。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_gv_conf.h、hi_gv_scrolltext.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

2.2.20.2 数据类型

相关数据类型、数据结构定义如下：

HIGV_DIRECTION_E：滚动方向。

HIGV_DIRECTION_E

【说明】

滚动方向。

【定义】

```
typedef enum {
    HIGV_DIRECTION_LEFT = 0,
    HIGV_DIRECTION_RIGHT,
    HIGV_DIRECTION_BUTT
} HIGV_DIRECTION_E;
```

【成员】

成员名称	描述
HIGV_DIRECTION_LEFT	自右向左滚动。

成员名称	描述
HIGV_DIRECTION_RIGHT	自左向右滚动。

【注意事项】

无。

【相关数据类型及接口】

无。

2.3 XML 解析模块

2.3.1.1 接口描述

HiGV XML 解析模块提供以下 API：

- HI_GV_PARSER_Init：Parser 模块初始化。
- HI_GV_PARSER_Deinit：Parser 模块去初始化
- HI_GV_PARSER_LoadFile：HIGVBin 文件加载，同时解析出 HIGVBin 文件基本信息。
- HI_GV_PARSER_UnLoadFile：卸载 HIGVBin 文件，释放资源。
- HI_GV_PARSER_ReleaseLoadRes：释放文件加载过程中申请的资源，在所有视图加载完成后调用。
- HI_GV_PARSER_GetViewNum：获取视图总数。
- HI_GV_PARSER_LoadViewByName：按视图名加载视图，加载时解析并创建视图。
- HI_GV_PARSER_LoadViewById：按视图 ID 加载视图，加载时解析并创建视图。
- HI_GV_PARSER_UnloadViewByID：按视图 ID 卸载视图，销毁视图控件。
- HI_GV_PARSER_LoadViewByIndex：按视图索引加载视图，加载时解析并创建视图，这个函数线程安全。
- HI_GV_PARSER_UnLoadViewExclude：删除除指定的视图外其他所有视图。
- HI_GV_PARSER_ViewGetWinsHandle：通过视图 ID 获取视图内所有窗口句柄。
- HI_GV_PARSER_GetVersion：获取版本号。

- HI_GV_PARSER_SetWidgetEventFunc：设置 STC 模式下事件函数列表。

HI_GV_PARSER_Init

【描述】

Parser 模块初始化。

【语法】

```
HI_S32 HI_GV_PARSER_Init(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_go.h、hi_gv.h、hi_gv_parser.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

参考第 4 章 “开发指引”。

HI_GV_PARSER_Deinit

【描述】

Parser 模块去初始化。

【语法】

```
HI_VOID HI_GV_PARSER_Deinit(HI_VOID);
```


【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_go.h、hi_gv.h、hi_gv_parser.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_PARSER_LoadFile

【描述】

HIGVBin 文件加载，同时解析出 HIGVBin 文件基本信息。

【语法】

```
HI_S32 HI_GV_PARSER_LoadFile(const HI_CHAR* fileName);
```

【参数】

参数名称	描述	输入/输出
fileName	HIGVBin 文件名。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_go.h、hi_gv.h、hi_gv_parser.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_PARSER_UnLoadFile

【描述】

卸载 HIGVBin 文件，释放资源。

【语法】

```
HI_S32 HI_GV_PARSER_UnLoadFile(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_go.h、hi_gv.h、hi_gv_parser.h

- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_PARSER_ReleaseLoadRes

【描述】

释放文件加载过程中申请的资源，在所有视图加载完成后调用。

【语法】

```
HI_S32 HI_GV_PARSER_ReleaseLoadRes(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_go.h、hi_gv.h、hi_gv_parser.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_PARSER_GetViewNum

【描述】

获取视图总数。

【语法】

```
HI_S32 HI_GV_PARSER_GetViewNum(HI_U32*    viewNum);
```

【参数】

参数名称	描述	输入/输出
viewNum	视图总数指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_go.h、hi_gv.h、hi_gv_parser.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_PARSER_LoadViewByName

【描述】

按视图名加载视图，加载时解析并创建视图。

【语法】


```
HI_S32 HI_GV_PARSER_LoadViewByName(const HI_CHAR* viewName);
```

【参数】

参数名称	描述	输入/输出
viewName	视图名指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_go.h、hi_gv.h、hi_gv_parser.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_PARSER_LoadViewById

【描述】

按视图 ID 加载视图，加载时解析并创建视图。

【语法】

```
HI_S32 HI_GV_PARSER_LoadViewById(HIGV_HANDLE viewId);
```

【参数】

参数名称	描述	输入/输出
viewId	视图 ID。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_go.h、hi_gv.h、hi_gv_parser.h
- 库文件：libhigv.a、libhigv.so

【注意】

如果 ViewId 为 LOADALLVIEW，则加载所有视图。

【举例】

参考第 4 章 “开发指引”。

HI_GV_PARSER_UnloadViewByID

【描述】

按视图 ID 卸载视图，销毁视图控件。

【语法】

```
HI_S32 HI_GV_PARSER_UnloadViewByID(HIGV_HANDLE viewId);
```

【参数】

参数名称	描述	输入/输出
viewId	视图 ID。	输入

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_go.h、hi_gv.h、hi_gv_parser.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_PARSER_LoadViewByIndex

【描述】

按视图索引加载视图，加载时解析并创建视图，这个函数线程安全。

【语法】

```
HI_S32 HI_GV_PARSER_LoadViewByIndex(HI_U32 viewIndex);
```

【参数】

参数名称	描述	输入/输出
viewIndex	视图索引。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_go.h、hi_gv.h、hi_gv_parser.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_PARSER_UnLoadViewExclude

【描述】

删除除指定的视图外其他所有视图。

【语法】

```
HI_S32 HI_GV_PARSER_UnLoadViewExclude(const HIGV_HANDLE* excludeViewId, HI_U32 excludeViewNum);
```

【参数】

参数名称	描述	输入/输出
excludeViewId	不需要删除视图 ID 数组指针。	输入
excludeViewNum	不需要删除视图数目。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_go.h、hi_gv.h、hi_gv_parser.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_PARSER_ViewGetWinsHandle

【描述】

通过视图 ID 获取视图内所有窗口句柄。

【语法】

```
HI_S32 HI_GV_PARSER_ViewGetWinsHandle(HIGV_HANDLE viewId, HIGV_HANDLE** winsArray, HI_U32* winsNum);
```

【参数】

参数名称	描述	输入/输出
viewId	视图 ID。	输入
winsArray	视图内窗口 handle 数组指针。	输出
winsNum	视图内窗口数目指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_go.h、hi_gv.h、hi_gv_parser.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_PARSER_GetVersion

【描述】

获取版本号。

【语法】

```
HI_S32 HI_GV_PARSER_GetVersion(HI_U32* mainVersion, HI_U32*subVersion);
```

【参数】

参数名称	描述	输入/输出
mainVersion	主版本号。	输出
subVersion	子版本号。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_go.h、hi_gv.h、hi_gv_parser.h
- 库文件：libhigv.a、libhigv.so

【注意】

无。

【举例】

无。

HI_GV_PARSER_SetWidgetEventFunc

【描述】

设置 STC 模式下事件函数列表。

【语法】

```
HI_S32 HI_GV_PARSER_SetWidgetEventFunc(const HIGV_MSG_PROC* eventProc, HI_U32 number);
```

【参数】

参数名称	描述	输入/输出
eventProc	事件函数数组指针。	输入
number	事件函数数组中函数个数。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_type.h、hi_go.h、hi_gv.h、hi_gv_parser.h
- 库文件：libhigv.a、libhigv.so

【注意】

HIGV_MSG_PROC 为回调函数，定义为 `typedef HI_S32 (*HIGV_MSG_PROC)(HIGV_HANDLE widgetHandle,HI_U32 wParam, HI_U32 lParam)`。

【举例】

无。

2.3.1.2 数据类型

无。

3 错误码

HiGV 提供的错误码如表 3-1 所示。

表3-1 HiGV 模块的错误码

错误代码	宏定义	描述
-1	HI_FAILURE	失败。
0	HI_SUCCESS	成功。
1	HI_ERR_COMM_UNKNOWN	错误原因未知，确定已经出错但是不能判定错误的原因。
2	HI_ERR_COMM_OTHER	其他错误，知道错误原因但是难以归类。
3	HI_ERR_COMM_INTER	内部错误，例如内部发生断言错误、某些内存崩溃、CPU 自测错、I/O 错、数学运算错、死锁等。
4	HI_ERR_COMM_VERSION	版本错误。
5	HI_ERR_COMM_PAERM	不支持的操作/功能/特性。
6	HI_ERR_COMM_INVAL	参数错误。
7	HI_ERR_COMM_NOTINIT	没有初始化。

错误代码	宏定义	描述
8	HI_ERR_COMM_NOTREADY	没有准备好，某些操作必须在具备一定条件后才能进行，或者需要的环境不正确，或者缺乏必需的资源。
9	HI_ERR_COMM_NORES	没有资源，例如申请内存失败、没有空闲缓冲区、没有空闲端口、没有空闲通道等。
10	HI_ERR_COMM_EXIST	资源已存在，欲申请或新建的资源已经存在。
11	HI_ERR_COMM_LOST	资源不存在，依赖的某种资源、地址、会话不存在。
12	HI_ERR_COMM_NOOP	资源不可操作，可能已损坏不可使用、校验错误、未发挥预期的作用、设备不兼容等。
13	HI_ERR_COMM_BUSY	资源正忙碌，例如被加锁。
14	HI_ERR_COMM_IDLE	资源正空闲。
15	HI_ERR_COMM_FULL	满，某种容器中已经被充满。
16	HI_ERR_COMM_EMPTY	空，某种容器中是空的。
17	HI_ERR_COMM_UNDERFLOW	下溢，某种容器中的容量已经下降到下溢水线之下。
18	HI_ERR_COMM_OVERFLOW	上溢，某种容器中的容量上升到上溢水线之上。
19	HI_ERR_COMM_ACCES	权限错误，例如没有权限、密码错误等。
20	HI_ERR_COMM_INTR	操作未完成，已经中断。
21	HI_ERR_COMM_CONTINUE	操作未完成，仍在继续。
22	HI_ERR_COMM_OVER	操作完成，已没有后续的操作对象。

错误代码	宏定义	描述
23	HI_ERR_COMM_UNSUPPORT	不支持的操作。
24	HI_ERR_COMM_OUTOFBOUND	超出边界。
25	HI_ERR_COMM_FILEOP	文件操作错误。
26	HI_ERR_COMM_SECURE	安全函数使用错误。
27	HI_ERR_COMM_NOEFFECT	对业务无影响返回值，有别于HI_SUCCESS。
500	HI_ERR_MSGM_VTOPMSGSEND	发送同步 VTOP 消息错误。
501	HI_ERR_MSGM_VTOPMSGCREATE	创建 VTOP 消息队列失败。
502	HI_ERR_MSGM_VTOPMSGSERVER	创建 VTOP 消息服务器失败。
503	HI_ERR_TASK_TASKNOTSTOP	任务未停止。
504	HI_ERR_RES_NOMEN	内存不足。
505	HI_ERR_RES_UNKNOWRES	未知的资源类型。
506	HI_ERR_RES_USED	资源正被使用。
507	HI_ERR_RES_INVRESTYPE	无效的资源类型。
508	HI_ERR_RES_NULL	保存资源信息的指针为空。
509	HI_ERR_RES_LOAD	资源加载失败。
510	HI_ERR_RES_NOLOAD	资源未加载。
511	HI_ERR_RES_CREATE	资源创建失败，一般指调用 HIGO 接口失败。
512	HI_ERR_WM_CREATE	窗口管理模块错误码的基准值。
513	HI_ERR_TYPE_NOEXISIT	指定 widget 类型不存在。
514	HI_ERR_MULTIPLE_THREAD_CALL	不支持在其它线程中调用。
515	HI_ERR_WIDGET_INVALDB	绑定的 DB 不可用。

错误代码	宏定义	描述
516	HI_ERR_WIDGET_SHOW	Widget 已显示。
517	HI_ERR_WIDGET_HIDE	Widget 已隐藏。
518	HI_ERR_INVSIBLE	剪切矩形为空，导致不能画。
519	HI_ERR_DDB_ZEROFIELD	没有字段属性。
520	HI_ERR_DDB_OUTOFRANGE	超出边界。
521	HI_ERR_DDB_NULLPTR	空指针。
522	HI_ERR_DDB_INVAIDPARA	非法参数。
523	HI_ERR_DDB_BUFFSMALL	BUFFER 太小。
524	HI_ERR_ADM_ZEROFIELD	没有字段属性。
525	HI_ERR_ADM_OUTOFRANGE	超出边界。
526	HI_ERR_ADM_NULLPTR	空指针。
527	HI_ERR_ADM_INVAIDPARA	非法参数。
528	HI_ERR_ADM_BUFFSMALL	BUFFER 太小。
529	HI_ERR_ADM_GETDATA	获取数据失败。
530	HI_ERR_ADM_UNKNOWNOPT	数据库未知操作。
531	HI_ERR_LANG_INVALID_LOCALE	没有设置多语言的 LOCALE。
532	HI_ERR_IM_OPENIRDEVICE	打开 IR 设备失败。
533	HI_ERR_IM_GETIRVALUE	获取 IR 值失败。
534	HI_ERR_IM_NOVIRKEY	没有对应的虚拟键。
535	HI_ERR_CHARSET_UNSUPPORT	不支持的字符集。
536	HI_ERR_CHARSET_CONVERT	无效的字符集转换。
537	HI_ERR_PARSER_NOINIT	未初始化。
538	HI_ERR_PARSER_INITED	已初始化。

错误代码	宏定义	描述
540	HI_ERR_PARSER_VERIFY	校验码错误
541	HI_ERR_PARSER_MARK	文件标识错误。
542	HI_ERR_PARSER_TYPE	文件类型错误。
543	HI_ERR_PARSER_DATALEN	数据长度错误。
544	HI_ERR_PARSER_UNSUPPORT	不支持。
545	HI_ERR_PARSER_DATAERR	HIGVBin 文件数据错误。
546	HI_ERR_PARSER_VIEWNOLOAD	视图未全部加载。
547	HI_ERR_PARSER_FILELOADED	文件已加载。
548	HI_ERR_PARSER_FILENOTLOAD	文件未加载。

表3-2 HiGO 模块的错误码

错误代码	宏定义	描述
0xB0008000	HIGO_ERR_NOTINIT	所依赖的模块未初始化
0xB0008001	HIGO_ERR_DEINITFAILED	模块初始化失败
0xB0008002	HIGO_ERR_INITFAILED	模块去初始化失败
0xB0008003	HIGO_ERR_NULLPTR	传入参数为空指针
0xB0008004	HIGO_ERR_INVHANDLE	传入无效的句柄
0xB0008005	HIGO_ERR_NOMEM	内存不足
0xB0008006	HIGO_ERR_INTERNAL	内部错误
0xB0008007	HIGO_ERR_INVSRCTYPE	无效的 IO 来源
0xB0008008	HIGO_ERR_INVFILE	无效的文件，文件操作失败
0xB0008009	HIGO_ERR_INVPARAM	无效的参数
0xB000800A	HIGO_ERR_INUSE	此句柄正在被使用

错误代码	宏定义	描述
0xB000800B	HIGO_ERR_UNSUPPORTED	无效的操作
0xB000800C	HIGO_ERR_DEPEND_TDE	依赖 TDE 出错
0xB000800D	HIGO_ERR_DEPEND_FB	依赖 FB 出错
0xB000800E	HIGO_ERR_DEPEND_MMZ	依赖 MMZ 出错
0xB000800F	HIGO_ERR_DEPEND_JPEG	依赖 JPEG 解码出错
0xB0008010	HIGO_ERR_DEPEND_PNG	依赖 PNG 解码出错
0xB0008011	HIGO_ERR_DEPEND_BMP	依赖 BMP 解码出错
0xB0008012	HIGO_ERR_DEPEND_GIF	依赖 GIF 解码出错
0xB0008013	HIGO_ERR_DEPEND_CURSOR	依赖 CURSOR 解码出错
0xB0008014	HIGO_ERR_DEPEND_JPGE	依赖 JPEG 编码失败
0xB0018000	HIGO_ERR_INVSURFACESIZE	surface 尺寸不正确
0xB0018001	HIGO_ERR_INVSURFACEPF	surface 像素格式不正确
0xB0018002	HIGO_ERR_NOTLOCKED	surface 未锁定，不能进行 surface 解锁操作
0xB0018003	HIGO_ERR_LOCKED	surface 已锁定，对 surface 进行的写操作被禁止
0xB0018004	HIGO_ERR_NOCOLORKEY	surface 不含有 colorKey 值
0xB0038000	HIGO_ERR_INVSIZE	无效的图层大小
0xB0038001	HIGO_ERR_INVLAYERID	无效的硬件图层 ID
0xB0038002	HIGO_ERR_INVPIXELFMT	无效的像素格式
0xB0038003	HIGO_ERR_INVFLUSHTYPE	图层刷新模式错误
0xB0038004	HIGO_ERR_FREEMEM	释放显存失败
0xB0038005	HIGO_ERR_CLOSELAYERFAILED	关闭图层设备失败
0xB0038006	HIGO_ERR_CANNOTCHANGE	图层 Z 序不可改变

错误代码	宏定义	描述
0xB0038007	HIGO_ERR_INVORDERFLAG	无效的 Z 序修改标志
0xB0038008	HIGO_ERR_SETALPHAFAILED	设置 surface alpha 失败
0xB0038009	HIGO_ERR_ALREADYSHOW	图层已经显示
0xB003800A	HIGO_ERR_ALREADYHIDE	图层已经隐藏
0xB003800B	HIGO_ERR_INVLAYERPOS	无效的图层起始位置
0xB003800C	HIGO_ERR_INVLAYERSURFACE	无效的 surface，表示对齐失败
0xB003800F	HIGO_ERR_INVANILEVEL	无效的图层抗闪烁级别
0xB0038010	HIGO_ERR_NOTOPEN	图层没有打开
0xB0038011	HIGO_ERR_FB_OPEN_FAILURE	打开 fb 设备失败
0xB0038012	HIGO_ERR_FB_GET_VSCREENINFO_FAILURE	获取可变信息失败
0xB0038013	HIGO_ERR_FB_PUT_VSCREENINFO_FAILURE	设置可变信息失败
0xB0038014	HIGO_ERR_FB_GET_LAYERINFO_FAILURE	获取图层信息失败
0xB0038015	HIGO_ERR_FB_PUT_LAYERINFO_FAILURE	设置图层信息失败
0xB0038016	HIGO_ERR_FB_GET_LAYERALPHA_FAILURE	获取图层 alpha 失败
0xB0038017	HIGO_ERR_FB_PUT_LAYERALPHA_FAILURE	设置图层 alpha 失败
0xB0038018	HIGO_ERR_FB_WRONG_LAYER_ID	错误的图层 ID
0xB0038019	HIGO_ERR_FB_REFRESH_FAILURE	刷新失败
0xB003801A	HIGO_ERR_FB_GET_CANVAS_BUFFER_FAILURE	获取 CANVAS BUFFER 失败
0xB003801B	HIGO_ERR_FB_PUT_CANVAS_BUFFER_FAILURE	设置 CANVAS BUFFER 失败
0xB003801C	HIGO_ERR_FB_GET_ZORDER_FAILURE	获取 zorder 失败

错误代码	宏定义	描述
0xB003801D	HIGO_ERR_FB_PUT_ZORDER_FAILURE	设置 zorder 失败
0xB0048000	HIGO_ERR_INVCOMPTYPE	错误的混合模式
0xB0048001	HIGO_ERR_INVCKEYTYPE	无效的 colorKey 操作
0xB0048002	HIGO_ERR_INVMIRRORTYPE	无效的镜像操作
0xB0048003	HIGO_ERR_INVROTATETYPE	无效的旋转操作
0xB0048004	HIGO_ERR_INVROPTYPE	无效的 ROP 操作
0xB0048005	HIGO_ERR_INVSCALING	缩放比例异常
0xB0048006	HIGO_ERR_OUTOFBOUNDS	矩形超出边界
0xB0048007	HIGO_ERR_EMPTYRECT	空矩形
0xB0048008	HIGO_ERR_OUTOFPAL	颜色不在调色板中
0xB0058000	HIGO_ERR_INVIMAGETYPE	无效的图片类型
0xB0058001	HIGO_ERR_INVINDEX	无效图片索引号
0xB0058002	HIGO_ERR_INVIMGDATA	无效图片数据
0xB0068000	HIGO_ERR_INVRECT	无效的矩形区域
0xB0068001	HIGO_ERR_UNSUPPORT_CHARSET	不支持的字符集
0xB0068002	HIGO_ERR_ISUSING	正在使用中
0xB0068003	HIGO_ERR_NOIMPLEMENT	该函数尚未实现
0xB0068004	HIGO_ERR_SHAPEFAILED	整形失败
0xB0068005	HIGO_ERR_MAX_CHAR	字符数量过多
0xB0068006	HIGO_ERR_CHAR_SET	字符集编码错误
0xB0068007	HIGO_ERROR_BIDI	双向处理错误
0xB006801F	HIGO_ERR_TEXTINTERNAL	内部错误
0xB0078000	HIGO_ERR_ALREADYBIND	桌面与图层已经绑定

错误代码	宏定义	描述
0xB0078001	HIGO_ERR_INVZORDERTYPE	无效的 Z 序调整方式
0xB0078002	HIGO_ERR_NOUPDATE	桌面无更新
0xB0078003	HIGO_ERR_INVPF	桌面无更新
0xB0078004	HIGO_ERR_INVTREE	桌面无更新
0xB0078005	HIGO_ERR_ALREADYSETMODE	已经设定窗口内存模式
0xB0088000	HIGO_ERR_INVHOTSPOT	无效的 cursor 热点坐标
0xB0088001	HIGO_ERR_NOCURSORINF	没有设置 cursor 信息

4 开发指引

4.1 HiGV 应用程序创建流程

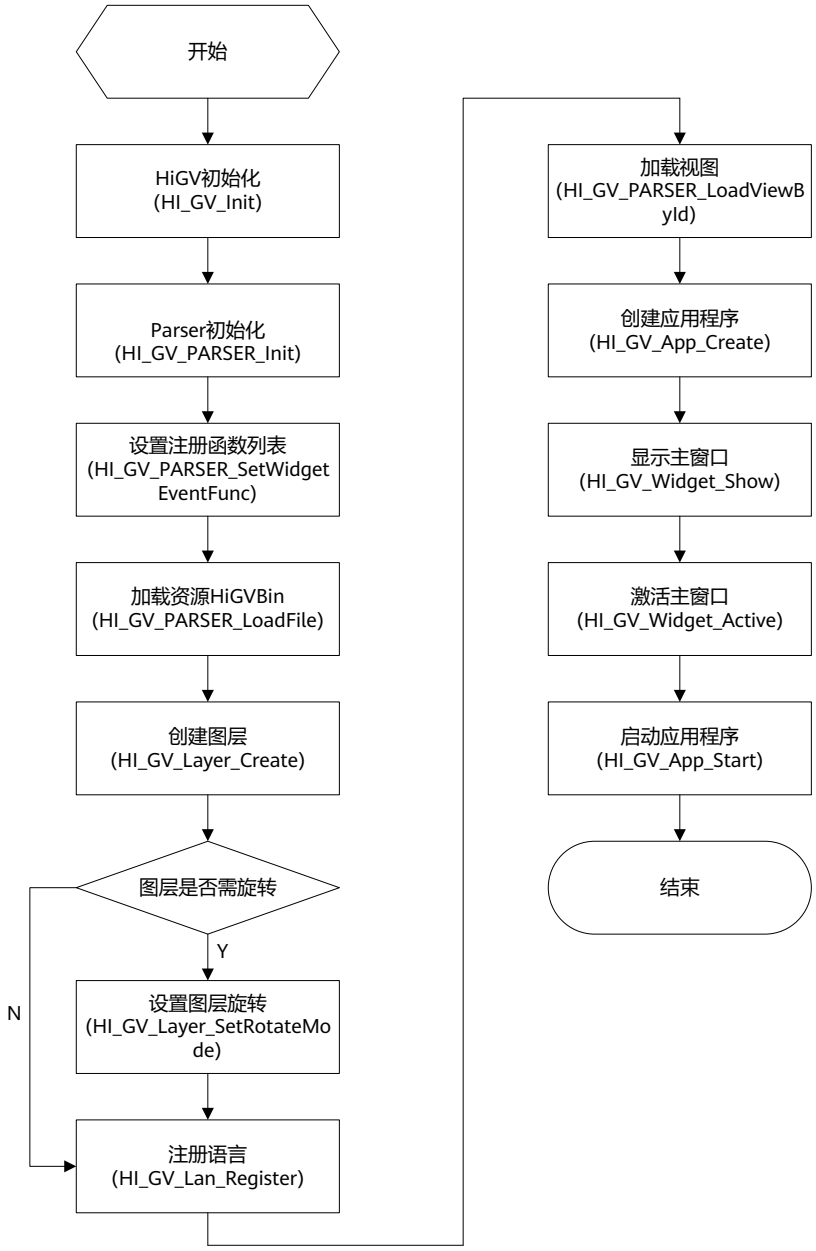
场景描述

创建一个 HiGV 应用程序。

工作流程

创建并启动 HiGV 应用程序业务流程如图 4-1 所示。

图4-1 创建 HiGV 应用程序业务流程



编程指导

创建 HiGV 应用程序业务流程如下：

步骤 1 初始化 HiGV。

步骤 2 初始化解析 XML 模块。

步骤 3 设置注册回调函数列表。

步骤 4 加载资源 HiGVBin 文件。

步骤 5 创建图层。

步骤 6 设置图层参数。

步骤 7 注册语言。

步骤 8 加载视图。

步骤 9 创建应用程序实例。

步骤 10 显示主窗口。

步骤 11 激活主窗口。

步骤 12 启动应用程序。

----结束

注意事项

- 设置旋转请参考 HI_GV_Layer_SetRotateMode 接口限制描述。
- 如需支持多语言则创建应用时注册多个语言，并选定其一为默认语言。

示例

具体请参考发布包中的 HiGV Sample 实例。

