

[WGI demo]

The CDS Definition of WGI demo Project

[V0.7]

版权所有，保留所有权利！

作者：赵平智

2011. 08. 06

★ 版权所有，保留所有权利 ★

文档修订记录

版本号	变更内容	日期	变更人
V0.5	创建。	20110706	赵平智
V0.7	补充。	20110806	赵平智

版权所有，保留所有权利！

目录

1	前言	5
2	宏定义.....	5
2.1	文本格式化方式	5
2.2	可视对象的风格	5
3	数据类型.....	5
3.1	CAPTIONINFO (标题信息)	5
3.2	TEXTINFO (文本信息)	6
3.3	POLYGON (多边形结构体)	7
3.4	ZPOLYGON (Z 轴多边形结构体)	7
3.5	RSCoord (矩形空间坐标结构体)	7
3.6	ATVCLNSLLNode (主动碰撞单向链表节点结构体)	8
3.7	MOVE (改变对象的矩形空间位置或改变矩形空间尺寸时的输入流)	8
3.8	PSVCollision (通知对象“被动碰撞”时的输入流)	9
3.9	ZPLGSLND (Z 轴多边形单向链表节点结构体)	9
3.10	EVOBSTATE (可视对象的 (外观) 状态枚举)	9
4	扩展成员.....	11
4.1	EM_VOBJECT.....	11
4.2	EM_WOBJECT.....	14
4.3	EM_WPANEL.....	15
4.4	EM_COBJECT.....	15
4.5	EM_CBUTTON.....	16
5	初始化.....	16
5.1	INIT_VOBJECT.....	16
5.2	INIT_WOBJECT.....	17
5.3	INIT_COBJECT.....	17
6	行为	17
6.1	AC0.....	17
6.1.1	CMD_WGI_SHOW (显示)	17
6.1.2	CMD_WGI_HIDE (隐藏)	17
6.1.3	CMD_WGI_ENABLE (使能)	17
6.1.4	CMD_WGI_DISABLE (禁能)	17
6.1.5	CRT_WGI_WINDOW (创建窗口)	17

6.1.6	DTR_WGI_WINDOW (销毁窗口)	17
6.1.7	MAK_WGI_CURRENTBITMAP0 (生成当前位图)	17
6.1.8	CMD_WGI_PAINT (显示当前位图)	17
6.1.9	MSG_WGI_ACTIVATE (激活对象)	17
6.1.10	MSG_WGI_DEACTIVATE (使对象退出激活状态)	17
6.1.11	MSG_WGI_SELECT (选中对象)	17
6.1.12	MSG_WGI_FOCUSOVERLEAVE (焦点 OVER 离开)	18
6.1.13	MSG_WGI_FOCUSONLEAVE (焦点 ON 离开)	18
6.1.14	CMD_WGI_PAGEUP (上翻页)	18
6.1.15	CMD_WGI_PAGEDOWN (下翻页)	18
6.1.16	CMD_WGI_PAGELEFT (左翻页)	18
6.1.17	CMD_WGI_PAGERIGHT (右翻页)	18
6.1.18	CMD_WGI_REPAINTSUPERIOR (重新绘制上级对象)	18
6.2	AC1	18
6.2.1	GET_WGI_CAPTIONINFO (获取标题信息)	18
6.2.2	GET_WGI_VALUE (获取数值)	18
6.2.3	GET_WGI_TEXTINFO (获取文本信息)	18
6.3	AC2	18
6.3.1	CMD_WGI_MOVE (改变对象的矩形空间位置或改变矩形空间尺寸)	18
6.3.2	SET_WGI_CAPTIONINFO (设置标题信息)	18
6.3.3	SET_WGI_VALUE (设置数值)	18
6.3.4	SET_WGI_TEXTINFO (设置文本信息)	18
6.3.5	EVT_WGI_PSYCOLLISION (被动碰撞事件)	18
6.3.6	MAK_WGI_CURRENTBITMAP2 (生成当前位图)	18
6.3.7	MSG_WGI_FOCUSOVERMOVE (焦点 OVER 移动)	18
6.3.8	MSG_WGI_FOCUSUP (焦点 UP)	18
6.3.9	MSG_WGI_FOCUSDOWN (焦点 DOWN)	19
6.3.10	MSG_WGI_FOCUSONMOVE (焦点 ON 移动)	19
6.3.11	MSG_WGI_FOCUSCLICK (焦点 CLICK)	19
6.3.12	MSG_WGI_FOCUSDDOUBLECLICK (焦点 DOUBLE-CLICK)	19
6.4	AC3	19
7	结果 (RESULT)	19
7.1	N 结果	19
7.2	O 结果	19
7.2.1	负 O 结果	19

7.2.1.1	RO_SUPERIORNULL (上级对象指针是 NULL)	19
7.2.2	正 O 结果.....	19
7.3	P 结果.....	19
8	XXX.....	19

版权所有，保留所有权利！

1 前言

WGI 是用 OIOIC 技术实现的 GUI（图形用户界面）库，也是 OIOIC 技术的代表作。
OIOIC 是一种专为 C 语言设计的全新的面向对象编程机制。

WGI 下载地址: <http://code.google.com/p/oic-wgi/downloads/list>

OIOIC 下载地址: <http://code.google.com/p/oioic/downloads/list>

2 宏定义

2.1 文本格式化方式

```
#define TF_LEFT DT_LEFT // aligns text to the left
#define TF_RIGHT DT_RIGHT // aligns text to the right
#define TF_CENTER DT_CENTER // centers text horizontally
```

2.2 可视对象的风格

```
#define VOBS_IMMOVABLE 0x00000001L // 1: Immovable, 0: Movable
#define VOBS_DISABLED 0x00000002L // 1: Disabled, 0: Enabled
#define VOBS_HIDDEN 0x00000004L // 1: Hidden, 0: Showed.
#define VOBS_XIRROLLABLE 0x00000008L // 1: Irrollable By X, 0: Rollable By X.
#define VOBS_YIRROLLABLE 0x00000010L // 1: Irrollable By Y, 0: Rollable By Y.
#define VOBS_INACTIVABLE 0x00000020L // 1: Inactivable, 0: Activable
#define VOBS_NOABSinACTIVE 0x00000040L // 1: no the state of "eVOBState_ABSInactive", 0: has the state of "eVOBState_ABSInactive".
```

批注 [微软用户1]: 取消, 用状态和行为实现。

3 数据类型

3.1 CAPTIONINFO（标题信息）

```
typedef struct TAG_CAPTIONINFO
{
    TCHAR* text;
    long cpty;
    long clen;
    RECT rect;
    UINT format;
    HFONT font;
    COLORREF color;
} CAPTIONINFO;
```

Members

text

指向文本缓冲区。

cpty

文本缓冲区容量, 以字符为单位。

clen

文本缓冲区内的当前字符串长度, 以字符为单位。

rect

矩形区域, 文本只能在这个区域内书写。坐标相对于对象的外部矩形。

format

文本格式化方式。

Value	Description
TF_LEFT	左对齐矩形区域。
TF_RIGHT	右对齐矩形区域。
TF_CENTER	居中于矩形区域。

font

文本字体。

color

文本颜色。

3.2 TEXTINFO (文本信息)

```
typedef struct TAG_TEXTINFO
```

```
{  
    TCHAR*    text;  
    long      cpty;  
    long      clen;  
    RECT      rect;  
    UINT      format;  
    HFONT     font;  
    COLORREF  color;  
} TEXTINFO;
```

Members

text

指向文本缓冲区。

cpty

文本缓冲区容量, 以字符为单位。

clen

文本缓冲区内的当前字符串长度, 以字符为单位。

rect

矩形区域, 文本只能在这个区域内书写。坐标相对于对象的外部矩形。

format

文本格式化方式。

Value	Description
-------	-------------

TF_LEFT	左对齐矩形区域。
TF_RIGHT	右对齐矩形区域。
TF_CENTER	居中于矩形区域。

font

文本字体。

color

文本颜色。

3.3 POLYGON（多边形结构体）

```
typedef struct TAG_POLYGON
{
    POINT*    pAPT;
    int       npt;
}POLYGON;
```

Members

pAPT

指向点坐标数组，顺时针方向，这些点的坐标值都相对于对象的外部矩形空间。

npt

数组里点的个数。

3.4 ZPOLYGON（Z 轴多边形结构体）

```
typedef struct TAG_ZPOLYGON
{
    POLYGON    plg;
    int        z;
}ZPOLYGON;
```

Members

plg

多边形。

z

Z 轴坐标值，相对于对象矩形空间的 Z 坐标。

3.5 RSCOORD（矩形空间坐标结构体）

```
typedef struct TAG_RSCORDINATE
{
    RECT       rcouter;
    RECT       rcinner;
    int        z;
}RSCOORD;
```

Members

rcouter

对象的外部矩形 (outer rectangle) 坐标相对于上级对象的内部矩形。

rcinner

对象的内部矩形 (inner rectangle), 坐标相对于外部矩形。

z

Z 轴坐标值, 相对于上级对象的矩形空间的 Z 坐标。

Members

rect

对象的矩形空间 (rectangle space)。

z

对象的 Z 轴坐标值。

3.6 ATVCLNSLLNODE (主动碰撞单向链表节点结构体)

主动碰撞 (active collision) 单向链表节点结构体:

```
typedef struct TAG_ATVCOLLISIONSLLNODE
{
    OBJECT*                pObj;
    ZPLGSLLND*             pRgn;
    TAG_ATVCOLLISIONSLLNODE* next;
} ATVCLNSLLNODE;
```

Members

pObj

被碰撞对象的指针。

pRgn

碰撞的区域 (自己的)。

next

下一个节点。

3.7 MOVE (改变对象的矩形空间位置或改变矩形空间尺寸时的输入流)

```
typedef struct TAG_MOVE
{
    RECT                rcouter;
    int                 dxUL;
    int                 dxLR;
    int                 dyUL;
    int                 dyLR;
    int                 dz;
    ATVCLNSLLNODE*      pacsn;
} MOVE;
```

批注 [微软用户2]: 去掉, 功能由 dxUL、dxLR、dyUL、dyLR 实现。

Members

dxUL

矩形空间左上角 X 坐标值的增量, 可以为负值。(UL: upper-left)

dxLR

矩形空间右下角 X 坐标值的增量, 可以为负值。(LR: lower-right)

dyUL

矩形空间左上角 Y 坐标值的增量, 可以为负值。

dyLR

矩形空间右下角 Y 坐标值的增量, 可以为负值。

dz

矩形空间的 Z 坐标值的增量, 可以为负值。

pacsn

主动碰撞单向链表的第一个节点指针。

3.8 PSVCOLLISION (通知对象“被动碰撞”时的输入流)

被动碰撞 (passive collision) 结构体:

```
typedef struct TAG_PSVCOLLISION
```

```
{  
    OBJECT*      pObj;  
    ZPLGSSLND*   pRgn;  
} PSVCOLLISION;
```

Members

pObj

主动碰撞对象的指针。

pRgn

被碰撞的区域。

3.9 ZPLGSSLND (Z 轴多边形单向链表节点结构体)

Z 轴多边形单向链表 (Singly Linked List) 节点结构体:

```
typedef struct TAG_ZPOLYONSSLNODE
```

```
{  
    ZPOLYGON      zp;  
    struct TAG_ZPOLYONSSLNODE* next;  
} ZPLGSSLND;
```

Members

zplg

该节点的 Z 轴多边形。

next

下一个节点。

3.10 EVOBSTATE (可视对象的 (外观) 状态枚举)

```
typedef enum TAG_EVOBSTATE
```

```
{
```

```
eVOBState_Normal,
eVOBState_FocusOver,
eVOBState_FocusOn,
eVOBState_Active,
eVOBState_ABSInactive,
eVOBState_Selected,
eVOBState_Disable,
eVOBState_Hidden,
/*- number of state -*/
eNumVOBState
} EVOBSTATE;
```

Members

eVOBState_Normal

正常状态。

eVOBState_FocusOver

焦点 OVER。

eVOBState_FocusOn

焦点 ON。

eVOBState_Active

活动状态。不能直接转为“FocusOver”和“FocusOn”状态，需要“Normal”状态中转。

eVOBState_ABSInactive

处于活动状态但上级对象（Active But Superior）处于非活动状态。

eVOBState_Selected

选中状态。不能直接转为“FocusOver”和“FocusOn”状态，需要“Normal”状态中转。

eVOBState_Disable

禁止使用。

eVOBState_Hidden

隐藏状态。

状态转换表							
	Normal	FocusOver	FocusOn	Active	ABSInactive	Selected	Disable
Normal	✓	✓	✓	✓	×	✓	✓
FocusOver	✓	✓	✓	✓	×	✓	✓
FocusOn	✓	✓	✓	✓	×	✓	✓
Active	✓	✓	✓	✓	✓	✓	✓
ABSInactive	✓	✓	✓	✓	✓	✓	✓
Selected	✓	✓	✓	✓	✓	✓	✓
Disable	✓	✓	✓	✓	✓	✓	✓

4 扩展成员

4.1 EM_VOBJECT

```
typedef struct TAG_EM_VOBJECT
```

```
{
    OBJECT*      pOIIOC;
    OBJECT*      pParent;
    OBJECT*      pSuperior;
    OBJECT*      pFirstSbdt;
    OBJECT*      pPrevCotd;
    OBJECT*      pNextCotd;
    OBJECT*      pFirstIIRSbdt;
    OBJECT*      pPrevISIRCotd;
    OBJECT*      pNextISIRCotd;
    OBJECT*      pActiveSbdt;
    OBJECT*      pCFOverSbdt;
    OBJECT*      pCFDownSbdt;
    OBJECT*      pCFOnSbdt;
    OBJECT*      pCFUpSbdt;
    long         ox_irr;
    long         oy_irr;
    long         maxw_irr;
    long         maxh_irr;
    HDC          hDC;
    const HWND*   pWndHandle;
    HBITMAP       ArrHBITMAP[eNumVOBState];
    HBITMAP       hBmpNormal;
    HBITMAP       hBmpFocusOver;
    HBITMAP       hBmpFocusOn;
    HBITMAP       hBmpActive;
    HBITMAP       hBmpABSLFocus;
    HBITMAP       hBmpSelected;
    HBITMAP       hBmpDisable;
    HBITMAP       hBmpCurrent;
    HBITMAP       hBmpCurBkgd;
    UI32          style;
    EVOBSTATE     state;
    CAPTIONINFO   ci;
    RSCoord       rsc;
    ZPLGSLND      as;
    ZPLGSLND      rgn;
    POINT         lastpt;
} EM_VOBJECT;
```

批注 [微软用户3]: 不需要, 功能由 hDC 实现。

批注 [微软用户4]: 去掉, 由 pWndHandle 代替。

批注 [微软用户5]: 取消, 由 ArrHBITMAP 数组取代。

批注 [微软用户6]: 不需要, 功能由 state 实现。

Members**pOIIC**

基对象 OIIC 的指针。

pParent

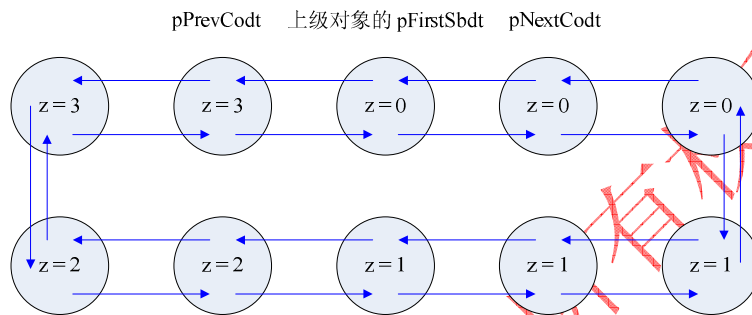
父对象指针。

pSuperior

上级对象指针。

pFirstSbdt

第一个下级 (subordinate) 对象指针, 这个对象必须是第一个矩形空间的 z 坐标值等于 0 的下级对象。



【图1】一个双向循环链表说明 pPrevCordt, pFirstSbdt, pNextCordt 三者之间的关系

pPrevCordt

上一个同级 (coordinate) 对象指针, 与 pNextCordt 共同构成双向循环链表, 参见【图1】。

pNextCordt

下一个同级对象指针。与 pPrevCordt 共同构成双向循环链表, 链表中的对象元素是按矩形空间的 z 坐标值向下递增的顺序循环排列, 相同 z 坐标值的对象元素都排在一起, 参见【图1】。

pFirstIIRSbdt

第一个在内部矩形空间里 (IIR: In Inner Rectangle) 的下级对象。只绘制显示在内部矩形空间里的下级对象, 由于滚动或翻页而移出内部矩形空间的下级对象将不会显示。pFirstIIRSbdt、pPrevISIRCordt 和 pNextISIRCordt 三者如同 pFirstSbdt、pPrevCordt 和 pNextCordt, 也是构成双向循环链表的关系, 但这个双向循环链表是后三者构成的双向循环链表的子集。引入这个小链表的目的是: 当有焦点事件发生时, 只需在这个链表里寻找响应焦点事件的下级对象, 而无须在 pFirstSbdt 所指的大链表里寻找, 提高响应速度。

pPrevISIRCordt

上一个在上级对象的内部矩形空间里 (ISIR: In Superior's Inner Rectangle) 的同级对象。

pNextISIRCordt

下一个在上级对象的内部矩形空间里的同级对象。

pActiveSbdt

指向活动的下级对象, 只能有一个被活动的下级对象。

pCFOverSbdt

当前焦点 OVER (CF: Currently Focus) 的下级对象。

pCFDownSbdt

当前焦点 DOWN 的下级对象。

pCFOnSbdt

批注 [微软用户7]: 不需要, 功能由 hDC 实现。

当前焦点 ON 的下级对象。

pCFUpSbdt

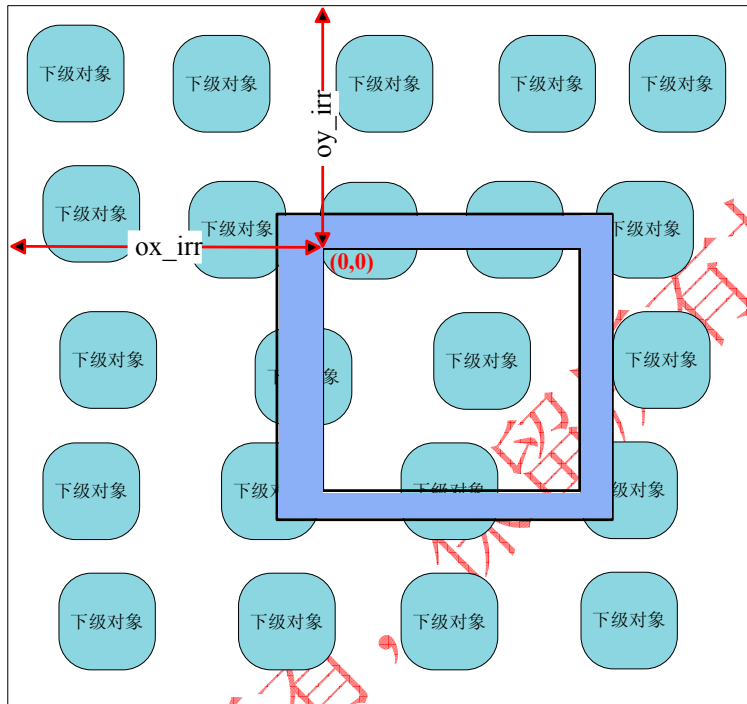
当前焦点 UP 的下级对象。

ox_irr

内部矩形滚动的 X 坐标偏移量, 数值范围 ≤ 0 。参见【图 2】。(ox: offset x, irr: inner rectangle roll)

oy_irr

内部矩形滚动的 Y 坐标偏移量, 数值范围 ≤ 0 。参见【图 2】。



【图2】内部矩形滚动区域的滚动偏移量 ox_irr 和 oy_irr

maxw_irr

内部矩形的最大滚动宽度, 默认值是内部矩形的宽度, 数值范围 ≥ 0 。

maxh_irr

内部矩形的最大滚动高度, 默认值是内部矩形的高度, 数值范围 ≥ 0 。

pWndHandle

指向窗口句柄。

为什么不直接定义为句柄类型而定义为句柄的指针类型呢? 因为控件对象没有窗口句柄, 但可以拥有指向其所依附的窗口对象的窗口句柄的指针, 所以定义句柄指针是合理的设计。例如, 把“住址”当做人的一个属性和把“住宅”当做人的一个属性, 前者的设计更合理。

ArrHBITMAP[eNumVOBState]

各种状态时的图片。

hBmpCurrent

当前显示的位图, 有可能在位图上已经绘制了下级对象。

hBmpCurBkgd

当前背景位图, 背景位图即没在上面写字也没在上面绘制下级对象的干净位图。某下级对象改变尺寸或移动时, 需要用这个位图刷新背景。该位图句柄没有自己的位图实例, 只是从其它位图句柄赋值, 所以不需要销毁操作。任何时候数值只是 NULL、hBmpNormal、hBmpWFocus、hBmpCFocus、hBmpActivated、hBmpABSLFocus、hBmpDisable 其中之一。

批注 [微软用户8]: 不需要, 功能由 state 实现。

style

指定风格, 可以是下面数值的任意组合。

Value	Description
VOBS_IMMOVABLE	不能移动。
VOBS_DISABLED	禁止使用。
VOBS_HIDDEN	隐藏的, 隐藏时不响应外界事件。
VOBS_XIRROLLABLE	不能沿 X 坐标方向滚动。
VOBS_YIRROLLABLE	不能沿 Y 坐标方向滚动。
VOBS_INACTIVABLE	不能激活的。
VOBS_NOABSINACTIVE	没有 “eVOBState_ABSInactive” 状态。

批注 [微软用户9]: 取消, 用状态和行为实现。

批注 [微软用户10]: 取消, 用状态行为实现。

state

状态, 数值是 EVOBSTATE 枚举。

rsc

对象的矩形空间坐标。

ci

对象的标题, 详见 CAPTIONINFO 结构体。

as

实际空间, 里面的各级 next 指针都是指向对象的空心区域。如果 as.zp.plg.npt 为 0, 表示实际空间就是外部矩形。

rgn

实际空间里的第一个区域, 里面的 next 指针指向下一个区域。如果为 rgn.zp.plg.npt 为 0, 表示实际空间只有一个区域, 而且这个区域就是外部矩形。

lastpt

记录焦点在此对象上的最后坐标, 坐标值相对于此对象自己的外部矩形。

4.2 EM_WOBJECT

```
typedef struct TAG_EM_WOBJECT
{
    OBJECT*      pOIIC;
    OBJECT*      pVObject;
    HINSTANCE    hInstance;
    const TCHAR* pClassName;
    const TCHAR* pWindowName;
} EM_WOBJECT;
```

Members

pOIIC

基对象 OIIC 的指针。

pVObject

基对象 VObject 的指针。

hInstance

应用程序实例句柄。

pClassName

指向窗口的注册的类名称 (pointer to registered class name)。

pWindowName

指向窗口名称 (pointer to window name)。

4.3 EM_WPNEL

```
typedef struct TAG_EM_WPNEL
{
    OBJECT*    pOIIC;
    OBJECT*    pVObject;
    OBJECT*    pWObject;
} EM_WPNEL;
```

Members

pOIIC

基对象 OIIC 的指针。

pVObject

基对象 VObject 的指针。

pWObject

基对象 WObject 的指针。

4.4 EM_COBJECT

```
typedef struct TAG_EM_COBJECT
{
    OBJECT*    pOIIC;
    OBJECT*    pVObject;
    long       acx;
    long       acy;
} EM_COBJECT;
```

批注 [微软用户11]: 去掉

Members

pOIIC

基对象 OIIC 的指针。

pVObject

基对象 VObject 的指针。

acx

外部矩形左上角的 X 轴的绝对坐标值, 相对于 Panel 上级对象的外部矩形的左上角。(ac: absolute coordinate)

acy

外部矩形左上角的 Y 轴的绝对坐标值, 相对于 Panel 上级对象的外部矩形的左上角。

批注 [微软用户12]: 去掉

4.5 EM_CBUTTON

```
typedef struct TAG_EM_CBUTTON
{
    OBJECT*    pOIIC;
    OBJECT*    pVObject;
    OBJECT*    pCObject;
}EM_CBUTTON;
```

Members

pOIIC

基对象 OIIC 的指针。

pVObject

基对象 VObject 的指针。

pCObject

基对象 CObject 的指针。

5 初始化

5.1 INIT_VOBJECT

```
typedef struct TAG_INIT_VOBJECT
{
    OBJECT*    pSuperior;
    OBJECT*    pFirstSbdt;
    OBJECT*    pPrevCodt;
    OBJECT*    pNextCodt;
    OBJECT*    pFirstIIRSbdt;
    OBJECT*    pPrevISIRCodt;
    OBJECT*    pNextISIRCodt;
    long       ox_irr;
    long       oy_irr;
    long       maxw_irr;
    long       maxh_irr;
    HDC        hDC;
    HBITMAP    ArrHBITMAP[eNumVOBState];
    UI32       style;
    EVOBSTATE  state;
    CAPTIONINFO ci;
    RSCoord    rsc;
    ZPLGSLLND  as;
    ZPLGSLLND  rgn;
} INIT_VOBJECT;
```

批注 [微软用户13]: 去掉

批注 [微软用户14]: 去掉

5.2 INIT_WOBJECT

```
typedef struct TAG_INIT_WOBJECT
{
    HINSTANCE      hInstance;
    HWND           hWnd;
    const TCHAR*   pClassName;
    const TCHAR*   pWindowName;
} INIT_WOBJECT;
```

批注 [微软用户15]: 去掉

5.3 INIT_COBJECT

```
typedef struct TAG_INIT_COBJECT
{
    const HWND*    pWndHandle;
} INIT_COBJECT;
```

6 行为

6.1 AC0

6.1.1 CMD_WGI_SHOW (显示)

6.1.2 CMD_WGI_HIDE (隐藏)

6.1.3 CMD_WGI_ENABLE (使能)

6.1.4 CMD_WGI_DISABLE (禁能)

6.1.5 CRT_WGI_WINDOW (创建窗口)

6.1.6 DTR_WGI_WINDOW (销毁窗口)

DTR: DesTRoy

6.1.7 MAK_WGI_CURRENTBITMAP0 (生成当前位图)

生成当前位图 (使用被访问对象的当前状态的位图), 实现节点: VObject。

6.1.8 CMD_WGI_PAINT (显示当前位图)

显示当前位图, 实现节点: VObject。

6.1.9 MSG_WGI_ACTIVATE (激活对象)

6.1.10 MSG_WGI_DEACTIVATE (使对象退出激活状态)

6.1.11 MSG_WGI_SELECT (选中对象)

选中下级对象。

6.1.12 MSG_WGI_FOCUSOVERLEAVE (焦点 OVER 离开)

6.1.13 MSG_WGI_FOCUSONLEAVE (焦点 ON 离开)

6.1.14 CMD_WGI_PAGEUP (上翻页)

上翻页对象内部矩形空间里的内容 (下级对象)。

6.1.15 CMD_WGI_PAGEDOWN (下翻页)

下翻页对象内部矩形空间里的内容 (下级对象)。

6.1.16 CMD_WGI_PAGELEFT (左翻页)

左翻页对象内部矩形空间里的内容 (下级对象)。

6.1.17 CMD_WGI_PAGERIGHT (右翻页)

右翻页对象内部矩形空间里的内容 (下级对象)。

6.1.18 CMD_WGI_REPAINTSUPERIOR (重新绘制上级对象)

重新绘制上级对象, 实现节点: VObject

6.2 AC1

6.2.1 GET_WGI_CAPTIONINFO (获取标题信息)

TSO 是 CAPTIONINFO。

6.2.2 GET_WGI_VALUE (获取数值)

6.2.3 GET_WGI_TEXTINFO (获取文本信息)

TSO 是 TEXTINFO。

6.3 AC2

6.3.1 CMD_WGI_MOVE (改变对象的矩形空间位置或改变矩形空间尺寸)

TSI 是 MOVE, 实现节点: CObject。

6.3.2 SET_WGI_CAPTIONINFO (设置标题信息)

TSI 是 CAPTIONINFO。

6.3.3 SET_WGI_VALUE (设置数值)

6.3.4 SET_WGI_TEXTINFO (设置文本信息)

TSI 是 TEXTINFO。

6.3.5 EVT_WGI_PSVCOLLISION (被动碰撞事件)

TSI 是 PSVCOLLISION, 实现节点: CObject。

6.3.6 MAK_WGI_CURRENTBITMAP2 (生成当前位图)

用指定的位图做背景生成当前位图, TSI 是 HBITMAP (背景位图的句柄)。实现节点: VObject。

6.3.7 MSG_WGI_FOCUSOVERMOVE (焦点 OVER 移动)

焦点在空中移动。TSI 是 POINT (焦点的当前坐标), 要转换成相对于受访者的外部矩形空间的坐标值。

6.3.8 MSG_WGI_FOCUSUP (焦点 UP)

焦点升起。TSI 是 POINT (焦点的当前坐标), 要转换成相对于受访者的外部矩形空间的坐标值。焦点 UP 后就是 OVER 状态。

6.3.9 MSG_WGI_FOCUSDOWN (焦点 DOWN)

焦点落地。TSI 是 POINT (焦点的当前坐标), 要转换成相对于受访者的外部矩形空间的坐标值。
焦点 DOWN 后就是 ON 状态。

6.3.10 MSG_WGI_FOCUSONMOVE (焦点 ON 移动)

焦点在地面移动。TSI 是 POINT (焦点的当前坐标), 要转换成相对于受访者的外部矩形空间的坐标值。

6.3.11 MSG_WGI_FOCUSCLICK (焦点 CLICK)

TSI 是 POINT (焦点的当前坐标), 要转换成相对于受访者的外部矩形空间的坐标值。

6.3.12 MSG_WGI_FOCUSDOUBLECLICK (焦点 DOUBLE-CLICK)

TSI 是 POINT (焦点的当前坐标), 要转换成相对于受访者的外部矩形空间的坐标值。

6.4 AC3

7 结果 (Result)

7.1 N 结果

7.2 O 结果

7.2.1 负 O 结果

7.2.1.1 RO_SUPERIORNULL (上级对象指针是 NULL)

批注 [微软用户16]: 取消

7.2.2 正 O 结果

7.3 P 结果

8 xxx