

奥维互动地图浏览器

C++开发工具接口（V2.1）



北京元生华网软件有限公司

2019年04月12日

版权声明

未经北京元生华网软件有限公司明确的书面许可，本文的任何部分不得以任何方式或手段进行复制或传播，或用于任何用途。

对于本文涉及的任何主题，本公司已用于专利、专利应用程序、商标、版权或其他知识产权。提供本文，并不意味着向您授予这些专利、商标、版权或其他知识产权的任何许可。

©奥维互动地图浏览器开发工具接口（V2.1）

版权所有 未经许可 严禁复制

客服电话：400-893-8099

公司网址：www.ovital.com

公司邮箱：info@ovital.com

公司地址：北京市海淀区西小口路 66 号中关村东升科技园·北领地 D-3 楼 3 层 302 室

文档修改记录

版本号	变化状态	简要说明 (变更内容和变更范围)	变更日期	作者
V1.0	C	创建	2018年12月17日	叶志翔
V1.1	M	修改文档	2018年12月20日	叶志翔
V1.2	A	增加接口	2018年12月27日	叶志翔
V1.3	M	修改文档	2019年1月18日	叶志翔
V1.4	A	增加接口	2019年2月20日	叶志翔
V1.5	A	增加说明	2019年2月27日	叶志翔
V1.6	A	增加接口	2019年8月15日	叶志翔
V1.7	A	增加接口	2019年9月20日	叶志翔
V1.8	M	修改接口	2019年10月18日	叶志翔
V1.9	M	修改	2019年10月24日	叶志翔
V2.0	A	增加接口	2019年12月12日	叶志翔
V2.1	A	增加接口	2020年04月12日	叶志翔

*变化状态：C—创建，A—增加，M—修改，D—删除

目录

1. 奥维第三方接口文件简介	1
1.1. DLGTOOLBOX.H	1
1.2. OMAPAPI.H	1
1.3. OMAPINTERFACE.H	1
1.4. OMAPINTERFACEDLG.H	1
1.5. DLGTOOLBOX.CPP	1
1.6. OMAPAPI.CPP	1
1.7. OMAPINTERFACE.CPP	1
1.8. OMAPINTERFACEDLG.CPP	1
2. 详细功能介绍	2
2.1. 接口工作机制	2
2.1.1. 接口使用 VIP 要求	2
2.1.1.1. 接口示例程序与奥维互动地图交互图	2
2.1.2. 消息交互机制	3
2.2. 常量介绍	7
2.3. 结构体介绍	11
2.3.1. tagOmapLatlng	11
2.3.2. tagOmapLocation	11
2.3.3. tagOmapLocationExt	11
2.3.4. tagWinMapInfo	11
2.3.5. tagOPhotoMapPic	12
2.3.6. tagOmapGeoDecode	12
2.3.7. tagOmapGetMap	12
2.3.8. tagOmapGetMapFile	13
2.3.9. tagOmapReqContour	14

2.3.10. tagOMapBtnInfo	14
2.3.11. tagOApiObjectOptHdr.....	15
2.3.12. tagOApiItemListV1	15
2.3.13. tagOApiObjGroupV1	16
2.3.14. tagOApiObjGroupV2	17
2.3.15. tagOApiMapSignV1	18
2.3.16. tagOApiMapSignV2	19
2.3.17. tagOApiMapTrackPointV1	21
2.3.18. tagOApiMapTrackV1	21
2.3.19. tagOApiMapTrackDraw	21
2.3.20. tagOApiMapTrackV2	22
2.3.21. tagOApiMapShapeV1	23
2.3.22. tagOApiMapShapeV2	25
2.3.23. tagOApiLatLngV1	27
2.3.24. tagOApiMapDirectionsUnitV1	27
2.3.25. tagOApiMapDirectionsV1.....	28
2.3.26. tagOApiMapDirectionsV2.....	29
2.3.27. tagOApiInterfaceVer	30
2.3.28. tagOApiMemBuf	30
2.3.29. tagOApiObjDataItem	30
2.3.30. tagOApiObjectSrhReq	30
2.3.31. tagOApiObjectSrhRes	31
2.3.32. tagOApiObjChageHdr	31
2.3.33. tagOApiShapeFillCadPatHdr.....	32
2.3.34. tagOApiShapeFillCadPatItem	32
2.3.35. tagOApiShapeFillCadPatUnit	32
2.3.36. tagOmapReqContour	33
2.3.37. tagOApiMapObjEvent	34

2.3.38. tagOApiMapObjShowWnd.....	34
2.3.39. tagCusMainMenu.....	35
2.3.40. tagOApiRevHdr	35
2.3.41. tagOApMapTrackDetail.....	35
2.3.42. tagOApMapShapeDetail.....	35
2.3.43. tagCusMainBtn	36
2.3.44. tagCusFavMenu.....	36
2.3.45. tagCusMainMenu.....	36
2.3.46. tagOApiMapTrackDetail.....	36
2.3.47. tagOApiMapShapeDetail.....	37
2.3.48. tagOApiPrj.....	37
2.3.49. tagOApiCusShowWnd.....	38
2.3.50. tagOApiWebSocketConf.....	38
2.3.51. tagOApiCustomMap	39
2.4. 函数介绍.....	40
2.4.1. 字符类型转换函数.....	40
2.4.2. 奥维对象初始化函数.....	40
2.4.3. 奥维对象数据处理函数.....	42
2.4.4. 设置奥维对象函数.....	45
2.4.5. 获取奥维对象函数.....	51
2.4.6. 删除奥维对象函数.....	55
2.4.7. 奥维校验函数	56
2.4.8. 窗体注册函数	56
2.4.9. 消息发送函数	57
2.4.10. 奥维地图视野跟随函数.....	57
2.5. 示例功能介绍.....	58
2.5.1. 取经纬度	58
2.5.2. 设置临时标签	58

2.5.3. 设置地图位置.....	59
2.5.4. 设置照片列表.....	59
2.5.5. 地址解析.....	60
2.5.6. 根据时间取轨迹经纬度.....	61
2.5.7. 取收藏夹对象列表.....	62
2.5.8. 取收藏夹对象.....	64
2.5.9. 设置收藏夹对象	66
2.5.10. 删除收藏夹对象	70
2.5.11. 设置临时对象	70
2.5.12. 清空临时对象	70
2.5.13. 获取奥维端选中的收藏夹对象.....	70
2.5.14. 添加工具条到奥维.....	72
2.5.15. 添加工具栏自定义按钮.....	73
2.5.16. 删除工具栏自定义按钮.....	74
2.5.17. 查询工具栏自定义按钮.....	74
2.5.18. 添加收藏夹对象自定义菜单项.....	74
2.5.19. 删除收藏夹对象自定义菜单项.....	74
2.5.20. 查询收藏夹对象自定义菜单项.....	75
2.5.21. 设置自定义弹窗状态.....	75
2.5.22. 设置弹窗位置和信息.....	76
2.5.23. 添加工具栏自定义菜单.....	77
2.5.24. 工具栏自定义功能性菜单项被点击.....	78
2.5.25. 删除工具栏自定义菜单.....	78
2.5.26. 获取轨迹基本属性.....	79
2.5.27. 获取图形基本信息.....	80
2.5.28. 设置系统坐标系参数.....	81
2.5.29. 获取系统坐标系参数.....	81
2.5.30. 获取奥维可显示区域范围.....	83

2.5.31. 设置非特定对象自定义弹窗.....	83
2.5.32. 查询非特定对象自定义弹窗.....	85
2.5.33. 删除非特定对象自定义弹窗.....	86
2.5.34. 获取奥维自定义地图列表.....	87
2.5.35. 获取奥维 <i>WebSocket</i> 配置信息.....	88

1. 奥维第三方接口文件简介

1.1. DlgToolbox.h

声明了奥维互动地图工具条相关函数，示范用于添加到奥维互动地图中的工具条

1.2. OmapApi.h

声明了奥维互动地图第三方接口中的函数和常量、奥维对象结构体。

1.3. OmapInterface.h

声明了奥维互动地图第三方接口类名

1.4. OmapInterfaceDlg.h

声明了奥维互动地图第三方接口示例程序相关函数

1.5. DlgToolBox.cpp

奥维互动地图中的工具条实现文件。

1.6. OmapApi.cpp

奥维互动地图第三方接口函数实现文件

1.7. OmapInterface.cpp

初始化奥维互动地图第三方接口示例应用程序类名实现文件

1.8. OmapInterfaceDlg.cpp

奥维互动地图第三方接口示例程序示例功能实现文件

2. 详细功能介绍

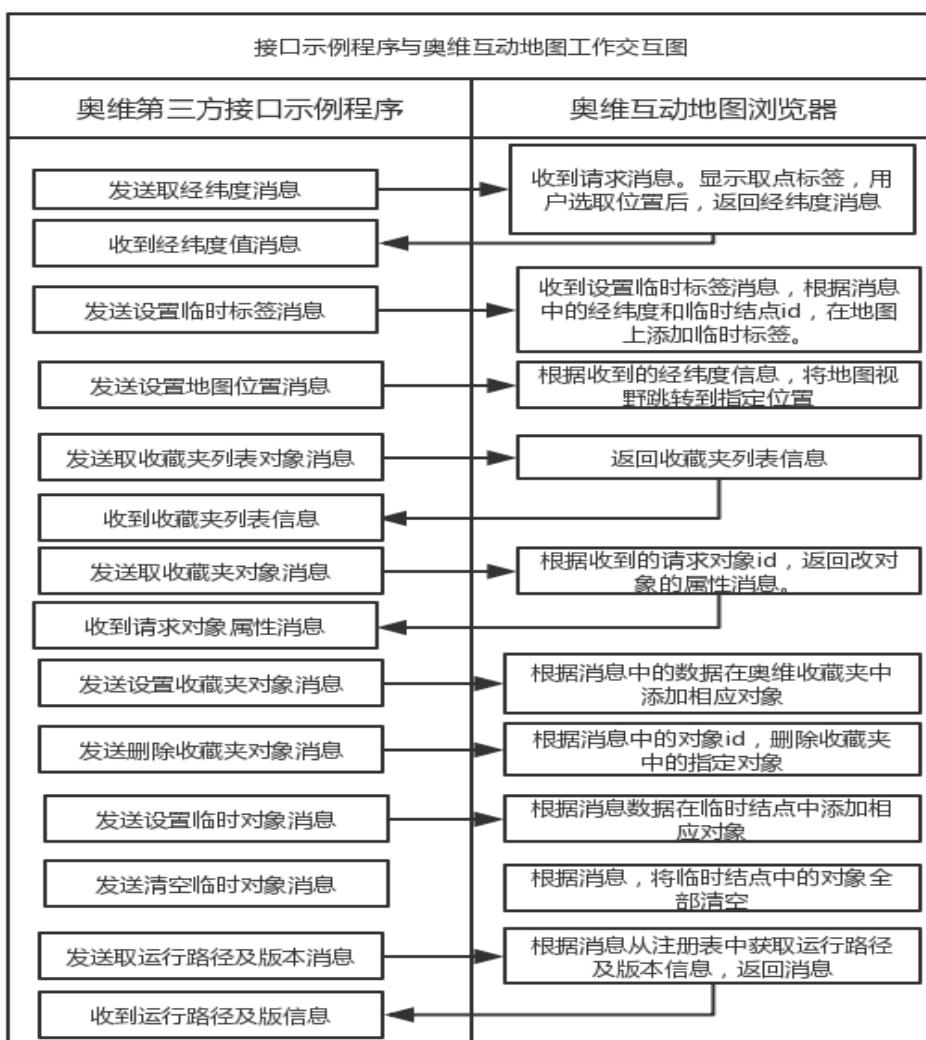
2.1. 接口工作机制

2.1.1. 接口使用 VIP 要求

获取地图图片数据、获取等高线数据功能需要 VIP5 及以上

其他功能至少需要 VIP1

2.1.1. 接口示例程序与奥维互动地图交互图



2.1.2. 消息交互机制

奥维互动浏览器第三方开发接口通过 Windows 消息机制与奥维互动地图进行交互。只有在奥维互动地图打开的情况下，第三方接口才能正常使用。

2.1.2.1. 获取目标窗口句柄

在发送消息之前，需要先获取目标窗口的句柄，否则将无法进行交互。

```
HWND FindWindow(LPCSTR lpClassName,LPCSTR lpWindowName);
```

参数说明：

lpClassName

指向一个以 NULL 字符结尾的、用来指定类名的字符串或一个可以确定类名字符串的原子。如果这个参数是一个原子，那么它必须是一个在调用此函数前已经通过 GlobalAddAtom 函数创建好的全局原子。这个原子（一个 16bit 的值），必须被放置在 lpClassName 的低位字节中，lpClassName 的高位字节置零。

如果该参数为 null 时，将会寻找任何与 lpWindowName 参数匹配的窗口。

lpWindowName

指向一个以 NULL 字符结尾的、用来指定窗口名（即窗口标题）的字符串。如果此参数为 NULL，则匹配所有窗口名。

返回值说明：

如果函数执行成功，则返回值是拥有指定窗口类名或窗口名的窗口的句柄。

如果函数执行失败，则返回值为 NULL 。

例：

奥维互动地图第三方接口向奥维互动地图客户端注册窗体类名

获取类名为omapWin的奥维互动地图窗口句柄，如果奥维地图没有打开，将无法获取窗口句柄。

```
HWND hwnd = ::FindWindow(L"omapWin", NULL);
if (!hwnd)
    return;
```

2.1.2.2. 发送消息

LRESULT SendMessage (HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)

将指定的消息发送到奥维互动地图窗口。此函数为指定的窗口调用窗口程序，直到窗口程序处理完消息再返回。

参数说明：

hWnd: 指定要接收消息的窗口的句柄。如果此参数为 **HWND_BROADCAST**，则消息将被发送到系统中所有顶层窗口，包括无效或不可见的非自身拥有的窗口、被覆盖的窗口和弹出式窗口，但消息不被发送到子窗口。

Msg: 指定被发送的消息。

wParam: 指定附加的消息特定信息。

lParam: 指定附加的消息特定信息。

返回值：返回值指定消息处理的结果，依赖于所发送的消息。

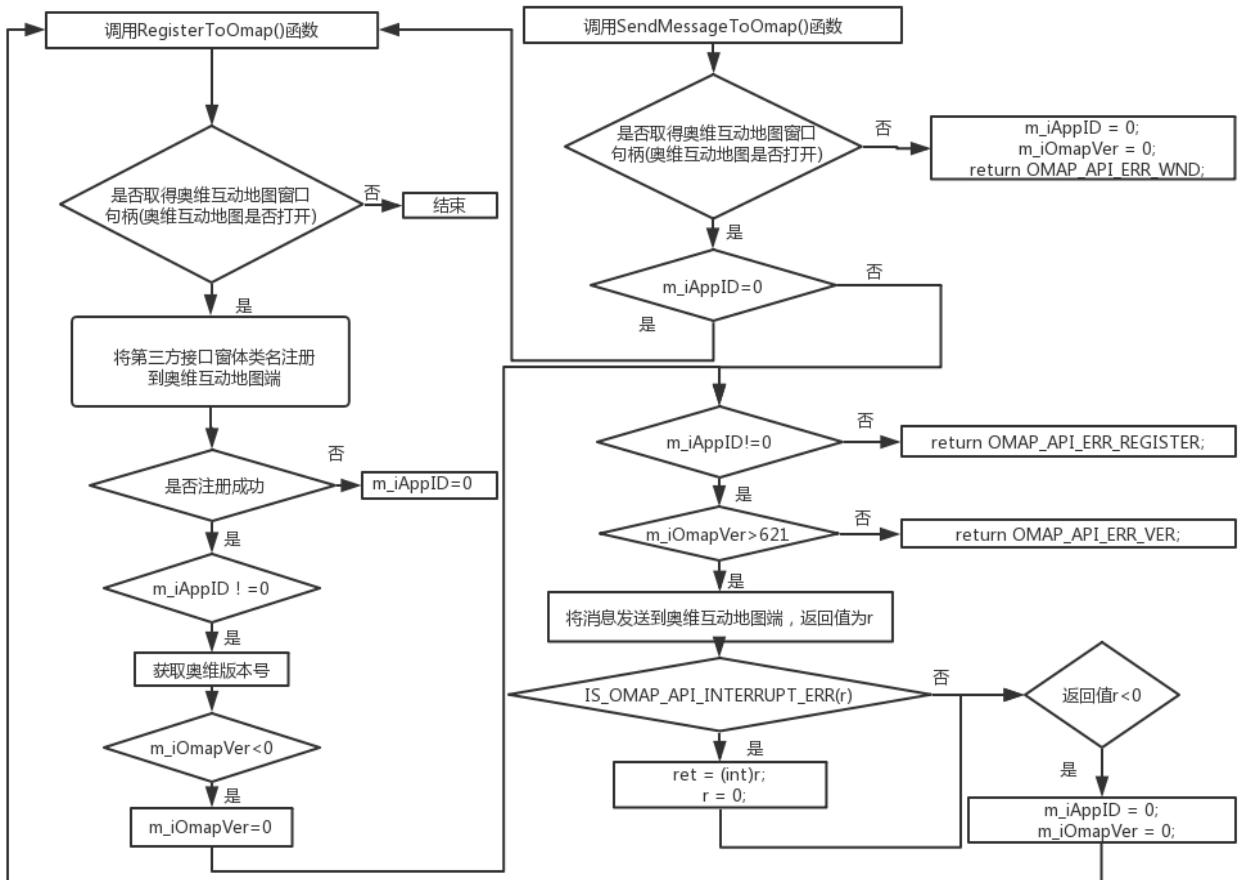
例：

奥维互动地图第三方接口向奥维互动地图客户端注册窗体类名

成功获取到奥维互动地图窗口句柄后，将第三方接口窗体类名发送给奥维互动地图，以便能够接收到奥维互动地图返回的消息。

```
wchar_t* wsClass = (wchar_t*)m_wsAppName;
COPYDATASTRUCT cd1;
memset(&cd1, 0, sizeof(cd1));
cd1.dwData = OMAP_WIN_CMD_REGISTER;
cd1.cbData = (int)wcslen(wsClass) * 2 + 2;
cd1.lpData = wsClass;
m_iAppID = (int)SendMessage(hwnd, WM_COPYDATA, OMAP_WIN_API_FLAG, (LPARAM)&cd1);
if (m_iAppID != 0)
{
    memset(&cd1, 0, sizeof(cd1));
    cd1.dwData = (m_iAppID << 16) | OMAP_WIN_CMD_GET_VERSION;
    m_iOmapVer = (int)SendMessage(hwnd, WM_COPYDATA, OMAP_WIN_API_FLAG,
(LPARAM)&cd1);
    if (m_iOmapVer < 0)
        m_iOmapVer = 0;
}
```

2.1.2.3. 发送消息流程图



2.1.2.4. 接收消息

LRESULT CALLBACK WindowProc (HWND hwnd,UINT uMsg,WPARAM wParam,LPARAM lParam);

处理奥维互动地图端发送给第三方接口窗口的消息。

参数说明：

hwnd：指向窗口的句柄。

uMsg：指定消息类型。

wParam：指定其余的、消息特定的信息。该参数的内容与 uMsg 参数值有关。

lParam：指定其余的、消息特定的信息。该参数的内容与 uMsg 参数值有关。

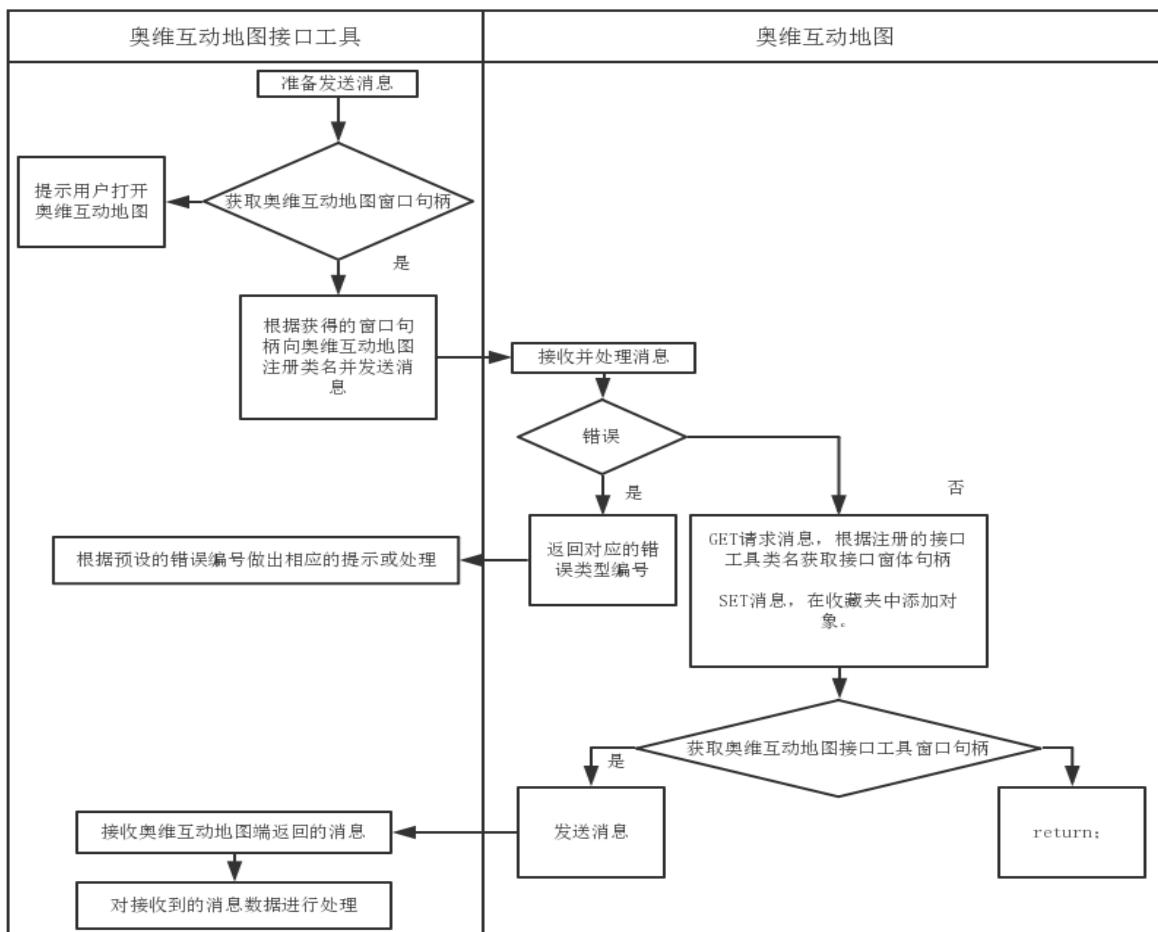
返回值：返回值就是消息处理结果，它与发送的消息有关。

例：

LRESULT COmapInterfaceDlg::WindowProc(UINT message, WPARAM wParam, LPARAM lParam)

奥维互动地图第三方接口接收到奥维互动地图端返回的消息都统一在该函数进行处理。

2.1.2.5. 接收消息流程图



2.2. 常量介绍

```
#define APP_CLASS_NAME L"OMapAPITest" //接口窗体类名
#define OMAP_WIN_CMD_REGISTER 20000 //注册类名
#define OMAP_WIN_CMD_GET_LATLNG 20001//获取经纬度
#define OMAP_WIN_CMD_SET_LATLNG 20002//设置经纬度
#define OMAP_WIN_CMD_LATLNG_GEOCODE 20003//解析经纬度地址
#define OMAP_WIN_CMD_SHOW_PIC 20004//往地图上添加图片
#define OMAP_WIN_CMD_TIME_TO_LATLNG 20005//根据时间取轨迹经纬度
#define OMAP_WIN_CMD_GET_VERSION 20006//获取版本号
#define OMAP_WIN_CMD_SET_MAP_LOCATION 20007//设置地图位置
#define OMAP_WIN_CMD_CLOSE 20008//关闭
#define OMAP_WIN_CMD_SAVE_CFG 20009//保存配置
#define OMAP_WIN_CMD_CLEAN_TMP_OBJ 20010//清除临时对象
#define OMAP_WIN_CMD_SET_MAP_LOCATION_EXT 20011//设置地图位置
#define OMAP_WIN_CMD_GET_MAP_INFO 20012//获取地图
#define OMAP_WIN_CMD_FOLLW_MAP_INFO 20013//地图视野跟随
#define OMAP_API_CMD_GET_OBJ_LIST 20100 //获取收藏夹对象列表
#define OMAP_API_CMD_GET_OBJ 20101//获取收藏夹对象
#define OMAP_API_CMD_SET_OBJ 20102//设置收藏夹对象
#define OMAP_API_CMD_DEL_OBJ 20103//删除收藏夹对象
#define OMAP_API_CMD_GET_SEL_LIST 20104//获取选中的收藏夹对象
#define OMAP_API_CMD_SRH_OBJ 20105//搜索收藏夹对象
#define OMAP_API_CMD_SET_SEL_ID_LIST 20106//根据 id 选中收藏夹对象
#define OMAP_API_CMD_SRH_OBJ_EXT 20107;//对象检索
#define OMAP_API_CMD_GET_SEL_OBJ_TREE 20110 //获取选中的收藏夹对象树形结构
// 返回 tagOApiObjDataItem 数组，会自动填充(修正)idParent 与 idObj
#define OMAP_API_CMD_SET_OBJ_TREE 20112 //
#define OMAP_API_CMD_SET_SYS_COOR_POINT 20201 //设置关联点
#define OMAP_API_CMD_GET_SYS_COOR_INFO 20202 //获取关联点
#define OMAP_API_CMD_REQ_INTERFACE_VER 20203 //获取接口版本号
#define OMAP_API_CMD_GET_SYS_COOR_CENTER 20211 //获取屏幕中心点
#define OMAP_API_CMD_GET_COOR_PIXEL_INFO 20212
#define OMAP_API_CMD_SET_COORD_PROJ 20221 //设置系统坐标系参数
#define OMAP_API_CMD_GET_COORD_PROJ 20222//获取系统坐标系参数
#define OMAP_API_CMD_GET_MAP_DATA 20301 //获取地图数据
#define OMAP_API_CMD_GET_MAP_FILE 20302 //获取地图图片文件
#define OMAP_API_CMD_GET_CONTOUR 20303 //获取地图等高线
#define OMAP_API_CMD_SET_BTN 20401 //设置按钮
#define OMAP_API_CMD_CLICK_BTN 20402 //按下按钮
#define OMAP_API_CMD_GET_STATUS_INFO 20501
#define OMAP_AIP_CMD_REQ_VIP5_DEMO 20503
```

```
#define OMAP_API_CMD_GET_SHAPE_FILL_PAT_LIST 20601 //获取图案填充类型列表
#define OMAP_API_CMD_CUS_MAIN_BTN_QUERY 20701 //工具栏自定义按钮查询
#define OMAP_API_CMD_CUS_MAIN_BTN_SET 20702 // 添加或者修改工具栏自定义按钮
#define OMAP_API_CMD_CUS_MAIN_BTN_DEL 20703 //删除工具栏自定义按钮
#define OMAP_API_CMD_CUS_MAIN_BTN_CLICK 20704 //工具栏自定义按钮被点击

#define OMAP_API_CMD_CUS_FAV_MENU_QUERY 20711 //收藏夹自定义菜单项查询
#define OMAP_API_CMD_CUS_FAV_MENU_SET 20712 // 添加或者修改收藏夹自定义菜单项
#define OMAP_API_CMD_CUS_FAV_MENU_DEL 20713 //删除收藏夹自定义菜单项
#define OMAP_API_CMD_CUS_FAV_MENU_CLICK 20714 //收藏夹自定义菜单项被点击

#define OMAP_API_CMD_OBJ_EVENT_QUERY 20721 //查询当前点击监听类型
#define OMAP_API_CMD_OBJ_EVENT_SET 20722 //设置当前点击事件监听类型
#define OMAP_API_CMD_OBJ_EVENT_TIRGGER 20723 //接收奥维窗体可显示区域数据及被点击对象坐标

#define OMAP_API_CMD_GET_OMAP_WND_RECT 20730 //获取奥维窗体可显示区域
#define OMAP_API_CMD_OBJ_SHOW_WND 20731 //显示奥维自定义弹窗
#define OMAP_API_CMD_OBJ_CLOSE_WND_EVENT 20732 //自定义弹窗关闭事件
#define OMAP_API_CMD_CUS_MAIN_MENU_QUERY 20741
#define OMAP_API_CMD_CUS_MAIN_MENU_SET 20742 // 添加或者修改工具栏自定义菜单项
#define OMAP_API_CMD_CUS_MAIN_MENU_CLICK 20744 //工具栏自定义菜单项被点击

#define OMAP_API_CMD_CUS_WND_SET 20750 //添加或修改非特定对象弹窗
#define OMAP_API_CMD_CUS_WND_QUERY 20751 //查询非特定对象弹窗
#define OMAP_API_CMD_CUS_WND_DEL 20752 //删除非特定对象弹窗
#define OMAP_API_CMD_CUS_WND_CLOSE_EVENT 20753 //非特定对象弹窗被点击
#define OMAP_API_CMD_GET_WEB_SOCKET_CONF 20850 //获取奥维 websocket 配置
#define OMAP_API_CMD_GET_CUS_MAP_CFG 20860 //获取奥维自定义地图列表

#define GEO_DECODE_ERR_WAIT -1 //解析地址等待
#define GEO_DECODE_ERR_NO_DATA -2 //解析错误
#define MAX_API_NAME_LEN 100 //奥维对象名称长度
#define MAX_API_DIR_WAYPOINTS 8 //中间导航点的经纬度最大数量
#define OMAP_OBJ_TYPE_GROUP 30 //文件夹对象
#define OMAP_OBJ_TYPE_SIGN 7 //标签对象
#define OMAP_OBJ_TYPE_TRACK 8 //轨迹对象
#define OMAP_OBJ_TYPE_SHAPE 13 //图形对象
#define OMAP_OBJ_TYPE_DIR 11 //导航线路对象
#define OMAP_API_ERR_OK 1
#define OMAP_API_ERR_WND -1 //没有找到奥维窗口
#define OMAP_API_ERR_REGISTER -2 //向奥维地图程序注册失败
#define OMAP_API_ERR_VER -3 //奥维地图版本太低
```

```
#define OMAP_API_ERR_ARGV -4 //参数非法
#define OMAP_API_ERR_NOT_VIP -5 //当前绑定用户不是 VIP
#define OMAP_API_ERR DEAL -6 //消息处理失败
#define OMAP_API_ERR_NOT_VIP5 -7 //当前绑定用户不是 VIP5 及以上用户
#define OMAP_API_ERR_UNLOAD_OBJ -8 //奥维对象未载入, 请稍等
#define OMAP_API_ERR_CMD -9 //
#define IS_OMAP_API_INTERRUPT_ERR(x) (x == OMAP_API_ERR_NOT_VIP || x ==
OMAP_API_ERR DEAL || x == OMAP_API_ERR_NOT_VIP5 || x == OMAP_API_ERR_UNLOAD_OBJ)
#define OAPI_APP_ID_QUERY -100 //查询当前值
#define OAPI_APP_ID_ALL -1 //奥维发送所有注册 AppID
#define OAPI_APP_ID_NONE 0 //取消对象监控
// 所有对象
#define OBJ_SELECT_CHECK_FLAG_ALL 0
// 选中对象, 选中的文件夹自动包含所有子对象
#define OBJ_SELECT_CHECK_FLAG_GROUP_JOIN 1
// 选中对象, 选中的文件夹不自动包括非选中子对象
#define OBJ_SELECT_CHECK_FLAG_SEL_ONLY 2
// 选中对象, 如果选中的是文件夹, 则排除文件夹本身去选中子结点
#define OBJ_SELECT_CHECK_FLAG_NO_GROUP 3

//临时对象分组节点 ID
#define OMAP_TMP_OBJ_GROUP_ID 202
#define OAPI_RET_ERR_VIP_LEVEL -1 //奥维 VIP 级别不够
#define OAPI_RET_ERR_VIP5_NEED -2 //需要 VIP5 及以上
#define OAPI_RET_ERR_LATLNG_INVALID -3 //坐标系错误
#define OAPI_RET_ERR_ARGV_ERR -4 //参数错误
#define OAPI_RET_ERR_MEMORY -5 //内存异常
#define OAPI_RET_ERR_UNKNOWN -6 //未知错误
#define OAPI_RET_ERR_ARGV_END -7 //与等高线请求相关

#define GET_MAP_FILE_RET_FULL_GET 1 //全部数据
#define GET_MAP_FILE_RET_PART_GET 2 //部分数据
#define GET_MAP_FILE_RET_FULL_NO 3 //没有数据
#define GET_MAP_FILE_RET_ERR_COORD_TYPE -101 //坐标系设置错误(或位置偏差过大)
#define GET_MAP_FILE_RET_ERR_IGNORE -102 //忽略
#define GET_MAP_FILE_RET_ERR_SIZE_EXCEED -103 //地图尺寸太大
#define GET_MAP_FILE_RET_ERR_PIXEL_EXCEED -104 //地图像素太高
#define GET_MAP_FILE_RET_ERR_CREATE_IMG -105 //创建地图失败
#define GET_MAP_FILE_RET_ERR_RESIZE -106 //地图扭曲太大

#define GET_MAP_FILE_IMG_TYPE_PNG 0 //地图数据为 png 格式
#define GET_MAP_FILE_IMG_TYPE_JPG 1 //地图数据为 jpg 格式
```

```
#define OMAP_ALIGN_8_BYTE(x) ((x) % 8 == 0) ? x : ((x) / 8 * 8 + 8)
#define SRH_OBJ_OPT_NAME_FUZZY      0 //名称模糊查找
#define SRH_OBJ_OPT_NAME_HEADER    1 //名称头字符
#define SRH_OBJ_OPT_COMMENT_FUZZY   2 //备注模糊查找
#define SRH_OBJ_OPT_COMMENT_HEADER  3 //备注头字符
#define SRH_OBJ_OPT_SRH_SAME       4
#define SRH_OBJ_OPT_NAME_MATCH     5 //名称完全匹配
#define SRH_OBJ_OPT_COMMENT_MATCH  6 //备注完全匹配

#define SRH_OBJ_OPT_SIGN_SHOW_STYLE 11 //标签显示样式
#define SRH_OBJ_OPT_SIGN_ICON_ID    12 //标签图标 id

#define SRH_OBJ_OPT_TRACK_LINET_TYPE 21 //轨迹线型
#define SRH_OBJ_OPT_TRACK_SHOW_STYLE 22 // 轨迹风格
#define SRH_OBJ_OPT_TRACK_CLASS0    23 //轨迹类别

#define IS_TXT_SRH_OBJ_OPT(x) (x==SRH_OBJ_OPT_NAME_FUZZY||x==
SRH_OBJ_OPT_NAME_HEADER||x==SRH_OBJ_OPT_COMMENT_FUZZY||x==
SRH_OBJ_OPT_COMMENT_HEADER||x==SRH_OBJ_OPT_NAME_MATCH||x==
SRH_OBJ_OPT_COMMENT_MATCH)

#define IS_INT_SRH_OBJ_OPT(x) (x==SRH_OBJ_OPT_SIGN_SHOW_STYLE||x==
SRH_OBJ_OPT_SIGN_ICON_ID||x==SRH_OBJ_OPT_TRACK_LINET_TYPE||x==
SRH_OBJ_OPT_TRACK_SHOW_STYLE || x == SRH_OBJ_OPT_TRACK_CLASS0)

#define IS_SAME_SRH_OBJ_OPT(x) (x == SRH_OBJ_OPT_SRH_SAME)
```

2.3. 结构体介绍

2.3.1. tagOmapLatlng

名称		tagOmapLatlng	
功能介绍		经纬度结构体	
参数		类型	
参数	lat	double	纬度
	lng	double	经度
	bReal	BYTE	0: 偏移经纬度 1: 真实经纬度 2: 平面坐标

2.3.2. tagOmapLocation

名称		tagOmapLocation	
功能介绍		奥维视图位置结构体	
参数		类型	
参数	ll	tagOmapLatlng	经纬度
	iMapID	int	地图类型 ID
	iMapLevel	int	地图级别

2.3.3. tagOmapLocationExt

名称		tagOmapLocationExt	
功能介绍		奥维视图位置、奥维窗体显示位置结构体	
参数		类型	
参数	ol	tagOmapLocation	奥维视图位置结构体
	bActive	BYTE	是否将奥维置为最屏幕最前端

2.3.4. tagWinMapInfo

名称		tagWinMapInfo	
功能介绍		奥维屏幕中心点位置结构体	
参数		类型	
参数	lng	double	经度
	lat	double	纬度
	bReal	BYTE	0:gcj02 经纬度 1:真实经纬度, 2:当前坐标系
	x_lng	double	平面坐标 X
	y_lat	double	平面坐标 Y
	iMapLevel	int	地图缩放级别

	iMapType	int	地图类型
	iMapX	int	奥维 X 坐标
	iMapY	int	奥维坐标 Y
	iRet	int	返回值
	ret[10]	DWORD	预留字段

2.3.5. tagOPhotoMapPic

名称 tagOPhotoMapPic		
功能介绍 图片结构体		
参数 类型 备注		
参数	idPic	INT64 照片 id
	tmPhoto	unsigned int 照片拍摄时间
	lat	double 纬度
	lng	double 经度
	iAction	int

2.3.6. tagOmapGeoDecode

名称 tagOmapGeoDecode		
功能介绍 地图解析结构体		
参数 类型 备注		
参数	oll	tagOmapLatlng 经纬度
	nLevel	int 地图解析的层级 如 省市区 3 级、省市县 3 级、市区 2 级
	strArea[8][128]	char 解析后地址

2.3.7. tagOmapGetMap

名称 tagOmapGetMap		
功能介绍 奥维地图图片数据请求结构体		
参数 类型 备注		
参数	x_lng1	double 请求区域左下角位置 x 坐标或经度
	y_lat1	double 请求区域左下角位置 y 坐标或纬度
	x_lng2	double 请求区域右上角位置 x 坐标或经度
	y_lat2	double 请求区域右上角位置 y 坐标或纬度
	bRealLl	BYTE 0:gcj02 经纬度 1:真实经纬度, 2:当前坐标系
	iMaxPixel	INT64 需要的图片最大像素
	iMaxLevel	int 地图缩放级别
	width	DWORD 需要的图片宽度
	height	DWORD 需要的图片高度
	ret	int 预留字段

2.3.8. tagOmapGetMapFile

名称	tagOmapGetMapFile	
功能介绍	奥维地图图片文件结构体	
参数	类型	备注
参数	idMap	INT64 该请求的编号，多次请求时相同编号新的会替换旧的
	iValue	INT64 重要程度
	iFlag	int 1 表示图不完全时直接忽略
	iRet	int 返回值
	strFile[256]	char 文件名为空时或者写入失败时将通过 pData 返回
	bFile	bool 表明写入状态
	pDate	void* 图片数据
	nData	int 数据长度
	x_lng1	double 请求区域左下角位置 x 坐标或经度
	y_lat1	double 请求区域左下角位置 y 坐标或纬度
	x_lng2	double 请求区域右上角位置 x 坐标或经度
	y_lat2	double 请求区域右上角位置 y 坐标或纬度
	bRealL1	BYTE 0:gcj02 经纬度 1:真实经纬度, 2:当前坐标系
	iMapLevel	int 地图级别
	iMapType	int 地图类型
	width	DWORD 需要的图片宽度
	height	DWORD 需要的图片高度
	iTotalPic	int
	iHasPic	int
	iPostCnt	int 取图过程每取多少张发送一次 g_msgGetMapProcess 消息 (WPARAM=iTotal, PARAM=iDeal);
	iTmp	int 奥维使用的临时变量
	bSendReq	BYTE 1 表示图不存在或者不完全时发送网络请求
	bImageType	BYTE 需要的图片类型 0:PNG 1:JPG
	bRetUse[3]	BYTE
	idApp	int
	iRetUse[18]	int

2.3.9. tagOmapReqContour

名称		tagOmapReqContour	
功能介绍		奥维等高线数据结构体	
参数		类型	备注
参数	idMap	INT64	该请求的编号, 多次请求时相同编号新的会替换旧的
	iValue	INT64	重要程度
	x_lng1	double	请求区域左下角位置 x 坐标或经度
	y_lat1	double	请求区域左下角位置 y 坐标或纬度
	x_lng2	double	请求区域右上角位置 x 坐标或经度
	y_lat2	double	请求区域右上角位置 y 坐标或纬度
	bRealL1	BYTE	0:gcj02 经纬度 1:真实经纬度, 2:当前坐标系
	iElevValue	INT64	等高线间隔
	iMapLevel	int	地图请求级别
	iCheckFlag	int	0: 表示直接取, 1: 表示只做检测行为
	iCheckResult	int	
	iRet	int	返回值
	iNeedVip	int	所需要的 VIP 级别
	bPart	BYTE	
	pData	void*	等高线数据
	nData	int	数据长度
	hWnd	void*	HWND hWnd
	iPostCnt	int	取等高线过程每取多少次间隔发送一下 g_msgGetContourProcess 消息(WPARAM=iTotal, PARAM=iDeal);
	idApp	int	用户自定义
	iRet1[19]	int	预留

2.3.10. tagOMapBtnInfo

名称		tagOMapBtnInfo	
功能介绍		自定义按钮结构体	
参数		类型	备注
参数	ret0	int	
	iAdd	int	按钮 ID
	strTip[256]	char	按钮名称
	ret1[10]	int	预留字段

2.3.11. tagOApiObjectOptHdr

名称		tagOApiObjectOptHdr	
功能介绍		存放收藏夹对象的结构体	
参数		类型	备注
参数	iVersion	WORD	奥维版本
	iType	BYTE	对象类型：文件夹、标签、轨迹、图形、路线
	bLlType	BYTE	坐标类型
	idParent	DWORD	收藏夹对象的父节点的 id 0 表示该对象为根节点
	idObj	DWORD	收藏夹对象的 id
	ret	int	节点所在位置

2.3.12. tagOApiItemListV1

名称		tagOApiItemListV1	
功能介绍		收藏夹对象列表	
参数		类型	备注
参数	idObj	tagOmapLatlng	经纬度
	iType	int	地图解析的层级 如 省市区 3 级、省市县 3 级、市区 2 级
	strName[MAX_API_NAME_LEN]	char	对象名称

2.3.13. tagOApiObjGroupV1

名称	tagOApObjGroupV1	
功能介绍	表示文件夹对象的结构体	
参数	类型	备注
参数	strName[MAX_API_NAME_LEN]	char 对象名称
	nChild	下一级子项的个数
	nAllChild	所有子项及其下属的总个数
	bLoadOk	是否加载成功
	bSaveMerge	存盘时 pChild 的内容是否合并存储，合并存储可大大加快载入时间，但使修改时存盘时间下降
	bGroup	是否组合，不可展开
	bAutoLoad	是否自动加载
	iShowLevel	是否隐藏
	iShowLevelMax	显示级别
	bCrypt	是否加密，如果设置了加密则自动加载将自动失效，接口可以获取该值但无法设置修改
	bReadOnly	是否只读 0: 不锁定, 1: 不带密码锁定 2: 带密码锁定
	bNotHotId	是否不响应鼠标事件
	bRelate	是否是关联对象 (目前用于标签与轨迹关联)
	pwd[16]	密码,此参数已禁用无法获取与修改
	ret[10]	保留字段

2.3.14. tagOApiObjGroupV2

名称	tagOApiObjGroupV2	
功能介绍	表示文件夹对象的结构体	
参数	类型	备注
参数	strName[MAX_API_NAME_LEN]	char 对象名称
	nChild	下一级子项的个数
	nAllChild	所有子项及其下属的总个数
	bLoadOk	是否加载成功
	bSaveMerge	存盘时 pChild 的内容是否合并存储，合并存储可大大加快载入时间，但使修改时存盘时间下降
	bGroup	是否组合，不可展开
	bAutoLoad	是否自动加载
	iShowLevel	是否隐藏
	iShowLevelMax	显示级别
	bShare	0 不分享，1 只读分享，2 读写分享
	bCrypt	是否加密，如果设置了加密则自动加载将自动失效，接口可以获取该值但无法设置修改
	bReadOnly	是否只读 0: 不锁定, 1: 不带密码锁定 2: 带密码锁定
	bNotHotId	是否不响应鼠标事件
	bRelate	是否是关联对象 (目前用于标签与轨迹关联)
	dwBind	是否绑定到当前设备，>0 为绑定且值为签名
	dwBindCheck	解绑设备时验证，0 为不可解绑，>0 则输入密码解绑
	bLink	是否是链接到其他用户的文件夹
	bLinkAutoCheck	是否自动检查链接是否有更新
	biChildOverlay	用位表示下属对象存在哪些叠放层次，该值在收藏夹有发生变动时自动刷新
	ret[10]	保留字段

2.3.15. tagOApiMapSignV1

名称	tagOApiMapSignV1		
功能介绍	表示标签的结构体		
参数	类型	备注	
参数	strName[MAX_API_NAME_LEN]	char	标签名称
	pstrComment	char*	标签备注
	lat	double	纬度
	lng	BYTE	经度
	bRealL1	BYTE	0:gcj02 经纬度 1:真实经纬度, 2:当前坐标系
	iAltitude	int	海拔
	iTime	int	生成时间
	bEditMode	BYTE	是否可编辑
	iOverlayIdx	BYTE	叠放层级
	iSignAlignFlag	BYTE	标签图标对齐标志, 0 为默认下中对齐 1 为正中对齐
	iSignPic	int	标签图标, 具体内容参考标签图标库 0:no picture >=1 has sign picture
	dwSignClr	DWORD	标签颜色
	iPicScale	BYTE	0:default >=1 32*iPicScale/10
	iTxtType	BYTE	0:不显示名称 1:显示名称 2:显示带框名称 3:自定义名称显示风格
	iShowLevel	BYTE	标签显示级别
	iShowLevelMax	BYTE	标签显示最大级别

2.3.16. tagOApiMapSignV2

名称	tagOApiMapSignV2	
功能介绍	表示标签的结构体	
参数	类型	备注
strName[MAX_API_NAME_LEN]	char	标签名称
pstrComment	char*	标签备注
lat	double	纬度
lng	BYTE	经度
bRealLl	BYTE	0:gcj02 经纬度 1:真实经纬度, 2:当前坐标系
iAltitude	int	海拔
iTime	int	生成时间
bEditMode	BYTE	是否可编辑
iOverlayIdx	BYTE	叠放层级
iTxtShowSta	BYTE	高 5 位为级别, 低 3 位为位置, 分别是 0 为正常位置, 1 左下, 2 右上, 3 左上, 4 上, 5 下, >6 不显示
iTxtShowStaSet	BYTE	设置位置项, 0 为默认, 可自动调整位置, 1-6 分别对应 iTxtShowSta 的 0-5
iSignAlignFlag	BYTE	标签图标对齐标志, 0 为默认下中对齐 1 为正中对齐
iSignPic	int	标签图标, 具体内容参考标签图标库 0:no picture >=1 has sign picture
dwSignClr	DWORD	标签颜色
iPicScale	BYTE	0:default >=1 32*iPicScale/10
iSignPicNumOffx		0:default, 1-63 右移, 65-127 左移
iSignPicNumOffy		0:default, 1-63 右移, 65-127 左移
iSignPicNumClr		8 种颜色, 0: (default), 1: 黑, 2: 白, 3: 红, 4: 绿, 5: 蓝, 6: 黄, 7: 绿蓝, 8: 红蓝
iSignPicNumSize		0:default, 1-8(加大 1-16 像素), 65-80(减少 1-16 像素)
iTxtType	BYTE	0:不显示名称 1:显示名称 2:显示带框名称 3:自定义名称显示风格
iShowLevel	BYTE	标签显示级别(可用于隐藏标签)

	iShowLevelBak	BYTE	用于隐藏与显示时恢复 iShowLevel 值
	iShowLevelMax	BYTE	标签显示最大级别
	bTimeUncertain	BYTE	iTime 的时间是否可信

2.3.17. tagOApiMapTrackPointV1

名称		tagOApiMapTrackPointV1	
功能介绍		轨迹上的点的信息	
参数		类型	备注
参数	lat	double	纬度
	lng	double	经度
	time	int	轨迹点的时间
	iAltitude	int	iAltitude >> 16 == 1 表示该点为新子图形(新子轨迹)的起始点即前一个点间断

2.3.18. tagOApiMapTrackV1

名称		tagOApiMapTrackV1	
功能介绍		表示轨迹的结构体	
参数		类型	备注
参数	strName[MAX_API_NAME_LEN]	char	轨迹名称
	pstrComment	char*	轨迹备注
	pMtp	tagOApiMapTrackPointV1*	轨迹上的点的信息
	nMtp	int	轨迹点数量
	bRealL1	BYTE	pMtp 中经纬度是否真实经纬度
	iTrackType	BYTE	轨迹类别: 自驾, 爬山 ...
	iOverlayIdx	BYTE	叠放层级
	iShowLevel	BYTE	显示级别
	iShowLevelMax	BYTE	最大显示级别
	bShowName	BYTE	是否显示轨迹名称
	bEdit	BYTE	轨迹是否可编辑

2.3.19. tagOApiMapTrackDraw

名称		tagOApiMapTrackDraw	
功能介绍		轨迹绘制的相关属性的结构体	
参数		类型	备注
参数	dwLineClr	DWORD	线条颜色
	iLineWidth	int	线宽
	iLineAlpha	BYTE	线条透明度
	iShowType	BYTE	轨迹风格: 显示加权距离, CAD 多段线 ...
	iLineType	BYTE	线型
	bNeedDrawDetail	BYTE	是否需要绘制角度、距离提示信息
	bClose	BYTE	是否闭合

2.3.20. tagOApiMapTrackV2

名称	tagOApiMapTrackV2		
功能介绍	表示轨迹的结构体		
参数	类型	备注	
参数	strName[MAX_API_NAME_LEN]	char	轨迹名称
	pstrComment	char*	轨迹备注
	pMtp	tagOApiMapTrackPointV1*	轨迹上的点的信息
	nMtp	int	轨迹点数量
	bRealL1	BYTE	pMtp 中经纬度是否真实经纬度
	iTrackType	BYTE	轨迹类别: 自驾, 爬山 ...
	iOverlayIdx	BYTE	叠放层级
	iShowLevel	BYTE	显示级别(可用于隐藏对象)
	iShowLevelBak	BYTE	用于隐藏与显示时恢 iShowLevel 值
	iShowLevelMax	BYTE	最大显示级别
	bShowName	BYTE	是否显示轨迹名称
	bEdit	BYTE	轨迹是否可编辑
	dwAltitudeColor	DWORD	海拔线颜色
	iCustomType1	BYTE	自定义字段用于保存轨迹的自定义属性
	iCustomType2	BYTE	
	iCustomType3	BYTE	
	iCustomType4	BYTE	
	dwCustom1	int	
	dwCustom2	int	
	tagMapTrackDraw	struct	详情查看 2.3.14

2.3.21. tagOApiMapShapeV1

名称	tagOApiMapShapeV1		
功能介绍	表示图形的结构体		
参数	类型	备注	
参数	strName[MAX_API_NAME_LEN]	char	图形名称
	pstrComment	char*	图形备注
	pMtp	tagOApiMapTrackPointV1*	图形的点信息
	nMtp	int	图形的节点数
	bRealL1	BYTE	0:gcj02 经纬度 1:真实经纬度, 2:当前坐标系
	iShowFlag	BYTE	0:不显示属性 1:显示名称 2:显示面积 3:显示名字与面积
	iOverlayIdx	BYTE	叠放层级
	dStartAngle	double	起始点
	dEndAngle	double	终止点 当图形为两个点时,通过起始角度与终止角度表示圆、扇形、矩形
	iLineWidth	BYTE	线宽
	iLineAlpha	BYTE	线条透明度
	iAreaAlpha	BYTE	区域透明度
	bEditMode	BYTE	是否可编辑
	iShowLevel	BYTE	显示级别
	iShowLevelMax	BYTE	显示最大级别
	iOuterRgnIdx	WORD	外部 Rgn 索引 0 为全部都是外部
	dwLineClr	DWORD	线条颜色
	dwAreaClr	DWORD	填充颜色(渐变色起始颜色)
	iFillType	int	0: 默认颜色填充 1: 渐变色填充 2: 图案填充 3: CAD 图案填充
	iFillSubType	int	渐变色填充时 0: 从左到右渐变 1: 从左到中间渐变再到右还原 2: 从外围到中心渐变 CAD 图案时 为是否随图缩放
	dwFillPattern	DWORD	填充图案 ID

	<code>dwFillChgClr</code>	<code>DWORD</code>	渐变色 2 (渐变色终止颜色), CAD 填充背景色 如果透明: 0x00ffffff 如果有颜色: 0x01ffffff
	<code>fFillRotate</code>	<code>float</code>	
	<code>fFillScale</code>	<code>float</code>	用于 CAD 图案填充
	<code>fFillOffsetX</code>	<code>double</code>	存储填充点的绝对值 X
	<code>fFillOffsetY</code>	<code>double</code>	存储填充点的绝对值 Y

2.3.22. tagOApiMapShapeV2

名称	tagOApiMapShapeV2		
功能介绍	表示图形的结构体		
参数	类型	备注	
参数	strName[MAX_API_NAME_LEN]	char	图形名称
	pstrComment	char*	图形备注
	pMtp	tagOApiMapTrackPointV1*	图形的点信息
	nMtp	int	图形的节点数
	bRealLl	BYTE	0:gcj02 经纬度 1:真实经纬度, 2:当前坐标系
	iShowFlag	BYTE	0:不显示属性 1:显示名称 2:显示面积 3:显示名字与面积
	iOverlayIdx	BYTE	叠放层级
	dStartAngle	double	起始点
	dEndAngle	double	终止点 当图形为两个点时,通过起始角度与终止角度表示圆、扇形、矩形
	iLineWidth	BYTE	线宽
	iLineAlpha	BYTE	线条透明度
	iAreaAlpha	BYTE	区域透明度
	bEditMode	BYTE	是否可编辑
	iShowLevel	BYTE	显示级别(可用于隐藏对象)
	iShowLevelBak	BYTE	用于取消隐藏时恢复 iShowLevel
	iShowLevelMax	BYTE	显示最大级别
	iOuterRgnIdx	WORD	外部 Rgn 索引, 0 为全部都是外部, >0 为前面多少个是外部, 后面是内部, 用于处理镂空
	dwLineClr	DWORD	线条颜色
	dwAreaClr	DWORD	填充颜色(渐变色起始颜色)
	iFillType	int	0: 默认颜色填充 1: 渐变色填充 2: 图案填充 3: CAD 图案填充
	iFillSubType	int	渐变色填充时 0: 从左到右渐变 1: 从左到中间渐变再到右还原 2: 从外围到中心渐变 CAD 图案时

		为是否随图缩放
dwFillPattern	DWORD	填充图案 ID
dwFillChgClr	DWORD	渐变色 2 (渐变色终止颜色), CAD 填充背景色 如果透明: 0x00ffffff 如果有颜色: 0x01ffffff
fFillRotate	float	
fFillScale	float	用于 CAD 图案填充
fFillOffsetX	double	存储填充点相对于中心点的经 纬度偏移量值
fFillOffsetY	double	

2.3.23. tagOApiLatLngV1

名称		tagOApiLatLngV1	
功能介绍		经纬度结构体	
参数		类型	备注
参 数	lat	double	纬度
	lng	double	经度

2.3.24. tagOApiMapDirectionsUnitV1

名称		tagOApiMapDirectionsUnitV1	
功能介绍		表示一个导航步骤的结构体	
参数		类型	备注
参 数	strTitle[128]	WORD	主步骤
	strSubTitle[10][128]	BYTE	子步骤
	nSubCount	BYTE	子步骤数
	iType	DWORD	交通工具类型
	nLl	DWORD	llList 个数
	llList	tagOApiLatLngV1*	经纬度结构体

2.3.25. tagOApiMapDirectionsV1

名称	tagOApiMapDirectionsV1		
功能介绍	表示导航路线的结构体		
参数	类型	备注	
参数	strName[MAX_API_NAME_LEN]	char	导航路线名称
	pstrComment	char*	导航路线备注
	punit	tagOApiMapDirectionsUnitV1*	一个导航步骤的结构体
	nUnit	int	导航步骤数
	strTitle[128]	char	导航步骤名称
	bRealLl	BYTE	0:gcj02 经纬度 1:真实经纬度, 2:当前坐标系
	iOverlayIdx	BYTE	叠放层级
	mpStart	tagOApiLatLngV1	出发点的经纬度
	mpEnd	tagOApiLatLngV1	终点的经纬度
	mpVia[MAX_API_DIR_WAYPOINTS]	tagOApiLatLngV1	中间导航点的经纬度
	iDirMode	int	
	iLineWidth	BYTE	线宽
	iLineAlpha	BYTE	线条透明度
	dwLineClr	DWORD	线条颜色
	bEditMode	BYTE	是否可编辑
	iShowLevel	BYTE	显示级别
	iShowLevelMax	BYTE	显示最大级别

2.3.26. tagOApiMapDirectionsV2

名称	tagOApiMapDirectionsV2		
功能介绍	表示导航路线的结构体		
参数	类型	备注	
参数	strName[MAX_API_NAME_LEN]	char	导航路线名称
	pstrComment	char*	导航路线备注
	punit	tagOApiMapDirectionsUnitV1*	一个导航步骤的结构体
	nUnit	int	导航步骤数
	strTitle[128]	char	导航步骤名称
	strDirStart[64]	char	起始段名称
	strDirEnd[64]	char	终止端名称
	bRealLl	BYTE	0:gcj02 经纬度 1:真实经纬度, 2:当前坐标系
	iOverlayIdx	BYTE	叠放层级
	mpStart	tagOApiLatLngV1	出发点的经纬度
	mpEnd	tagOApiLatLngV1	终点的经纬度
	mpVia[MAX_API_DIR_WAYPOINTS]	tagOApiLatLngV1	中间导航点的经纬度
	iDirMode	int	
	iLineWidth	BYTE	线宽
	iLineAlpha	BYTE	线条透明度
	dwLineClr	DWORD	线条颜色
	bEditMode	BYTE	是否可编辑
	iShowLevel	BYTE	显示级别(可用于隐藏对象)
	iShowLevelBak	BYTE	取消隐藏恢复 iShowLevel
	iShowLevelMax	BYTE	显示最大级别

2.3.27. tagOApiInterfaceVer

名称		tagOApiInterfaceVer	
功能介绍		接口版本结构体	
参数	参数	类型	备注
	iType	int	请求类型
	iVer	int	版本号
	iBuf	int	数据长度
	pBuf	char *	数据内容
	iSendReq	int	请求编号

2.3.28. tagOApiMemBuf

名称		tagOApiMemBuf	
功能介绍		数据结构体	
参数	参数	类型	备注
	pBuf	char *	数据
	nBuf	int	数据长度
	nMem	int	总长度

2.3.29. tagOApiObjDataItem

名称		tagOApiObjDataItem	
功能介绍		tagOApiObjDataItem[必选, iDataLen指明后面跟着的数据总长度] tagOApiObjDataItem+具体数据[可选, iDataLen 指明后面具体数据的长度, 当该条目为文件夹时后面紧接着的数据是作为同级还是子结点主要看 nChild 值]	
参数	参数	类型	备注
	iDataLen	int	后面紧跟数据的长度
	iObjType	int	后面紧跟数据的类型
	idObj	DWORD	后面紧跟对象的 ID
	idParent	DWORD	后面紧跟对象的父节点 ID
	iCustomData	INT64	后方数据长度
	ret[10]	DWORD	预留字段

2.3.30. tagOApiObjectSrhReq

名称		tagOApiObjectSrhReq	
功能介绍		搜索请求	
参数	参数	类型	备注
	iReqVer	int	版本号 目前为 0
	iSrhOpt	int	参照定义 SRH_OBJ_OPT_XXX
	iObjType	int	0 表示任何类型
	idParent	DWORD	0 表示收藏夹
	idShapeSelect	DWORD	0 表示不限制图形

	bNoChild	BYTE	0 表示遍历搜索子文件夹, 1 表示搜索 idParent 一级
	iMaxCnt	int	0 表示不限制返回数量
	strSrhTxt[MAX_API_NAME_LEN]	char	搜索对象名称
	iSrhValue	int	搜索对象的图标 id、线型、

2.3.31. tagOApiObjectSrhRes

名称	tagOApiObjectSrhRes	
功能介绍	搜索响应	
参数	类型	备注
参数	idObj	DWORD 对象 id
	iObjType	int 对象类型

2.3.32. tagOApiObjChageHdr

名称	tagOApiObjChageHdr	
功能介绍	奥维对象改变监测结构体	
参数	类型	备注
参数	iVer	int 版本号
	iFlag	对象改变类型 0: 添加或修改 1: 删除
	nList	改变对象数量
	pdwList	改变对象id
	dwRet[20]	预留字段

2.3.33. tagOApiShapeFillCadPatHdr

名称		tagOApiShapeFillCadPatHdr	
功能介绍		填充对象结构体	
参数		类型	备注
参数	iVer	int	版本号
	nSfcpi	int	填充图形个数
	pSfcpi	tagOApiShapeFillCadPatItem*	填充图形属性
	dwRet[20]	DWORD	预留字段

2.3.34. tagOApiShapeFillCadPatItem

名称		tagOApiShapeFillCadPatItem	
功能介绍		表示一个填充图案的结构体	
参数		类型	备注
参数	strName[64]	char	填充图案名称
	strComment[255]	char	备注内容
	dDefaultScale	double	旋转角度
	id	DWORD	图案 id
	nSfcpu	int	填充位置个数
	pSfcpu	tagOApiShapeFillCadPatUnit *	填充位置结构体

2.3.35. tagOApiShapeFillCadPatUnit

名称		tagOApiShapeFillCadPatUnit	
功能介绍		表示一个填充图案的结构体	
参数		类型	备注
参数	dAngle	double	偏移角度
	dxOrg	double	
	dyOrg	double	
	dxDelta	double	线条增量
	dyDelta	double	
	nDash	int	DashList 数量
	dDashList[20]	double	

2.3.36. tagOmapReqContour

名称	tagOmapReqContour	
功能介绍	请求等高线结构体	
参数	类型	备注
idMap	INT64	该请求的编号, 多次请求时相同编号新的会替换旧的
	INT64	iValue
	double	x_lng1
	double	y_lat1
	double	x_lng2
	double	y_lat2
	BYTE	bRealL1 0:gcj02 经纬度 1:真实经纬度, 2:当前坐标系
	int	iMapLevel
	int	iElevValue
	int	iRet
	int	iNeedVip
	void *	pData
	int	nData
	void *	hWnd 取等高线过程每取多少次间隔发送一下 g_msgGetContourProcess 消息 (WPARAM=iTotal, PARAM=iDeal);
iPostCnt	int	

2.3.37. tagOApiMapObjEvent

名称	tagOApiMapObjEvent	
功能介绍	获取点击位置属性	
参数	类型	备注
iVersion	int	该请求的编号, 多次请求时相同编号新的会替换旧的
	DWORD	需要显示弹窗的奥维对象 ID
	RECT	奥维地图页面距离主程序窗口上下左右距离地图子窗口, 相对于程序主窗口的位置
	POINT	点击事件相对于地图子窗口的位置
	DWORD	预留字段

2.3.38. tagOApiMapObjShowWnd

名称	tagOApiMapObjShowWnd	
功能介绍	设置自定义弹窗属性	
参数	类型	备注
iVersion	int	该请求的编号, 多次请求时相同编号新的会替换旧的
	DWORD	需要显示弹窗的奥维对象 ID
	BYTE	关闭标志 0: 默认 1: 只有点击关闭按钮时关闭弹窗
	BYTE	关闭事件 0: 默认 1: 发送通知
	RECT	需要显示的窗口大小
	POINT	对象箭头指定的位置 (ptClickOfMap 点击事件相对于地图子窗口的位置)
	DWORD	预留字段
nHtml	int	Html 数据长度
pHtml	char*	以网页的形式显示窗口内容, Html 数据内容

2.3.39. tagCusMainMenu

名称		tagCusMainMenu	
功能介绍		工具栏自定义菜单栏属性	
参数		类型	备注
	nId	int	1~100: 表示功能性菜单项 <0 表示作为父结点后面跟的子菜单数量
	strTxt	DWORD	菜单项文字

2.3.40. tagOApiRevHdr

名称		tagOApiRevHdr	
功能介绍		版本号预留的通用结构体[100 长度]	
参数		类型	备注
	iVersion	int	版本号, 默认都为 0
	dwRet[24]	DWORD	预留字段

2.3.41. tagOApMapTrackDetail

名称		tagOApMapTrackDetail	
功能介绍		轨迹对象基本信息	
参数		类型	备注
参 数	idObj	int	返回 ID, 若 ID 为 0 表示没找到轨迹对象
	tStart	int	开始时间
	tEnd	int	结束时间
	iTotalPoint	int	轨迹点数量
	iTotalDist	int	轨迹长度(m)
	iAvgSpeed	int	平均速度(km/h)
	iMaxSpeed	int	最大速度(km/h)
	iMinAltitude	int	最低点海拔(m)
	iMaxAltitude	int	最高点海拔(m)
	idRet[20]	DWORD	预留字段

2.3.42. tagOApMapShapeDetail

名称		tagOApMapShapeDetail	
功能介绍		图形对象基本信息	
参数		类型	备注
参 数	idObj	int	返回 ID, 若 ID 为 0 表示没找到图形对象
	iPointCnt	int	顶点个数
	iAreaCnt	int	区域个数
	dPerimeter	double	周长(km)
	dAreaSize	double	面积(m ²)
	idRet[20]	DWORD	预留字段

2.3.43. tagCusMainBtn

名称		tagCusMainBtn	
功能介绍		自定义工具栏按钮结构体	
参数		类型	备注
参数	nId	int	按钮 id (范围 1-10), ID 相同覆盖原按钮
	strTxt[81]	char	按钮文字提示
	dwRet[20]	DWORD	预留字段
	iResult	int	查询时查询结果 -1: id 超出范围, 0: 不存在, 1: 存在
	nPngLen	int	图标数据长度
	pPngData	void*	图标数据

2.3.44. tagCusFavMenu

名称		tagCusFavMenu	
功能介绍		自定义收藏夹菜单项	
参数		类型	备注
参数	nId	int	菜单项 id (范围 1-10), ID 相同覆盖原菜单
	strTxt[81]	char	菜单名
	dwRet[20]	DWORD	预留字段
	iResult	int	查询时查询结果 -1: id 超出范围, 0: 不存在, 1: 存在
	nList	int	点击菜单时, 返回 ID 列表
	pidList	DWORD	如果有数据跟在本结构体的后面

2.3.45. tagCusMainMenu

名称		tagCusMainMenu	
功能介绍		自定义工具菜单栏	
参数		类型	备注
参数	nId	int	菜单项 id 1~100 表示有效的菜单 <0 表示作为父结点后面跟的子菜单数量
	strTxt[81]	char	菜单按钮名称
	dwRet[20]	DWORD	预留字段

2.3.46. tagOApiMapTrackDetail

名称		tagOApiMapTrackDetail	
功能介绍		轨迹信息结构体	
参数		类型	备注
参数	idObj	DWORD	轨迹对象 ID
	tStart	int	起始时间
	tEnd	int	终止时间

	iTotalPoint	int	轨迹点数量
	iTotalDist	int	轨迹长度 (米)
	iAvgSpeed	int	平均速度 (km/h)
	iMaxSpeed	int	最大速度 (km/h)
	iMinAltitude	int	最低海拔 (米)
	iMaxAltitude	int	最高海拔 (米)
	idRet[20]	DWORD	预留字段

2.3.47. tagOApiMapShapeDetail

名称		tagOApiMapShapeDetail	
功能介绍		轨迹信息结构体	
参数	参数	类型	备注
	idObj	DWORD	轨迹对象 ID
	iPointCnt	int	顶点个数
	iAreaCnt	int	区域个数
	dPerimeter	double	周长 (Km)
	dAreaSize	double	面积 (m ²)

2.3.48. tagOApiPrj

名称		tagOApiPrj	
功能介绍		奥维系统坐标系结构体	
参数	参数	类型	备注
	strPrjType	char	GEOGCS 开头地理坐标系; PROJCS 开头是投影坐标系
	strPrjName	char	
	strGeogcs	char	地理坐标系、定椭球体类型 也叫基准面
	strDatum	char	大地基准面
	strSpheroid	char	椭球体定义
	dSemimajorAxis	double	半长轴
	dSemiminorAxis	double	半短轴
	dInverseFlattening	double	反扁率
	strPrimem	char	本初子午线
	dPrimeMeridian	double	
	strAngularUnit	char	角单位
	dAngularUnit	double	角单位 每单位弧度
	strLinearUnit	char	线单位
	dLinearUnit	double	线单位
	strProjection	char	投影名称 投影坐标系用
	dFalseEasting	double	东偏移
	dFalseNorthing	double	北偏移
	dCentralMeridian	double	中央经线

dScaleFactor	double	比例因子
dLatitudeOfOrigin	double	远点
以下在 gis 系统中没有看到		
dLatitudeOfCenter	double	中心点
dStandardParallel_1	double	标准平行线 1
dStandardParallel_2	double	标准平行线 2
dLatitude_Of_1st_Point	double	第一个点纬度
dLatitude_Of_2	double	第二个点纬度
dLongitude_Of_1st_Point	double	第一个点经度
dLongitude_Of_2nd_Point	double	第二个点经度
dAzimuth	double	方位角
dLongitudeOfCenter	double	中心经度
iPrjType	int	坐标系类别
iCoordType	int	投影类型

2.3.49. tagOApiCusShowWnd

名称	tagOApiCusShowWnd		
功能介绍	奥维自定义非特定对象弹窗		
参数	类型	备注	
参数	idWnd	int	idWnd 范围请设置在 1-10 之间,id 相同则替换原窗口
	rectWnd	RECT	需要显示的窗口大小
	strDlgTitle	char	弹窗标题
	bCloseEvent	BYTE	关闭事件: 0: 无 1: 发送 [OMAP_API_CMD_CUS_WND_CLOSE_EVENT, tagOApiRevHdr+DWORD]

2.3.50. tagOApiWebSocketConf

名称	tagOApiWebSocketConf		
功能介绍	奥维 websocket 配置		
参数	类型	备注	
参数	iWssFlag	int	服务选项 0:不启用 1:启用 WebSocket 协议 2:启用 WebSocket 协议+SSL 协议
	dwWssUserLimitIP	DWORD	认证 ID
	wPort	DWORD	服务端口号
	strPsk	char	预共享秘钥

	strSslCaFn	char	CA 证书路径
	strSsl	char	服务器证书路径
	strSslKeyFn	char	服务器私钥路径
	bHttpTileService	int	是否启用 HTTP 瓦块服务

2.3.51. tagOApiCustomMap

名称	tagOApiCustomMap		
功能介绍	奥维自定义地图列表		
参数	类型	备注	
参 数	idMap	int	自定义地图 ID
	idLayerBig	int	最大图层 ID
	idLayerSmall	int	最小图层 ID
	iLayerBigFlag	DWORD	最大图层标志
	iLayerSmallFlag	DWORD	最小图层标志
	MinLevel	int	最小显示级别
	iMaxLevel	int	最大显示级别
	iCoordType	int	投影类型
	iTileFormat	int	
	iVersion	DWORD	版本号
	bUseLayer	BYTE	叠加层 0: 无 1: 使用
	bIsSatellite	BYTE	图片类型 0: 普通地图 1: 卫星影像地图
	strName	char	地图名称
	strHost	char	主机名
	wPort	WORD	端口号
	iHostStart	BYTE	主机起始编号
	iHostEnd	BYTE	主机终止编号
	strUrl	char	URL

2.4. 函数介绍

2.4.1. 字符类型转换函数

2.4.1.1. int OApiUtf16ToUtf8(WORD* wsUtf16, BYTE* strUtf8, int nBuf)

函数		int OApiUtf16ToUtf8(WORD* wsUtf16, BYTE* strUtf8, int nBuf)	
功能介绍		将 UTF16 类型字符转换为 UTF8 类型	
参数		类型	备注
参数	wsUtf16	WORD*	需要转换的字符
	strUtf8	BYTE*	转换后字符
	nBuf	int	转换后字符长度
返回值		int	转换后字符长度

2.4.1.2. WORD* OApiUtf8ToUtf16(char* strUtf8, WORD* utf16, int* pnU16)

函数		WORD* OApiUtf8ToUtf16(char* strUtf8, WORD* utf16, int* pnU16)	
功能介绍		将 UTF8 类型字符转换为 UTF16 类型	
参数		类型	备注
参数	strUtf8	char*	需要转换的字符
	utf16	BYTE*	转换后字符
	nBuf	int	转换后字符的长度
返回值		WORD*	转换后字符

2.4.2. 奥维对象初始化函数

2.4.2.1. tagOApiObjectOptHdr* OApiNewOptHdr(int iExtLen = 0)

函数		tagOApiObjectOptHdr* OApiNewOptHdr(int iExtLen = 0)	
功能介绍		初始化一个长度为 tagOapiObjectOptHdr+iextLen 存放收藏夹对象结构体指针	
参数		类型	备注
参数	iExtLen	int	需要转换的字符
返回值		tagOApiObjectOptHdr*	转换后字符

2.4.2.2. char* OApiNewObjectComment(char* strComment)

函数		char* OApiNewObjectComment(char* strComment)	
功能介绍		让一个新的 char 指针指向 strComment 地址	
参数		类型	备注
参数	strComment	char*	奥维对象备注内容
返回值		char*	指向 strComment 地址的指针

2.4.2.3. tagOApiObjGroupV1* NewOmapGroup()

函数	tagOApiObjGroupV1* NewOmapGroup()
功能介绍	初始化一个奥维文件夹对象，并设置为自动加载

2.4.2.4. tagOApiMapSignV1* NewOmapSign()

函数	tagOApiMapSignV1* NewOmapSign()
功能介绍	初始化一个奥维标签对象

2.4.2.5. tapTrackV1* NewOmapTrack()

函数	tagOApiMapTrackV1* NewOmapTrack()
功能介绍	初始化一个奥维轨迹对象，线宽为 3 像素、不透明度 50%，线条颜色为蓝绿色。(用户可自行设置)

2.4.2.6. tagOApiMapShapeV1* NewOmapShape()

函数	tagOApiMapShapeV1* NewOmapShape()
功能介绍	初始化一个奥维图形对象，线条宽度为 1 像素、线条不透明度与填充不透明度为 50%，线条颜色与填充颜色为蓝色。(用户可自行设置)

2.4.2.7. tagOApiMapShapeV1* NewOmapDir()

函数	tagOApiMapDirectionsV1* NewOmapDir()
功能介绍	初始化一个奥维路线对象，颜色为红色。(用户可自行设置)

2.4.2.8. void* OApiZeroMalloc(int iSize)

函数	void* OApiZeroMalloc(int iSize)		
功能介绍	初始化一个长度为 iSize,每一位值都为 0 的 void 指针		
参数	类型	备注	
参数	iSize	int	奥维对象的备注内容
返回值	void*	长度为 iSize,每一位值都为 0 的 void 指针	

2.4.2.9. Init(WORD* wsAppClassName)

函数	void Init(WORD* wsAppClassName)		
功能介绍	初始化窗体类名		
参数	类型	备注	
参数	wsAppClassName	WORD*	窗体类名

2.4.3. 奥维对象数据处理函数

2.4.3.1. void FreeOmapObject(void* pObject, int iObjType)

函数		void FreeOmapObject(void* pObject, int iObjType)	
功能介绍		释放结构体对象所有指针	
参数		类型	备注
参数	pObject	void*	结构体对象指针
	iObjType	int	结构体类型id

2.4.3.2. tagOApiObjGroupV1* GetOmapGroupFromBuf(void* pBuf, int nBuf)

函数		tagOApiObjGroupV1* GetOmapGroupFromBuf(void* pBuf, int nBuf)	
功能介绍		从 pBuf 中取长度为 nBuf 的数据并转换成 tagOapiObjGroupV1 结构体数据	
参数		类型	备注
参数	pBuf	void*	奥维互动地图端返回的消息数据
	nBuf	int	需要转换的数据长度
返回值	tagOApiObjGroupV1*	tagOapiObjGroup 结构体	

2.4.3.3. tagOApiMapSignV1* GetOmapSignFromBuf(void* pBuf, int nBuf)

函数		tagOApiMapSignV1* GetOmapSignFromBuf(void* pBuf, int nBuf)	
功能介绍		从 pBuf 中取出长度为 nBuf 的数据并转换成 tagOApiMapSignV1 结构体数据	
参数		类型	备注
参数	pBuf	void*	奥维互动地图端返回的消息数据
	nBuf	int	需要转换的数据长度
返回值	tagOApiMapSignV1*	tagOApiMapSignV1 结构体	

2.4.3.4. tagOApiMapTrackV1* GetOmapTrackFromBuf(void* pBuf, int nBuf)

函数		tagOApiMapTrackV1* GetOmapTrackFromBuf(void* pBuf, int nBuf)	
功能介绍		从 pBuf 中取长度为 nBuf 的数据并转换成 tagOApiMapTrackV1 结构体数据	
参数		类型	备注
参数	pBuf	void*	奥维互动地图端返回的消息数据
	nBuf	int	需要转换的数据长度
返回值	tagOApiMapTrackV1*	tagOApiMapTrackV1 结构体	

2.4.3.5. tagOApiMapShapeV1* GetOmapShapeFromBuf(void* pBuf, int nBuf)

函数	tagOApiMapShapeV1* GetOmapShapeFromBuf(void* pBuf, int nBuf)	
功能介绍	从 pBuf 中取长度为 nBuf 的数据并转换成 tagOApiMapShapeV1 结构体数据	
参数	类型	备注
参数	pBuf	void*
	nBuf	int
返回值	tagOApiMapShapeV1*	tagOApiMapShapeV1结构体

2.4.3.6. tagOApiMapDirectionsV1* GetOmapDirFromBuf(void* pBuf, int nBuf)

函数	tagOApiMapDirectionsV1* GetOmapDirFromBuf(void* pBuf, int nBuf)	
功能介绍	从 pBuf 中取长度为 nBuf 的数据并转换成 tagOApiMapDirectionsV1 结构体数据	
参数	类型	备注
参数	pBuf	void*
	nBuf	int
返回值	tagOApiMapDirectionsV1*	tagOApiMapDirectionsV1结构体

2.4.3.7. BYTE *SetOmapObjectToBuf(void *pObj, int iObjType, int &nBuf)

函数	BYTE *SetOmapObjectToBuf(void *pObj, int iObjType, int &nBuf)	
功能介绍	将 pObj 结构体对象转换为 BYTE 类型数据	
参数	类型	备注
参数	pObj	tagOApiMemBuf *
	iObjType	char *
	nBuf	int
返回值	BYTE *	转换后的数据

2.4.3.8. char* OApiCopyComment(void* pBuf, int nComment)

函数	char* OApiCopyComment(void* pBuf, int nComment)	
功能介绍	获取 pBuf 后长度为 nComment 的地址	
参数	类型	备注
参数	pBuf	void*
	nComment	int
返回值	char*	奥维对象的备注内容

2. 4. 3. 9. bool AppendMemBuf(tagOApiMemBuf *pMb, char *pBuf, int nBuf)

函数	bool AppendMemBuf(tagOApiMemBuf *pMb, char *pBuf, int nBuf)	
功能介绍	将长度为 nBuf 的 pBuf 拼接到 pMb 后	
参数	类型	备注
参数	pMb	tagOApiMemBuf *
	pBuf	char *
	nBuf	int
返回值	bool	拼接成功将返回true， 失败返回false

2. 4. 3. 10. void FreeShapeFillPatList(tagOApiShapeFillCadPatHdr *pHdr, bool bFreeMe)

函数	void FreeShapeFillPatList(tagOApiShapeFillCadPatHdr *pHdr, bool bFreeMe)	
功能介绍	释放 tagOapiShapeFillCadPatHdr 所有指针	
参数	类型	备注
参数	pHdr	tagOApiShapeFillCadPatHdr *
	bFreeMe	Bool
		是否释放tagOapiShapeFillCadPatHdr指针

2.4.4. 设置奥维对象函数

2.4.4.1. void SetOmapTmpSign(double lat, double lng, bool bGcj02)

函数	void SetOmapTmpSign(double lat, double lng, bool bGcj02)		
功能介绍	设置临时标签		
参数	参数	类型	备注
参数	lat	double	经度
	lng	double	纬度
	bGcj02	bool	(真实经纬度为true, 偏移经纬度为false)

2.4.4.2. int SetOmapLevelAndLocation(int iMapID, int iMapLevel, double lng, double lat)

函数	int SetOmapLevelAndLocation(int iMapID, int iMapLevel, double lng, double lat)		
功能介绍	将 pObj 结构体对象转换为 BYTE 类型数据		
参数	参数	类型	备注
参数	iMapID	int	地图id
	iMapLevel	int	地图显示级别
	lng	double	纬度
	lat	double	经度
返回值	int	返回值大于0表示成功	

2.4.4.3. BYTE *SetOmapObjectToBuf(void *pObj, int iObjType, int &nBuf)

函数	BYTE *SetOmapObjectToBuf(void *pObj, int iObjType, int &nBuf)		
功能介绍	将 pObj 结构体对象转换为 BYTE 类型数据		
参数	参数	类型	备注
参数	pObj	tagOApiMemBuf *	数据结构体数据
	iObjType	char *	结构体类型id
	nBuf	int	转换后的数据长度
返回值	BYTE *	转换后的数据	

2.4.4.4. int MoveOmapObject(DWORD idParent, int nIndexInParent, DWORD* pidList, int nId)

函数	int MoveOmapObject(DWORD idParent, int nIndexInParent, DWORD* pidList, int nId);		
功能介绍	将 pObj 结构体对象转换为 BYTE 类型数据		
参数	参数	类型	备注
参数	idParent	DWORD	移动到指定位置ID
	nIndexInParent	int	插入顺序
	pidList	DWORD*	需要移动的对象id
	nId	int	移动对象数量
返回值	int	等于0表示移动成功	

2. 4. 4. 5. int SetOmapLevelAndLocationExt(int iMapID, int iMapLevel, double x_lng, double y_lat, BYTE bActive = FALSE, BYTE bReal = 0)

函数	int SetOmapLevelAndLocationExt(int iMapID, int iMapLevel, double x_lng, double y_lat, BYTE bActive = FALSE, BYTE bReal = 0)		
功能介绍	设置地图位置		
参数	参数	类型	备注
参数	iMapID	int	地图id
	iMapLevel	int	地图显示级别
	x_lng	double	bReal=0 或 bReal=1 传入经纬度
	y_lat	double	bReal=2 传入平面坐标
	bActive	BYTE	0: 不改变 1: 将奥维置为最屏幕最前端
	bReal	BYTE	0: 偏移经纬度 1: 真实经纬度 2: 平面坐标
返回值	int	返回值大于0表示成功	

2. 4. 4. 6. int SetOmapSelObjectTree(int iCmd, tagOApiMemBuf tag, bool bFreeBuf = true)

函数	int SetOmapSelObjectTree(int iCmd, tagOApiMemBuf tag, bool bFreeBuf = true)		
功能介绍	将树形结构数据添加到奥维互动地图收藏夹中		
参数	参数	类型	备注
参数	iCmd	int	OMAP_API_CMD_GET_SEL_OBJ_TREE
	tag	tagOApiMemBuf	需要添加的数据
	bFreeBuf	bool	是否销毁释放tag
返回值	int	返回值大于0表示成功	

2. 4. 4. 7. int SetOmapSelectIdList(DWORD* pidList, int nId)

函数	int SetOmapSelectIdList(DWORD* pidList, int nId)		
功能介绍	根据 id 选中奥维互动地图端收藏夹对象		
参数	参数	类型	备注
参数	pidList	DWORD*	需要选中的对象id列表
	bLlType	BYTE	需要选中对象的数量
返回值	int	返回值大于0表示成功	

2. 4. 4. 8. int SetOmapButton(int iAdd, char *strTip)

函数	int SetOmapButton(int iAdd, char *strTip)		
功能介绍	设置按钮		
参数	参数	类型	备注
参数	iAdd	DWORD*	需要选中的对象id列表
	strTip	BYTE	名称
返回值	int	返回值大于0表示成功	

2. 4. 4. 9. int SetSysCoorPoint(int iCoorType, tagOApiRelatePointV1 *pPoint, int nPoint)

函数	int SetSysCoorPoint(int iCoorType, tagOApiRelatePointV1 *pPoint, int nPoint)	
功能介绍	设置关联点	
参数	类型	备注
参数	iCoorType	int 0: 奥维平面坐标 3: 关联点 4: 简化参数
	pPoint	tagOApiRelatePointV1 *
	nPoint	int iCoorType=0与4只有第1个关联点有效 icoorType=3支持多个关联点
返回值	int	返回值大于0表示成功

2. 4. 4. 10. int SetOmapObject(DWORD idObj, tagOApiObjGroupV1* pGroup, DWORD idParent = 0, int iPosition = 0);

函数	int SetOmapObject(DWORD idObj, tagOApiObjGroupV1* pGroup, DWORD idParent = 0, int iPosition = 0)	
功能介绍	设置文件夹对象属性并添加到收藏夹	
参数	类型	备注
参数	idObj	DWORD idObj:对象ID, 为0时则表示要添加, 此时需要设置idParent
	pGroup	tagOApiObjGroupV1*
	idParent	DWORD 当指定了idObj的值时, 此项值无用 idObj=0, 指示要插入的父节点的ID, 此时idParent=0表示插入到根节点下
	iPosition	DWORD 当指定了idObj的值时, 此项值无用 当idObj=0时, iPosition指示要插入到父节点下的哪个位置 iPosition=0表示插入到最后
返回值	int	返回值大于0表示添加成功

2.4.4.11. **int SetOmapObject(DWORD idObj, tagOApiMapSignV1* pSign, DWORD idParent = 0, int iPosition = 0)**

函数	int SetOmapObject(DWORD idObj, tagOApiMapSignV1* pSign, DWORD idParent = 0, int iPosition = 0)	
功能介绍	设置标签对象属性并添加到收藏夹	
参数	类型	备注
	idObj	DWORD idObj:对象ID, 为0时则表示要添加, 此时需要设置idParent
	pSign	tagOApiMapSignV1* 标签对象
参数	idParent	当指定了idObj的值时, 此项值无用 当idObj=0时, 指示要插入的父节点的ID, 此时idParent=0表示插入到根节点下
		当指定了idObj的值时, 此项值无用 当idObj=0时, iPosition指示要插入到父节点下的哪个位置 iPosition=0表示插入到最后
返回值	int	返回值大于0表示添加成功

2.4.4.12. **int SetOmapObject(DWORD idObj, tagOApiMapTrackV1* pTrack, DWORD idParent = 0, int iPosition = 0)**

函数	int SetOmapObject(DWORD idObj, tagOApiMapTrackV1* pTrack, DWORD idParent = 0, int iPosition = 0)	
功能介绍	设置轨迹对象属性并添加到收藏夹	
参数	类型	备注
	idObj	DWORD idObj:对象ID, 为0时则表示要添加, 此时需要设置idParent
	pTrack	tagOApiMapSignV1* 轨迹对象
参数	idParent	当指定了idObj的值时, 此项值无用 当idObj=0时, 指示要插入的父节点的ID, 此时idParent=0表示插入到根节点下
		当指定了idObj的值时, 此项值无用 当idObj=0时, iPosition指示要插入到父节点下的哪个位置 iPosition=0表示插入到最后
返回值	int	返回值大于0表示添加成功

2.4.4.13. **int SetOmapObject(DWORD idObj, tagOApiMapShapeV1* pShape, DWORD idParent = 0, int iPosition = 0)**

函数	int SetOmapObject(DWORD idObj, tagOApiMapShapeV1* pShape, DWORD idParent = 0, int iPosition = 0)		
功能介绍	设置图形对象属性并添加到收藏夹		
参数	参数	类型	
参数	idObj	DWORD	idObj:对象ID, 为0时则表示要添加, 此时需要设置idParent
	pShape	tagOApiMapSignV1*	图形对象
	idParent	DWORD	当指定了idObj的值时, 此项值无用 当idObj=0时, 指示要插入的父节点的ID, 此时idParent=0表示插入到根节点下
	iPosition	DWORD	当指定了idObj的值时, 此项值无用 当idObj=0时, iPosition指示要插入到父节点下的哪个位置 iPosition=0表示插入到最后
返回值	int	返回值大于0表示添加成功	

2.4.4.14. **int SetOmapObject(DWORD idObj, tagOApiMapDirectionsV1* pDir, DWORD idParent = 0, int iPosition = 0)**

函数	int SetOmapObject(DWORD idObj, tagOApiMapDirectionsV1* pDir, DWORD idParent = 0, int iPosition = 0)		
功能介绍	设置导航对象属性并添加到收藏夹		
参数	参数	类型	
参数	idObj	DWORD	idObj:对象ID, 为0时则表示要添加, 此时需要设置idParent
	pDir	tagOApiMapSignV1*	导航对象
	idParent	DWORD	当指定了idObj的值时, 此项值无用 当idObj=0时, 指示要插入的父节点的ID, 此时idParent=0表示插入到根节点下
	iPosition	DWORD	当指定了idObj的值时, 此项值无用 当idObj=0时, 指示要插入的父节点的ID, 此时idParent=0表示插入到根节下
返回值	int	返回值大于0表示添加成功	

2.4.4.15. int SetOmapObject(DWORD idObj, int iObjType, void* pObj, DWORD idParent = 0, int iPosition = 0)

函数	int SetOmapObject(DWORD idObj, int iObjType, void* pObj, DWORD idParent = 0, int iPosition = 0)		
功能介绍	删除收藏夹对象		
参数	类型	备注	
参数	idObj	DWORD	idObj:对象ID, 为0时则表示要添加, 此时需要设置idParent
	iObjType	int	对象类型id (例如: OMAP_OBJ_TYPE_GROUP)
	pObj	void*	文件夹对象
	idParent	DWORD	当指定了idObj的值时, 此项值无用 当idObj=0时, 指示要插入的父节点的ID, 此时idParent=0表示插入到根节点下
	iPosition	DWORD	当指定了idObj的值时, 此项值无用 当idObj=0时, 指示要插入的父节点的ID, 此时idParent=0表示插入到根节下
	返回值	int	返回值大于0表示添加成功

2.4.5. 获取奥维对象函数

2.4.5.1. void GeoDecodeLatlng(double lat, double lng, bool bGcj02)

函数	void GeoDecodeLatlng(double lat, double lng, bool bGcj02)		
功能介绍	根据经纬度解析地址		
参数	类型	备注	
参数	lat	double	经度
	lng	double	纬度
	bGcj02	bool	(真实经纬度为true, 偏移经纬度为false)

2.4.5.2. int GetOmapSelObjectTree(BYTE bLIType = 0);

函数	int GetOmapSelObjectTree(BYTE bLIType = 0)		
功能介绍	获取收藏夹对象的树		
参数	类型	备注	
参数	BYTE	bLIType	坐标类型
返回值	int		返回值大于0表示成功

2.4.5.3. int GetOmapObjectList(DWORD idParent)

函数	int GetOmapObjectList(DWORD idParent)		
功能介绍	获取收藏夹对象列表		
参数	类型	备注	
参数	DWORD	idParent	获取收藏夹对象列表的父节点id。 idParent=0, 表示取根节点
返回值	int		返回值大于0表示成功

2.4.5.4. int GetOmapObject(DWORD idObj, BYTE bLIType = 0)

函数	int GetOmapObject(DWORD idObj, BYTE bLIType = 0)		
功能介绍	获取收藏夹对象		
参数	类型	备注	
参数	DWORD	idObj	当前对象的id
参数	BYTE	bLIType	坐标类型,默认为0
返回值	int		返回值大于0表示成功

2. 4. 5. 5. int GetOmapSelectList(int iFlag = 1)

函数		int GetOmapSelectList(int iFlag = 1)	
功能介绍		获取奥维互动地图收藏夹中选中的对象列表	
参数	参数	类型	备注
参数	iFlag	int	所有对象 OBJ_SELECT_CHECK_FLAG_ALL 0 选中对象，选中的文件夹自动包含所有子对象 OBJ_SELECT_CHECK_FLAG_GROUP_JOIN 1 选中对象，选中的文件夹不自动包括非选中子对象 OBJ_SELECT_CHECK_FLAG_SEL_ONLY 2
返回值		int	返回值大于0表示成功

2. 4. 5. 6. int SrhOmapObject(tagOApiObjectSrhReq *pSrhReq)

函数		int SrhOmapObject(tagOApiObjectSrhReq *pSrhReq)	
功能介绍		根据条件结构体检索对象 命令回调返回tagOApiObjectSrhReq+ n*tagOApiObjectSrhRes 数据长度: sizeof(tagOApiObjectSrhReq)+sizeof(tagOApiObjectSrhRes)*n	
参数	参数	类型	备注
参数	pSrhReq	tagOApiObjectSrhReq *	检索条件结构体
返回值		int	返回值大于0表示成功

2. 4. 5. 7. int GetOmapMapFile(tagOmapGetMapFile *pReq, int nReq)

函数		int GetOmapMapFile(tagOmapGetMapFile *pReq, int nReq)	
功能介绍		取图命令可能分多次返回，每次返回一个tagOmapGetMapFile。若后面还有多余数据时,多余部分为图片数据	
参数	参数	类型	备注
参数	pReq	tagOmapGetMapFile *	取图结构体
参数	nReq	int	数量
返回值		int	返回值大于0表示成功

2. 4. 5. 8. void GetSysCoorCenter()

函数		void GetSysCoorCenter()
功能介绍		获取平面坐标中心点 返回消息为tagOApiRelatePointV1 目前只平面坐标x, y可用

2.4.5.9. int GetOmapMapData(double x_lng1, double y_lat1, double x_lng2, double y_lat2, INT64 iMaxPixel, int iMaxLevel)

函数	int GetOmapMapData(double x_lng1, double y_lat1, double x_lng2, double y_lat2, INT64 iMaxPixel, int iMaxLevel)		
功能介绍	通过发送两个对角点的经纬度获取地图数据		
参数	参数	类型	备注
参数	x_lng1	double	第一个点纬度
	y_lat1	double	第一个点经度
	x_lng2	double	第二个点纬度
	y_lat2	double	第二个点经度
	iMaxPixel	INT64	像素（尺寸）
	iMaxLevel	int	地图级别
返回值	int	返回值大于0表示成功	

2.4.5.10. void GetSysCoorInfo()

函数	void GetSysCoorInfo()
功能介绍	获取关联点

2.4.5.11. int FollowObjChange(int iAppID)

函数	int FollowObjChange(int iAppID)		
功能介绍	根据条件监控奥维对象改变情况 对象改变时，奥维主动发送OMAP_WIN_CMD_FOLLW_OBJ_CHG_LIST命令 命令回调返回tagOApiObjChageHdr + (n* DWORD) n个对象id int nLen = sizeof(tagOApiObjChageHdr) + sizeof(DWORD)*nId; tagOApiObjChageHdr *pObjChange = (tagOApiObjChageHdr *)zMalloc(nLen)		
参数	参数	类型	备注
参数	iAppID	int	OAPI_APP_ID_QUERY: -100 查询当前值 OAPI_APP_ID_ALL: -1 奥维发送所有注册AppID OAPI_APP_ID_NONE: 0 不监控（取消监控） >0: 指定AppID
返回值	int	返回值大于0表示监控成功	

2.4.5.12. int SendGetShapeFillPatList()

函数	SendGetShapeFillPatList()
功能介绍	获取所有填充图案列表

2.4.5.13. tagOApiShapeFillCadPatHdr *DeocdeShapeFillPatList(void *pBuf, int nBuf)

函数	tagOApiShapeFillCadPatHdr *DeocdeShapeFillPatList(void *pBuf, int nBuf)		
功能介绍	解析返回的填充图案列表		
参数	类型	备注	
参 数	pBuf	void *	填充图案列表数据
	nBuf	int	数据长度
返回值	tagOApiShapeFillCadPatHdr *	填充对象结构体对象	

2.4.5.14. int GetOmapContour(tagOmapReqContour *pReq, int nReq)

函数	GetOmapContour(tagOmapReqContour *pReq, int nReq)		
功能介绍	获取地图等高线		
参数	类型	备注	
参 数	pReq	tagOmapReqContour *	请求等高线数据结构体
	nReq	int	请求区域数量
返回值	int	填充对象结构体对象	

2.4.6. 删除奥维对象函数

2.4.6.1. void CleanOmapTmpObject()

函数	void CleanOmapTmpObject()
功能介绍	清除奥维临时对象

2.4.6.2. int DelOmapObject(DWORD* pidList, int nId, bool bNoGroup = false, bool bForceDel = false)

函数	int DelOmapObject(DWORD* pidList, int nId, bool bNoGroup = false, bool bForceDel = false)		
功能介绍	删除收藏夹对象		
参数	类型	备注	
pidList	DWORD*	删除对象的id组	
nId	int	删除对象的数量	
bNoGroup	bool	bNoGroup为真时, 不删除最顶层的本身文件夹	
bForceDel	bool	bForceDel为真时会强行删除, 忽略只读属性	
返回值	int	返回值大于0表示成功	

2.4.7. 奥维校验函数

2.4.7.1. bool CmdCheck(int iCmdId)

函数		bool CmdCheck(int iCmdId)	
功能介绍		校验版本和请求参数 id。版本低于 621 的奥维客户端不支持设置地图位置、获取收藏夹对象列表、获取收藏夹对象、设置收藏夹对象、删除收藏夹对象	
参数		类型	备注
参数	iCmdId	int	请求参数 id
返回值		bool*	校验成功返回 true 失败返回 false

2.4.7.2. int ReqVip5Demo(bool bSend)

函数		int ReqVip5Demo(bool bSend)	
功能介绍		判断用户是否为 VIP5	
参数		类型	备注
参数	bSend	bool	是否请求
返回值		int	返回值大于 0 表示成功

2.4.7.3. int CheckInterfaceVer(int iType, int iVer, bool bSendReq)

函数		int CheckInterfaceVer(int iType, int iVer, bool bSendReq)	
功能介绍		校验接口版本号	
参数		类型	备注
参数	iType	int	OAPI_INTERFACE_TYPE_XXX 需要自定义 iType为0时表示不限制类型
	iVer	int	当前版本号
	bSendReq	bool	bSendReq为true时忽略iVer的值
返回值		int	返回值大于0表示成功

2.4.8. 窗体注册函数

2.4.8.1. void RegisterToOmap()

函数		void RegisterToOmap()
功能介绍		将奥维第三方接口程序窗体类名发送给奥维互动地图（注册类名）
返回值		没有返回值。如果注册失败，无法接收到奥维互动地图返回的消息。

2.4.9. 消息发送函数

2.4.9.1. **int SendMessageToOmap(int iCmdId, void* pData, int nData, INT64 *plResult = NULL, int iTimeoutMs = -1)**

函数	int SendMessageToOmap(int iCmdId, void* pData, int nData, INT64 *plResult = NULL, int iTimeoutMs = -1)	
功能介绍	发送消息给奥维互动地图客户端	
参数	参数	类型
参数	iCmdId	int *
	pData	void*
	nData	int
	plResult	INT64 *
	iTimeoutMs	int
返回值		int
		返回值大于0表示成功

2.4.10. 奥维地图视野跟随函数

2.4.10.1. **int FollowWinMapInfo(int iFollowMs)**

函数	int FollowWinMapInfo(int iFollowMs)		
功能介绍	地图视野跟随		
参数	参数	类型	
参数	iFollowMs	int	-1: 查询 1: 停止 > 10: ms 建议值:500, 1*1000, 2*1000, 5*1000。 因为奥维检测精度不高
返回值		int	返回值大于0表示成功

2.5. 示例功能介绍

2.5.1. 取经纬度

发送消息: SendMessageToOmap(OMAP_WIN_CMD_GET_LATLNG, NULL, 0)

接收消息: WindowProc(UINT message, WPARAM wParam, LPARAM lParam)

处理消息:

```
if (message == WM_COPYDATA && wParam == OMAP_WIN_API_FLAG)
{
    COPYDATASTRUCT* pcd = (COPYDATASTRUCT*)lParam;
    int nAppId = (pcd->dwData >> 16)&0xffff;
    int nCmdId = pcd->dwData & 0xffff;
    if (nCmdId == OMAP_WIN_CMD_GET_LATLNG)
    {
        tagOmapLatlng oll;
        memcpy(&oll, pcd->lpData, sizeof(oll));
        //取得经纬度: oll.lat、oll.lng
    }
}
```

2.5.2. 设置临时标签

调用函数: SetOmapTmpSign(double lat, double lng, bool bGcj02)

```
{
    SetOmapTmpSign(double lat, double lng, bool bGcj02)
    tagOmapLatlng oll;
    oll.bReal = bGcj02 ? 0 : 1;
    oll.lat = lat;
    oll.lng = lng;
    SendMessageToOmap(OMAP_WIN_CMD_SET_LATLNG, &oll, sizeof(oll));
}
```

示例: SetOmapTmpSign(40.12345, 116.23456, false)

发送消息: SendMessageToOmap(OMAP_WIN_CMD_SET_LATLNG, &oll, sizeof(oll));

接收消息: 无

处理消息: 无

2.5.3. 设置地图位置

调用函数：

SetOmapLevelAndLocationExt(int iMapID, int iMapLevel, double x_lng, double y_lat, BYTE bActive, BYTE bReal)

```
{  
    tagOmapLocationExt ole;  
    memset(&ole, 0, sizeof(ole));  
    ole.ol.iMapID = iMapID; //地图id  
    ole.ol.iMapLevel = iMapLevel;//显示级别  
    ole.ol.ll.lat = y_lat;//经度 (Y值)  
    ole.ol.ll.lng = x_lng;//纬度 (X值)  
    ole.ol.ll.bReal = bReal;// 0:偏移经纬度 1:真实经纬度 2:平面坐标  
    ole.bActive = bActive;  
    SendMessageToOmap(OMAP_WIN_CMD_SET_MAP_LOCATION_EXT, &ole, sizeof(ole));  
}
```

示例： SetOmapLevelAndLocationExt(101, 16, 40437103.91, 2936164.61, 0, 2)

发送消息： SendMessageToOmap(OMAP_WIN_CMD_SET_MAP_LOCATION_EXT, &ole, sizeof(ole));

接收消息：无

处理消息：无

2.5.4. 设置照片列表

结构体赋值：

```
tagOPhotoMapPic omp[1000];  
memset(omp, 0, sizeof(omp));  
int i;  
for (i = 0; i<1000; i++)  
{  
    omp[i].idPic = i + 10000;  
    omp[i].lat = 30 + 0.001*i;  
    omp[i].lng = 116 + 0.001*i;  
}
```

发送消息： SendMessageToOmap(OMAP_WIN_CMD_SHOW_PICTURE, omp, sizeof(omp));

接收消息：无

处理消息：无

2.5.5. 地址解析

调用函数：

```
GeoDecodeLatlng(double lat, double lng, bool bGcj02)
{
    tagOmapGeoDecode ogd;
    memset(&ogd, 0, sizeof(ogd));
    ogd.oll.bReal = bGcj02 ? 0 : 1;
    ogd.oll.lat = lat;
    ogd.oll.lng = lng;
    SendMessageToOmap(OMAP_WIN_CMD_LATLNG_GEOCODE, &ogd, sizeof(ogd));
}
```

示例：g_omapApi.GeoDecodeLatlng(40.12345,116.22334, false) //求指定经纬度的地理位置

发送消息：SendMessageToOmap(OMAP_WIN_CMD_LATLNG_GEOCODE, &ogd, sizeof(ogd));

接收消息：WindowProc(UINT message, WPARAM wParam, LPARAM lParam)

处理消息：if (message == WM_COPYDATA && wParam == OMAP_WIN_API_FLAG)

```
{
    COPYDATASTRUCT* pcd = (COPYDATASTRUCT*)lParam;
    int nAppId = (pcd->dwData >> 16)&0xffff;
    int nCmdId = pcd->dwData & 0xffff;
    if (nCmdId == OMAP_WIN_CMD_LATLNG_GEOCODE)
    {
        tagOmapGeoDecode* pgd = (tagOmapGeoDecode*)pcd->lpData;
        CString ss;
        ss.Format(L"经纬度: %lf,%lf, 解析结果:\r\n", pgd->oll.lat, pgd->oll.lng);
        //如果是返回GEO_DECODE_ERR_WAIT, 则需要等1秒后再请求, omap程序正从服务器下载行政地区数据
        if (pgd->nLevel == GEO_DECODE_ERR_WAIT)
            ss.AppendFormat(L"等待数据\r\n");
        else if (pgd->nLevel == GEO_DECODE_ERR_NO_DATA)
            ss.AppendFormat(L"找不到对应信息\r\n");
        int i;
        for (i = 0; i < pgd->nLevel; i++)
        {
            ss.AppendFormat(L"%os\r\n", Utf8ToCString(pgd->strArea[i]));
        }
        //解析后地址: ss
    }
}
```

2.5.6. 根据时间取轨迹经纬度

```
结构体: tagOPhotoMapPic omp[100];
memset(&omp, 0, sizeof(omp));
int i;
for (i = 0; i < 100; i++)
{
    omp[i].tmPhoto = (unsigned int)time(NULL) - 86400 + i * 60;
}
```

发送消息: SendMessageToOmap(OMAP_WIN_CMD_TIME_TO_LATLNG, &omp, sizeof(omp));

接收消息: WindowProc(UINT message, WPARAM wParam, LPARAM lParam)

```
处理消息: if (message == WM_COPYDATA && wParam == OMAP_WIN_API_FLAG)
{
    COPYDATASTRUCT* pcd = (COPYDATASTRUCT*)lParam;
    int nAppId = (pcd->dwData >> 16)&0xffff;
    int nCmdId = pcd->dwData & 0xffff;
    if (nCmdId == OMAP_WIN_CMD_TIME_TO_LATLNG)
    {
        tagOPhotoMapPic* pOmp = (tagOPhotoMapPic*)pcd->lpData;
        int nOmp = pcd->cbData / sizeof(tagOPhotoMapPic);
        int nCount = 0;
        int i;
        for (i = 0; i < nOmp; i++)
        {
            if (pOmp[i].lat != 0 || pOmp[i].lng != 0)
            {
                nCount++;
            }
        }
        CString ss;
        ss.Format(L"共有%d张照片获得经纬度:\r\n", nCount);
        for (i = 0; i < nOmp; i++)
        {
            if (pOmp[i].lat != 0 || pOmp[i].lng != 0)
            {
                ss.AppendFormat(L"第%d张经纬度: %lf,%lf\r\n", i + 1, pOmp[i].lat, pOmp[i].lng);
            }
        }
        //取得的所有经纬度信息 ss
    }
}
```

2.5.7. 取收藏夹对象列表

//idParent: 要获取的父节点的 ID, 0 表示根节点 idObj 为 1 表示取根节点

调用函数: GetOmapObjectList(DWORD idParent)

```
{  
    tagOApiObjectOptHdr* pOoh = OApiNewOptHdr(0);  
    pOoh->idObj = idParent == 0 ? 1 : idParent;  
    int r = SendMessageToOmap(OMAP_API_CMD_GET_OBJ_LIST, pOoh,  
    sizeof(tagOApiObjectOptHdr));  
    free(pOoh);  
    return r;  
}
```

示例: GetOmapObjectList(0)//取得整个收藏夹对象列表

发送消息: SendMessageToOmap(OMAP_API_CMD_GET_OBJ_LIST, pOoh,
sizeof(tagOApiObjectOptHdr));

接收消息: WindowProc(UINT message, WPARAM wParam, LPARAM lParam)

```
处理消息: if (message == WM_COPYDATA && wParam == OMAP_WIN_API_FLAG)  
{  
    COPYDATASTRUCT* pcd = (COPYDATASTRUCT*)lParam;  
    int nAppId = (pcd->dwData >> 16)&0xffff;  
    int nCmdId = pcd->dwData & 0xffff;  
  
    if (nCmdId == OMAP_API_CMD_GET_OBJ_LIST)  
    {  
        if (pcd->cbData == 0)  
        {  
            MessageBox(_T("当前奥维地图绑定用户不是VIP用户，不支持此API接口"));  
            return TRUE;  
        }  
        BYTE* pData = (BYTE*)pcd->lpData;  
        int nList = (pcd->cbData - sizeof(tagOApiObjectOptHdr)) / sizeof(tagOApiItemListV1);  
        tagOApiItemListV1* pOlv = (tagOApiItemListV1*)(pData + sizeof(tagOApiObjectOptHdr));  
        m_lcResult.DeleteAllItems();  
        for (int i = 0; i < nList; i++)  
        {  
            CString strType;//对象类型  
            if (pOlv[i].iType == OMAP_OBJ_TYPE_GROUP)  
            {  
                strType = L"文件夹";  
            }
```

```
        }
        else if (pOlv[i].iType == OMAP_OBJ_TYPE_SIGN)
        {
            strType = L"标签";
        }
        else if (pOlv[i].iType == OMAP_OBJ_TYPE_TRACK)
        {
            strType = L"轨迹";
        }
        else if (pOlv[i].iType == OMAP_OBJ_TYPE_SHAPE)
        {
            strType = L"图形";
        }
        else if (pOlv[i].iType == OMAP_OBJ_TYPE_DIR)
        {
            strType = L"路线";
        }
        else
            strType = L"未知";
        m_lcResult.InsertItem(i, strType);//将对象类型添加到第一列
        m_lcResult.SetItemText(i, 1, Utf8ToCString(pOlv[i].strName));//将对象名称添加到第二列
        m_lcResult.SetItemData(i, pOlv[i].idObj);//保存id到对应位置，可用来获取对象信息
    }
}
}
```

2.5.8. 取收藏夹对象

调用函数：GetOmapObject(DWORD idObj, BYTE bLIType)

```
{  
    tagOApiObjectOptHdr* pOoh = OApiNewOptHdr(0);  
    pOoh->idObj = idObj;  
    pOoh->bLIType = bLIType;  
    int r = SendMessageToOmap(OMAP_API_CMD_GET_OBJ, pOoh,  
    sizeof(tagOApiObjectOptHdr));  
    free(pOoh);  
    return r;  
}
```

示例：GetOmapObject(idObj);//对象 id

发送消息：SendMessageToOmap(OMAP_API_CMD_GET_OBJ, pOoh, sizeof(tagOApiObjectOptHdr));

接收消息：WindowProc(UINT message, WPARAM wParam, LPARAM lParam)

处理消息：

```
if (message == WM_COPYDATA && wParam == OMAP_WIN_API_FLAG)  
{  
    COPYDATASTRUCT* pcd = (COPYDATASTRUCT*)lParam;  
    int nAppId = (pcd->dwData >> 16)&0xffff;  
    int nCmdId = pcd->dwData & 0xffff;  
    if (nCmdId == OMAP_API_CMD_GET_OBJ)  
    {  
        tagOApiObjectOptHdr* pOoh = (tagOApiObjectOptHdr*)pcd->lpData;  
        CString strInfo;  
        if (pOoh->iType == OMAP_OBJ_TYPE_GROUP)//文件夹对象  
        {  
            tagOApiObjGroupV1* pGroup = g_omapApi.GetOmapGroupFromBuf(pOoh + 1,  
            pcd->cbData - sizeof(tagOApiObjectOptHdr));  
            strInfo.AppendFormat(_T("name = %s\r\n"), Utf8ToString(pGroup->strName));//对象名称  
            strInfo.AppendFormat(_T("child = %d\r\n"), pGroup->nChild);//文件夹对象子对象个数  
            g_omapApi.FreeOmapObject(pGroup, OMAP_OBJ_TYPE_GROUP);//释放指针  
        }  
    }  
    else if (pOoh->iType == OMAP_OBJ_TYPE_SIGN)//标签对象  
    {  
        tagOApiMapSignV1* pSign = g_omapApi.GetOmapSignFromBuf(pOoh + 1, pcd->cbData -  
        sizeof(tagOApiObjectOptHdr));
```

```
strInfo.AppendFormat(_T("name = %s\r\n"), Utf8ToCString(pSign->strName));
strInfo.AppendFormat(_T("latlng = %lf,%lf\r\n"), pSign->lat, pSign->lng);
g_omapApi.FreeOmapObject(pSign, OMAP_OBJ_TYPE_SIGN);
}

else if (pOoh->iType == OMAP_OBJ_TYPE_TRACK)//轨迹对象
{
    tagOApiMapTrackV1* pTrack = g_omapApi.GetOmapTrackFromBuf(pOoh + 1, pcd->cbData
- sizeof(tagOApiObjectOptHdr));
    strInfo.AppendFormat(_T("name = %s\r\n"), Utf8ToCString(pTrack->strName));
    strInfo.AppendFormat(_T("TrackPointCount = %d\r\n"), pTrack->nMtp);
    g_omapApi.FreeOmapObject(pTrack, OMAP_OBJ_TYPE_TRACK);
}

else if (pOoh->iType == OMAP_OBJ_TYPE_SHAPE)//图形对象
{
    tagOApiMapShapeV1* pShape = g_omapApi.GetOmapShapeFromBuf(pOoh + 1,
pcd->cbData - sizeof(tagOApiObjectOptHdr));
    strInfo.AppendFormat(_T("name = %s\r\n"), Utf8ToCString(pShape->strName));
    strInfo.AppendFormat(_T("VertexCount = %d\r\n"), pShape->nMtp);
    g_omapApi.FreeOmapObject(pShape, OMAP_OBJ_TYPE_SHAPE);
}

else if (pOoh->iType == OMAP_OBJ_TYPE_DIR)//导航路线对象
{
    tagOApiMapDirectionsV1* pDir = g_omapApi.GetOmapDirFromBuf(pOoh + 1, pcd->cbData
- sizeof(tagOApiObjectOptHdr));
    strInfo.AppendFormat(_T("name = %s\r\n"), Utf8ToCString(pDir->strName));
    strInfo.AppendFormat(_T("nUnit = %d\r\n"), pDir->nUnit);
    g_omapApi.FreeOmapObject(pDir, OMAP_OBJ_TYPE_DIR);
}

else
{
    strInfo = L"不支持的对象";
}

m_edResult.SetWindowText(strInfo);
}
```

2.5.9. 设置收藏夹对象

调用函数： SetObjectToOmap(idParent);

2.5.9.1. 往指定目录添加一个文件夹

```
SetObjectToOmap(int idParent)
{
    CString strName = L"我的文件夹"; //文件夹名称
    tagOApiObjGroupV1* pGroup = g_omapApi.NewOmapGroup();
    strncpy_s(pGroup->strName, CStringToUtf8(strName, strUtf8, sizeof(strUtf8)), sizeof(pGroup->strName));
    int r = g_omapApi.SetOmapObject(0, pGroup, idParent, 0);
    g_omapApi.FreeOmapObject(pGroup, OMAP_OBJ_TYPE_GROUP);
    AlertErrSendOmapMsg(r);
    if (r <= 0)
        return;
}
```

调用函数： SetOmapObject(DWORD idObj, tagOApiObjGroupV1* pGroup, DWORD idParent, int iPosition) “

示例： SetOmapObject(0, pGroup, idParent, 0)

发送消息： SendMessageToOmap(OMAP_API_CMD_SET_OBJ, pOoh, sizeof(tagOApiObjectOptHdr) + iExtLen);

2.5.9.2. 往指定目录添加一个标签

```
SetObjectToOmap(int idParent)
{
    strName = L"我的标签";
    tagOApiMapSignV1* pSign = g_omapApi.NewOmapSign();
    strncpy_s(pSign->strName, CStringToUtf8(strName, strUtf8, sizeof(strUtf8)),
    sizeof(pSign->strName))pSign->pstrComment = OApiNewObjectComment("abcd1234");//设置备注内容
    pSign->lat = 40.1111;//经度
    pSign->lng = 116.2222;//纬度
    pSign->iSignPic = 1;//标签样式
    pSign->iTxtType = 1;//显示标签名称 (0: 不显示, 2: 显示带边框名称 3: 自定义风格)
    r = g_omapApi.SetOmapObject(0, pSign, idParent, 0);
    g_omapApi.FreeOmapObject(pSign, OMAP_OBJ_TYPE_SIGN);
    AlertErrSendOmapMsg(r);
    if (r <= 0)
        return;
}
```

调用函数： SetOmapObject(DWORD idObj, tagOApiMapSignV1* pSign, DWORD idParent, int iPosition)

示例： SetOmapObject(0, pSign, idParent, 0)

发送消息： SendMessageToOmap(OMAP_API_CMD_SET_OBJ, pOoh, sizeof(tagOApiObjectOptHdr) + iExtLen);

2.5.9.3. 往指定目录添加一条轨迹

```
SetObjectToOmap(int idParent)
{
    strName = L"我的轨迹";
    tagOApiMapTrackV1* pTrack = g_omapApi.NewOmapTrack();
    strncpy_s(pTrack->strName, CStringToUtf8(strName, strUtf8, sizeof(strUtf8)), sizeof(pTrack->strName));
    strName = L"中文备注";
    pTrack->pstrComment = OApiNewObjectComment(CStringToUtf8(strName, strUtf8, sizeof(strUtf8)));
    pTrack->mtd.dwLineClr = RGB(0, 224, 224);
    pTrack->nMtp = 100;//轨迹点数量
    pTrack->pMtp = (tagOApiMapTrackPointV1*)OApiZeroMalloc(sizeof(tagOApiMapTrackPointV1) *
pTrack->nMtp);
    for (int i = 0; i < pTrack->nMtp; i++)
    {
        pTrack->pMtp[i].lat = 40.1122 + i*0.001;
        pTrack->pMtp[i].lng = 116.2223 + i*(i*0.1)*0.002;
    }
    r = g_omapApi.SetOmapObject(0, pTrack, idParent, 0);
    g_omapApi.FreeOmapObject(pTrack, OMAP_OBJ_TYPE_TRACK);
    AlertErrSendOmapMsg(r);
    if (r <= 0)
        return;
}
```

调用函数： SetOmapObject(DWORD idObj, tagOApiMapTrackV1* pTrack, DWORD idParent, int iPosition)

示例： SetOmapObject(0, pTrack, idParent, 0)

发送消息： SendMessageToOmap(OMAP_API_CMD_SET_OBJ, pOoh, sizeof(tagOApiObjectOptHdr) + iExtLen);

2.5.9.4. 往指定目录添加一个图形

```
SetObjectToOmap(int idParent)
{
    strName = L"我的图形";
    tagOApiMapShapeV1* pShape = g_omapApi.NewOmapShape();
    strncpy_s(pShape->strName, CStringToUtf8(strName, strUtf8, sizeof(strUtf8)), sizeof(pShape->strName));
    strName = L"abcdefg中文备注";
    pShape->pstrComment = OApiNewObjectComment(CStringToUtf8(strName, strUtf8, sizeof(strUtf8)));
    pShape->nMtp = 10;
    pShape->pMtp = (tagOApiMapTrackPointV1*)OApiZeroMalloc(sizeof(tagOApiMapTrackPointV1) *
```

```
pShape->nMtp);
for (int i = 0; i < pShape->nMtp; i++)
{
    pShape->pMtp[i].lat = 40.1112 + i*0.001;
    pShape->pMtp[i].lng = 116.2223 + i*(i*0.1)*0.002;
}
r = g_omapApi.SetOmapObject(0, pShape, idParent, 0);
g_omapApi.FreeOmapObject(pShape, OMAP_OBJ_TYPE_SHAPE);
AlertErrSendOmapMsg(r);
if (r <= 0)
    return;
}
```

调用函数： SetOmapObject(DWORD idObj, tagOApiMapShapeV1* pTrack, DWORD idParent, int iPosition)

示例： SetOmapObject(0, pShape, idParent, 0)

发送消息： SendMessageToOmap(OMAP_API_CMD_SET_OBJ, pOoh, sizeof(tagOApiObjectOptHdr) + iExtLen);

2.5.9.5. 往指定目录添加一个路线

```
SetObjectToOmap(int idParent)
{
    strName = L"我的路线";
    tagOApiMapDirectionsV1* pDir = g_omapApi.NewOmapDir();
    strncpy_s(pDir->strName, CStringToUtf8(strName, strUtf8, sizeof(strUtf8)), sizeof(pDir->strName));
    strName = L"abcdefg中文备注ffss";
    pDir->pstrComment = OApiNewObjectComment(CStringToUtf8(strName, strUtf8, sizeof(strUtf8)));
    strcpy_s(pDir->strTitle, "test dir");
    pDir->nUnit = 10;
    pDir->dirSrh.iDirMode = 0;
    pDir->dirSrh.mpStart.lat = 40.099;
    pDir->dirSrh.mpStart.lng = 116.099;
    pDir->punit = (tagOApiMapDirectionsUnitV1*)OApiZeroMalloc(sizeof(tagOApiMapDirectionsUnitV1) *
pDir->nUnit);
    double lat = 40.1;
    double lng = 116.1;
    for (int i = 0; i < pDir->nUnit; i++)
    {
        sprintf_s(pDir->punit[i].strTitle, "step%d", i + 1);
        if (i == 0)
        {
            strcpy_s(pDir->punit[i].strSubTitle[0], "substep1");
        }
    }
}
```

```
strcpy_s(pDir->punit[i].strSubTitle[1], "substep2");
pDir->punit[i].nSubCount = 2;
}
pDir->punit[i].nLl = 20;
pDir->punit[i].llList = (tagOApiLatLngV1*)OApiZeroMalloc(sizeof(tagOApiLatLngV1) *
pDir->punit[i].nLl);
int a = rand() % 10 + 1;
int b = rand() % 10 + 1;
for (int j = 0; j < pDir->punit[i].nLl; j++)
{
    if (j > 0)
    {
        lat += a / 1000.0;
        lng += b / 1000.0;
    }
    pDir->punit[i].llList[j].lat = lat;
    pDir->punit[i].llList[j].lng = lng;
}
pDir->dirSrh.mpEnd.lat = lat + 0.001;
pDir->dirSrh.mpEnd.lng = lng + 0.001;
r = g_omapApi.SetOmapObject(0, pDir, idParent, 0);
g_omapApi.FreeOmapObject(pDir, OMAP_OBJ_TYPE_DIR);
AlertErrSendOmapMsg(r);
}
```

调用函数： SetOmapObject(DWORD idObj, tagOApiMapDirectionsV1* pTrack, DWORD idParent, int iPosition)

示例： SetOmapObject(0, pDir, idParent, 0)

发送消息： SendMessageToOmap(OMAP_API_CMD_SET_OBJ, pOoh, sizeof(tagOApiObjectOptHdr) + iExtLen);

2.5.10. 删除收藏夹对象

调用函数： DelOmapObject(DWORD* pidList, int nId, bool bNoGroup, bool bForceDel)

// bNoGroup为真时，不删除最顶层的本身文件夹

// bForceDel 为真时会强行删除，忽略只读属性

示例： g_omapApi.DelOmapObject(pdwObj, nId); //pdwObj:对象 id 数组， nId:删除数量

发送消息： SendMessageToOmap(OMAP_API_CMD_DEL_OBJ, pOoh, sizeof(tagOApiObjectOptHdr) + nId * 4);

2.5.11. 设置临时对象

调用函数： SetObjectToOmap(OMAP_TMP_OBJ_GROUP_ID); //临时分组结点 OMAP_TMP_OBJ_GROUP_ID
将对象添加到临时分组结点中，可直接清空临时对象。其余功能和添加收藏夹对象相同。

2.5.12. 清空临时对象

调用函数： CleanOmapTmpObject();

发送消息： SendMessageToOmap(OMAP_WIN_CMD_CLEAN_TMP_OBJ, NULL, 0);

2.5.13. 获取奥维端选中的收藏夹对象

调用函数：

GetOmapSelectList(int iFlag)

```
{  
    // iFlag支持的参数为  
    // OBJ_SELECT_CHECK_FLAG_GROUP_JOIN 1 选中对象，选中的文件夹自动包含所有子对象  
    //OBJ_SELECT_CHECK_FLAG_SEL_ONLY 2 选中对象，选中的文件夹不自动包括非选中子对象  
    //BJ_SELECT_CHECK_FLAG_NO_GROUP 3 选中对象，如果选中的是文件夹，则排除文件夹本身去  
    //选中子结点  
    int iExtLen = sizeof(int);  
    tagOApiObjectOptHdr* pOoh = OApiNewOptHdr(iExtLen);  
    int *piFlag = (int*)(pOoh + 1);  
    *piFlag = iFlag;  
    int r = SendMessageToOmap(OMAP_API_CMD_GET_SEL_LIST, pOoh,  
    sizeof(tagOApiObjectOptHdr)+iExtLen);  
    free(pOoh);  
    return r;  
}
```

示例： GetOmapSelectList(3); //若选中对象为文件夹，则只获取文件夹子对象。//可获取对象名称、id、类型
发送消息：

SendMessageToOmap(OMAP_API_CMD_GET_SEL_LIST, pOoh, sizeof(tagOApiObjectOptHdr)+iExtLen);

接收消息： WindowProc(UINT message, WPARAM wParam, LPARAM lParam)

```
处理消息：if (message == WM_COPYDATA && wParam == OMAP_WIN_API_FLAG)
{
    COPYDATASTRUCT* pcd = (COPYDATASTRUCT*)lParam;
    int nAppId = (pcd->dwData >> 16)&0xffff;
    int nCmdId = pcd->dwData & 0xffff;
    if (nCmdId == OMAP_API_CMD_GET_SEL_LIST)
    {
        if (pcd->cbData == 0)
        {
            MessageBox(_T("当前奥维地图绑定用户不是VIP用户，不支持此API接口"));
            return TRUE;
        }
        BYTE* pData = (BYTE*)pcd->lpData;
        tagOApiItemListV1* pOlv = (tagOApiItemListV1*)(pData + sizeof(tagOApiObjectOptHdr));
        int nList = (pcd->cbData - sizeof(tagOApiObjectOptHdr)) / sizeof(tagOApiItemListV1);
        for (int i = 0; i < nList; i++)
        { //数据处理过程
            CString strName;
            DWORD id;
            if (pOlv[i].iType == OMAP_OBJ_TYPE_GROUP)
            {
                strName = pOlv[i].strName; //文件夹对象名称
                id = pOlv[i].idObj; //文件夹对象id
            }
            else if (pOlv[i].iType == OMAP_OBJ_TYPE_SIGN)
            {
                strName = pOlv[i].strName; //标签对象名称
                id = pOlv[i].idObj; //标签对象id
            }
            else if (pOlv[i].iType == OMAP_OBJ_TYPE_TRACK)
            {
                strName = pOlv[i].strName; //轨迹对象名称
                id = pOlv[i].idObj; //轨迹对象id
            }
            else if (pOlv[i].iType == OMAP_OBJ_TYPE_SHAPE)
            {
                strName = pOlv[i].strName; //图形对象名称
                id = pOlv[i].idObj; //图形对象id
            }
            else if (pOlv[i].iType == OMAP_OBJ_TYPE_DIR)
            {
                strName = pOlv[i].strName; //导航线路名称
            }
        }
    }
}
```

```
    id = pOlv[i].idObj; //导航线路ids
}
}
}
```

2.5.14. 添加工具条到奥维

```
if(g_dlgToolBox == NULL) //如果工具条对象不存在
{
    g_dlgToolBox = new DlgToolbox();
    g_dlgToolBox->Create(g_dlgToolBox->IDD); //创建一个新的工具条对象
}
g_dlgToolBox->>ShowWindow(SW_SHOW); //显示工具条对象
```

工具条详细功能在查看 `DlgToolBox.cpp` 代码后开发。

2.5.15. 添加工具栏自定义按钮

nId 取值范围 1-10，工具条最多还差 10 个按钮

```
void COmapInterfaceDlg::OnBnClickedBtnSetCusMainBtn()
{
    char *pData = (char *)OApiZeroMalloc(3000 * 2);
    //设置第一个图标
    tagCusMainBtn *pCmb = (tagCusMainBtn*)pData;
    pCmb->nId = 1;
    strcpy_s(pCmb->strTxt, "aaa");
    //图标规格：16*16的png
    FILE *pfile = fopen("./res\\btn1.png", "rb");
    char strBuf[3000] = { 0 };
    int nlen1 = fread(strBuf, 1, sizeof(strBuf), pfile);
    fclose(pfile);
    pCmb->nPngLen = nlen1;
    memcpy((char*)(pCmb + 1), strBuf, nlen1);
    //设置第二个图标
    pCmb = (tagCusMainBtn*)(pData + sizeof(tagCusMainBtn) + nlen1);
    pCmb->nId = 2;
    strcpy_s(pCmb->strTxt, "bbb");
    pfile = fopen("./res\\btn2.png", "rb");
    char strBuf2[3000] = { 0 };
    int nlen2 = fread(strBuf2, 1, sizeof(strBuf2), pfile);
    fclose(pfile);
    pCmb->nPngLen = nlen2;
    memcpy((char*)(pCmb + 1), strBuf2, nlen2);

    g_omapApi.SetCusMainBtn((tagCusMainBtn*)pData, sizeof(tagCusMainBtn) * 2 + nlen1 + nlen2);
    free(pData);
}
```

2.5.16. 删除工具栏自定义按钮

```
void COmapInterfaceDlg::OnBnClickedBtnDelCusMainBtn()
{
    int ids[2] = { 1, 2 };
    g_omapApi.DelCusMainBtn(ids, 2);
}
```

2.5.17. 查询工具栏自定义按钮

```
void COmapInterfaceDlg::OnBnClickedBtnQueryCusMainBtn()
{
    int ids[2] = { 1, 2 };
    g_omapApi.QueryCusMainBtn(ids, 2);
}
```

2.5.18. 添加收藏夹对象自定义菜单项

nId 取值范围 1-10，最多支持添加 10 个菜单项。重复添加将覆盖原菜单项

```
void COmapInterfaceDlg::OnBnClickedBtnSetCusFavMenu()
{
    char *pData = (char *)OApiZeroMalloc(3000 * 2);
    //设置第一个菜单
    tagCusFavMenu *pCfm = (tagCusFavMenu*)pData;
    pCfm->nId = 1;
    strcpy_s(pCfm->strTxt, "AAA");
    //设置第二个菜单
    pCfm = (tagCusFavMenu*) (pData + sizeof(tagCusFavMenu));
    pCfm->nId = 2;
    strcpy_s(pCfm->strTxt, "BBB");
    g_omapApi.SetCusFavMenu((tagCusFavMenu*)pData, sizeof(tagCusFavMenu) * 2);
    free(pData);
}
```

2.5.19. 删除收藏夹对象自定义菜单项

```
void COmapInterfaceDlg::OnBnClickedBtnDelCusFavMenu()
{
    int ids[2] = { 1, 2 }; //删除id为1, 2的菜单项
    g_omapApi.DelCusFavMenu(ids, 1);
}
```

2.5.20. 查询收藏夹对象自定义菜单项

```
void COmapInterfaceDlg::OnBnClickedBtnQueryCusFavMenu() {  
    int ids[2] = { 1, 2 }; //查询id为1, 2的菜单项  
    g_omapApi.QueryCusFavMenu(ids, 1);  
}
```

2.5.21. 设置自定义弹窗状态

```
#define OAPI_MAP_OBJ_EVENT_NONE      0      //显示奥维默认弹窗  
#define OAPI_MAP_OBJ_EVENT_SEND_ONLY  1      //只显示自定义弹窗，对象本身备注弹窗不显示  
#define OAPI_MAP_OBJ_EVENT_SEND_EXT   2      //显示默认弹窗，并通知接口  
  
int a = 0;  
void COmapInterfaceDlg::OnBnClickedBtnSetObjEvent()  
{  
    if (a % 3 == 0)  
    {  
        g_omapApi.SetObjEvent(OAPI_MAP_OBJ_EVENT_NONE);  
    }  
    else if (a % 3 == 1)  
    {  
        g_omapApi.SetObjEvent(OAPI_MAP_OBJ_EVENT_SEND_ONLY);  
    }  
    else if (a % 3 == 2)  
    {  
        g_omapApi.SetObjEvent(OAPI_MAP_OBJ_EVENT_SEND_EXT);  
    }  
    a++;  
}
```

2.5.22. 设置弹窗位置和信息

当奥维自定义弹窗状态设为 1 时，
点击奥维互动地图浏览器地图上的对象时，自动调用
若不设置 cx,cy，则弹窗默认自适应宽度和高度

```
/* 可以直接赋值
char *strHtml = "<?ovital_ct name=\"ovital_sample\">{\
\"user\": \"Andy\", \
\"birthday\" : \"\", \
\"sex\" : \"man\", \
\"Marital_status\" : false, \
\"work_address\" : \"0\"\
}";
*/
//读取本地文件赋值
FILE *pfile = fopen("./\\res\\test.html", "rb");
char strBuf[3000] = { 0 };
int nlen = fread(strBuf, 1, sizeof(strBuf), pfile);
char *strHtml = strBuf;

if (_m_iObjEventType == OAPI_MAP_OBJ_EVENT_SEND_ONLY)
{
    testShowWnd(pObjEvent);
}

void testShowWnd(tagOApiMapObjEvent *pObjEvent, BYTE bCloseType, BYTE bCloseEvent)
{
    int nlen = strlen(strHtml);
    tagOApiMapObjShowWnd *pObjShowWnd = (tagOApiMapObjShowWnd
*)OApiZeroMalloc(sizeof(tagOApiMapObjShowWnd)+nlen+16);
    pObjShowWnd->pMapWnd = pObjEvent->pClickOfMap;
    pObjShowWnd->sizeObjWnd.cx = 400;
    pObjShowWnd->sizeObjWnd.cy = 500;
    pObjShowWnd->idObj = pObjEvent->idObj;
    pObjShowWnd->bCloseType = bCloseType;
    pObjShowWnd->bCloseEvent = bCloseEvent;
    pObjShowWnd->pHtml = (char *) (pObjShowWnd+1);
    strcpy(pObjShowWnd->pHtml, strHtml);
    pObjShowWnd->nHtml = nlen;
    g_omapApi.SetObjShowWnd(pObjShowWnd);
    free(pObjShowWnd);
}
```

2.5.23. 添加工具栏自定义菜单

```
//说明：id 范围：1-100;  
//如果 id 为-n，则表示接下来的 n 个 Menu 为子菜单；  
例：  
// [-2, 交通工具],[1, 汽车],[2, 火车],[-2, 颜色],[3,红色],[4,蓝色], [-2, 天气],[5, 晴天],[6, 阴天],[7,人员]  
菜单项：交通工具->汽车、火车  
    颜色->红色、蓝色  
    天气->晴天、阴天  
    人员  
void COmapInterfaceDlg::OnBnClickedBtnSetCusmainmenu()  
{  
    int nId = 1;  
    int index = 0;  
    tagCusMainMenu Menus[10];//添加10个菜单  
    Menus[index].nId = -2;  
    char strUtf8[100];  
    CString strTitle = _T("交通工具");  
    CStringToUtf8(strTitle, strUtf8, sizeof(strUtf8));  
  
    strcpy(Menu[index++].strTxt, strUtf8);  
    Menus[index].nId = nId++;  
    strTitle = _T("汽车");  
    strcpy(Menu[index++].strTxt, CStringToUtf8(strTitle, strUtf8, sizeof(strUtf8)));  
    Menus[index].nId = nId++;  
    strTitle = _T("火车");  
    strcpy(Menu[index++].strTxt, CStringToUtf8(strTitle, strUtf8, sizeof(strUtf8)));  
    Menus[index].nId = -2;  
    strTitle = _T("颜色");  
    strcpy(Menu[index++].strTxt, CStringToUtf8(strTitle, strUtf8, sizeof(strUtf8)));  
    Menus[index].nId = nId++;  
    strTitle = _T("红色");  
    strcpy(Menu[index++].strTxt, CStringToUtf8(strTitle, strUtf8, sizeof(strUtf8)));  
    Menus[index].nId = nId++;  
    strTitle = _T("蓝色");  
    strcpy(Menu[index++].strTxt, CStringToUtf8(strTitle, strUtf8, sizeof(strUtf8)));  
    Menus[index].nId = -2;  
    strTitle = _T("天气");  
    strcpy(Menu[index++].strTxt, CStringToUtf8(strTitle, strUtf8, sizeof(strUtf8)));  
    Menus[index].nId = nId++;  
    strTitle = _T("晴天");  
    strcpy(Menu[index++].strTxt, CStringToUtf8(strTitle, strUtf8, sizeof(strUtf8)));
```

```
Menus[index].nId = nId++;
strTitle = _T("阴天");
strcpy(Menu[index].strTxt, CStringToUtf8(strTitle, strUtf8, sizeof(strUtf8)));
Menus[index].nId = nId++;
strTitle = _T("人员");
strcpy(Menu[index].strTxt, CStringToUtf8(strTitle, strUtf8, sizeof(strUtf8)));
g_omapApi.SetCusMainMenu(Menu, index);
}
```

2.5.24. 工具栏自定义功能性菜单项被点击

被点击功能性菜单项菜单项 nId

```
if (nCmdId == OMAP_API_CMD_CUS_MAIN_MENU_CLICK)
{
    if (pcd->cbData < sizeof(tagOApiRevHdr))
    {
        return TRUE;
    }
    int nMenu = (pcd->cbData - sizeof(tagOApiRevHdr)) / sizeof(tagCusMainMenu);
    tagCusMainMenu *pMenu = (tagCusMainMenu*)((char*)pcd->lpData + sizeof(tagOApiRevHdr));
    CString strInfo;
    for (int i = 0; i < nMenu; i++)
    {
        strInfo.AppendFormat(TEXT("Clicked MenuId:%d:\r\n"), pMenu[i].nId);
    }

    m_strResult = strInfo;
    UpdateData(FALSE);
}
```

2.5.25. 删除工具栏自定义菜单

```
void COmapInterfaceDlg::OnBnClickedBtnDelCusmainmenu2()
{
    g_omapApi.SetCusMainMenu(nullptr, 0);
}
```

2.5.26. 获取轨迹基本属性

调用函数：

```
void COmapInterfaceDlg::OnBnClickedBtnGettrackdetail()
{
    int nSelect = GetListSelectedItem(&m_lcResult);
    if (nSelect < 0)
    {
        MessageBox(_T("请选择对象"));
        return;
    }
    CString strType = m_lcResult.GetItemText(nSelect, 0);
    if (strType != TEXT("轨迹"))
    {
        MessageBox(_T("请选择轨迹对象"));
        return;
    }
    DWORD idObj = (DWORD)m_lcResult.GetItemData(nSelect);
    int r = g_omapApi.GetTrackDetail(idObj);
    AlertErrSendOmapMsg(r);
}
```

处理消息：

```
if (nCmdId == OMAP_API_CMD_GET_TRACK_DETAIL)
{
    if (pcd->cbData != sizeof(tagOApiRevHdr)+sizeof(tagOApiMapTrackDetail))
    {
        return TRUE;
    }
    tagOApiMapTrackDetail *pTrackDetail = (tagOApiMapTrackDetail*)((char*)pcd->lpData +
    sizeof(tagOApiRevHdr));
    CString strInfo;
    strInfo.AppendFormat(TEXT("idObj:%d;\r\n"), pTrackDetail->idObj);
    if (pTrackDetail->idObj > 0)
    {
        strInfo.AppendFormat(TEXT("tStart:%d;\r\n"), pTrackDetail->tStart);
        strInfo.AppendFormat(TEXT("tEnd:%d;\r\n"), pTrackDetail->tEnd);
        strInfo.AppendFormat(TEXT("TotalPoint:%d;\r\n"), pTrackDetail->iTotalPoint);
        strInfo.AppendFormat(TEXT("TotalDist:%d;\r\n"), pTrackDetail->iTotalDist);
        strInfo.AppendFormat(TEXT("AvgSpeed:%d;\r\n"), pTrackDetail->iAvgSpeed);
        strInfo.AppendFormat(TEXT("MaxSpeed:%d;\r\n"), pTrackDetail->iMaxSpeed);
        strInfo.AppendFormat(TEXT("MaxAltitude:%d;\r\n"), pTrackDetail->iMaxAltitude);
        strInfo.AppendFormat(TEXT("MinAltitude:%d;\r\n"), pTrackDetail->iMinAltitude);
    }
}
```

```
        }
        m_strResult = strInfo;
        UpdateData(FALSE);
    }
```

2.5.27. 获取图形基本信息

```
void COmapInterfaceD1g::OnBnClickedBtnGetshapedetail()
{
    int nSelect = GetListSelectedItem(&m_lcResult);
    if (nSelect < 0)
    {
        MessageBox(_T("请选择对象"));
        return;
    }
    CString strType = m_lcResult.GetItemText(nSelect, 0);
    if (strType !=TEXT("图形"))
    {
        MessageBox(_T("请选择图形对象"));
        return;
    }
    DWORD idObj = (DWORD)m_lcResult.GetItemData(nSelect);
    int r = g_omapApi.GetShapeDetail(idObj);
    AlertErrSendOmapMsg(r);
}

处理消息:
if (nCmdId == OMAP_API_CMD_GET_SHAPE_DETAIL)
{
    if (pcd->cbData != sizeof(tagOApiRevHdr) + sizeof(tagOApiMapShapeDetail))
    {
        return TRUE;
    }
    tagOApiMapShapeDetail *pShapeDetail = (tagOApiMapShapeDetail*)((char*)pcd->lpData +
sizeof(tagOApiRevHdr));
    CString strInfo;
    strInfo.AppendFormat(TEXT("idObj:%d;\r\n"), pShapeDetail->idObj);
    if (pShapeDetail->idObj > 0)
    {
        strInfo.AppendFormat(TEXT("PointCnt:%d;\r\n"), pShapeDetail->iPointCnt);
        strInfo.AppendFormat(TEXT("AreaCnt:%d;\r\n"), pShapeDetail->iAreaCnt);
        strInfo.AppendFormat(TEXT("Perimeter:%f(公里);\r\n"), pShapeDetail->dPerimeter);
        strInfo.AppendFormat(TEXT("AreaSize:%f(平方米);\r\n"), pShapeDetail->dAreaSize);
    }
}
```

```
        }
        m_strResult = strInfo;
        UpdateData(FALSE);
    }
```

2.5.28. 设置系统坐标系参数

```
void COmapInterfaceDlg::OnBnClickedBtnSetCoordProj()
{
    //CoordType 2 经纬度(如果要设置经纬度坐标, 不需要其他参数); 4 横轴墨卡托
    //iPrjType enum eWGS_1984=0, eXian_1980=1, eKrasovsky_1940=2, eUTM=3, eCGCS2000=4, eCustomType=5
    tag0ApiPrj *pPrj = (tag0ApiPrj *)OApiZeroMalloc(sizeof(tag0ApiPrj));
    pPrj->iCoordType = 4;
    pPrj->iPrjType = 2;
    pPrj->dFalseEasting = 500000;
    pPrj->dFalseNorthing = 0;
    pPrj->dScaleFactor = 1;
    pPrj->dCentralMeridian = 105;
    pPrj->dAngularUnit = 0.5;
    pPrj->dLinearUnit = 1.0;
    pPrj->dSemimajorAxis = 6378245.0;
    pPrj->dInverseFlattening = 298.3;

    int r = SetSysCoordProj(pPrj);
    free(pPrj);
    AlertErrSendOmapMsg(r);

}
```

2.5.29. 获取系统坐标系参数

```
int OMapApi::GetSysCoordProj()
{
    return g_omapApi.SendMessageToOmap(OMAP_API_CMD_GET_COORD_PROJ, NULL, 0);
}

处理消息:
if (nCmdId == OMAP_API_CMD_GET_COORD_PROJ)
{
    if (pcd->cbData != sizeof(tag0ApiPrj))
    {
        return TRUE;
    }
}
```

```
    tag0ApiPrj* pPrj = (tag0ApiPrj*)pcd->lpData;
    CString strInfo;
    strInfo.AppendFormat(TEXT("strPrjType:%s\r\n"), Utf8ToCString(pPrj->strPrjType));
    strInfo.AppendFormat(TEXT("strPrjName:%s\r\n"), Utf8ToCString(pPrj->strPrjName));
    strInfo.AppendFormat(TEXT("strGeogcs:%s\r\n"), Utf8ToCString(pPrj->strGeogcs));
    strInfo.AppendFormat(TEXT("strDatum:%s\r\n"), Utf8ToCString(pPrj->strDatum));
    strInfo.AppendFormat(TEXT("strSpheroid:%s\r\n"), Utf8ToCString(pPrj->strSpheroid));
    strInfo.AppendFormat(TEXT("dSemimajorAxis:%f\r\n"), pPrj->dSemimajorAxis);
    strInfo.AppendFormat(TEXT("dSemiminorAxis:%f\r\n"), pPrj->dSemiminorAxis);
    strInfo.AppendFormat(TEXT("dInverseFlattening:%f\r\n"), pPrj->dInverseFlattening);
    strInfo.AppendFormat(TEXT("strPrimem:%s\r\n"), Utf8ToCString(pPrj->strPrimem));
    strInfo.AppendFormat(TEXT("dPrimeMeridian:%f\r\n"), pPrj->dPrimeMeridian);
    strInfo.AppendFormat(TEXT("strAngularUnit:%s\r\n"), Utf8ToCString(pPrj->strAngularUnit));
    strInfo.AppendFormat(TEXT("dAngularUnit:%f\r\n"), pPrj->dAngularUnit);
    strInfo.AppendFormat(TEXT("strLinearUnit:%s\r\n"), Utf8ToCString(pPrj->strLinearUnit));
    strInfo.AppendFormat(TEXT("dLinearUnit:%f\r\n"), pPrj->dLinearUnit);
    strInfo.AppendFormat(TEXT("strProjection:%s\r\n"), Utf8ToCString(pPrj->strProjection));
    strInfo.AppendFormat(TEXT("dFalseEasting:%f\r\n"), pPrj->dFalseEasting);
    strInfo.AppendFormat(TEXT("dFalseNorthing:%f\r\n"), pPrj->dFalseNorthing);
    strInfo.AppendFormat(TEXT("dCentralMeridian:%f\r\n"), pPrj->dCentralMeridian);
    strInfo.AppendFormat(TEXT("dScaleFactor:%f\r\n"), pPrj->dScaleFactor);
    strInfo.AppendFormat(TEXT("dLatitudeOfOrigin:%f\r\n"), pPrj->dLatitudeOfOrigin);
    strInfo.AppendFormat(TEXT("dLatitudeOfCenter:%f\r\n"), pPrj->dLatitudeOfCenter);
    strInfo.AppendFormat(TEXT("dStandardParallel_1:%f\r\n"), pPrj->dStandardParallel_1);
    strInfo.AppendFormat(TEXT("dStandardParallel_2:%f\r\n"), pPrj->dStandardParallel_2);
    strInfo.AppendFormat(TEXT("dLatitude_Of_1st_Point:%f\r\n"),
pPrj->dLatitude_Of_1st_Point);
    strInfo.AppendFormat(TEXT("dLatitude_Of_2nd_Point:%f\r\n"), pPrj->dLinearUnit);
    strInfo.AppendFormat(TEXT("dLinearUnit:%f\r\n"), pPrj->dLatitude_Of_2nd_Point);
    strInfo.AppendFormat(TEXT("dLongitude_Of_1st_Point:%f\r\n"),
pPrj->dLongitude_Of_1st_Point);
    strInfo.AppendFormat(TEXT("dLongitude_Of_2nd_Point:%f\r\n"),
pPrj->dLongitude_Of_2nd_Point);
    strInfo.AppendFormat(TEXT("dAzimuth:%f\r\n"), pPrj->dAzimuth);
    strInfo.AppendFormat(TEXT("dLongitudeOfCenter:%f\r\n"), pPrj->dLongitudeOfCenter);
    strInfo.AppendFormat(TEXT("iPrjType:%d\r\n"), pPrj->iPrjType);
    strInfo.AppendFormat(TEXT("iCoordType:%d\r\n"), pPrj->iCoordType);
    m_strResult = strInfo;
    UpdateData(FALSE);
}
```

2.5.30. 获取奥维可显示区域范围

```
int OMapApi::GetOmapWndRect()
{
    return SendMessageToOmap(OMAP_API_CMD_GET_OMAP_WND_RECT, NULL, 0);
}

处理消息:

if (nCmdId == OMAP_API_CMD_GET_OMAP_WND_RECT)
{
    if (pcd->cbData != sizeof(tagOApiOmapWndRect))
    {
        return TRUE;
    }

    tagOApiOmapWndRect* pWndRect = (tagOApiOmapWndRect*)pcd->lpData;
    CString strInfo;
    strInfo.AppendFormat(TEXT("rectMapOfWnd:[l:%d, t:%d, r:%d, b:%d]\r\n"),
pWndRect->rectMapOfWnd. left,
pWndRect->rectMapOfWnd. top, pWndRect->rectMapOfWnd. right,
pWndRect->rectMapOfWnd. bottom);
    m_strResult = strInfo;
    UpdateData(FALSE);
}
```

2.5.31. 设置非特定对象自定义弹窗

```
void COmapInterfaceDlg::OnBnClickedBtnSetCmdCusWnd()
{
    CString strWndTitle = L"自定义窗体名称";
    int WndId = 1; //窗口id，取值范围1-10
    BYTE bCloseEvent = 1;
    int Left = 350;
    int Right = 750;
    int Top = 200;
    int Bottom = 800;
    testCusShowWnd(WndId, bCloseEvent, strWndTitle, strHtml, Left, Right, Top, Bottom);
}

处理消息:

if (nCmdId == OMAP_API_CMD_CUS_WND_SET || nCmdId == OMAP_API_CMD_CUS_WND_QUERY || nCmdId ==
OMAP_API_CMD_CUS_WND_DEL)
{
    CString strInfo;
    int iShowCusWndRet = 0;
```

```
memcpy(&iShowCusWndRet, pcd->lpData, sizeof(int));
if (iShowCusWndRet > 0)
{
    if (nCmdId == OMAP_API_CMD_CUS_WND_SET)
    {
        strInfo.Format(TEXT("SetCusWnd:%d"), iShowCusWndRet);
    }
    else if (nCmdId == OMAP_API_CMD_CUS_WND_QUERY)
    {
        strInfo.Format(TEXT("QueryCusWnd:%d"), iShowCusWndRet);
    }
    else
    {
        strInfo.Format(TEXT("DelCusWnd:%d"), iShowCusWndRet);
    }
}
else
{
    if (nCmdId == OMAP_API_CMD_CUS_WND_SET)
    {
        strInfo.Format(TEXT("SetCusWndFail"));
    }
    else if (nCmdId == OMAP_API_CMD_CUS_WND_QUERY)
    {
        strInfo.Format(TEXT("QueryCusWndFail"));
    }
    else
    {
        strInfo.Format(TEXT("DelCusWndFail"));
    }
}
m_strResult = strInfo;
UpdateData(FALSE);
}
```

2.5.32. 查询非特定对象自定义弹窗

```
void COmapInterfaceDlg::OnBnClickedBtnQueryCusWnd()
{
    int WndID = 1; //窗口id， 取值范围1-10
    SendMessageToOmap(OMAP_API_CMD_CUS_WND_QUERY, &idWnd, sizeof(int));
}

处理消息：
if (nCmdId == OMAP_API_CMD_CUS_WND_SET || nCmdId == OMAP_API_CMD_CUS_WND_QUERY || nCmdId ==
OMAP_API_CMD_CUS_WND_DEL)
{
    CString strInfo;
    int iShowCusWndRet = 0;
    memcpy(&iShowCusWndRet, pcd->lpData, sizeof(int));
    if (iShowCusWndRet > 0)
    {
        if (nCmdId == OMAP_API_CMD_CUS_WND_SET)
        {
            strInfo.Format(TEXT("SetCusWnd:%d"), iShowCusWndRet);
        }
        else if (nCmdId == OMAP_API_CMD_CUS_WND_QUERY)
        {
            strInfo.Format(TEXT("QueryCusWnd:%d"), iShowCusWndRet);
        }
        else
        {
            strInfo.Format(TEXT("DelCusWnd:%d"), iShowCusWndRet);
        }
    }
    else
    {
        if (nCmdId == OMAP_API_CMD_CUS_WND_SET)
        {
            strInfo.Format(TEXT("SetCusWndFail"));
        }
        else if (nCmdId == OMAP_API_CMD_CUS_WND_QUERY)
        {
            strInfo.Format(TEXT("QueryCusWndFail"));
        }
        else
        {
            strInfo.Format(TEXT("DelCusWndFail"));
        }
    }
}
```

```
        }  
    }  
    m_strResult = strInfo;  
    UpdateData(FALSE);  
  
}
```

2.5.33. 删除非特定对象自定义弹窗

```
void C0mapInterfaceDlg::OnBnClickedBtnDelCusWnd()  
{  
    int WndID = 1; //窗口id，取值范围1-10  
    SendMessageTo0map(OMAP_API_CMD_CUS_WND_DEL, &idWnd, sizeof(int));  
}  
  
处理消息：  
if (nCmdId == OMAP_API_CMD_CUS_WND_SET || nCmdId == OMAP_API_CMD_CUS_WND_QUERY || nCmdId ==  
OMAP_API_CMD_CUS_WND_DEL)  
{  
    CString strInfo;  
    int iShowCusWndRet = 0;  
    memcpy(&iShowCusWndRet, pcd->lpData, sizeof(int));  
    if (iShowCusWndRet > 0)  
    {  
        if (nCmdId == OMAP_API_CMD_CUS_WND_SET)  
        {  
            strInfo.Format(TEXT("SetCusWnd:%d"), iShowCusWndRet);  
        }  
        else if (nCmdId == OMAP_API_CMD_CUS_WND_QUERY)  
        {  
            strInfo.Format(TEXT("QueryCusWnd:%d"), iShowCusWndRet);  
        }  
        else  
        {  
            strInfo.Format(TEXT("DelCusWnd:%d"), iShowCusWndRet);  
        }  
    }  
    else  
    {  
        if (nCmdId == OMAP_API_CMD_CUS_WND_SET)  
        {  
            strInfo.Format(TEXT("SetCusWndFail"));  
        }  
    }  
}
```

```
    else if (nCmdId == OMAP_API_CMD_CUS_WND_QUERY)
    {
        strInfo.Format(TEXT("QueryCusWndFail"));
    }
    else
    {
        strInfo.Format(TEXT("DelCusWndFail"));
    }
}
m_strResult = strInfo;
UpdateData(FALSE);
}
```

2.5.34. 获取奥维自定义地图列表

```
void COmapInterfaceDlg::OnBnClickedBtnGetCusMapCfg()
{
    g_omapApi.SendMessageToOmap(OMAP_API_CMD_GET_CUS_MAP_CFG, NULL, 0)
}

处理消息:
if (nCmdId == OMAP_API_CMD_GET_CUS_MAP_CFG)
{
    if (pcd->cbData != sizeof(tag0ApiCustomMap))
    {
        return TRUE;
    }
    tag0ApiCustomMap* pCustomMap = (tag0ApiCustomMap*)pcd->lpData;
    CString strInfo;
    strInfo.AppendFormat(TEXT("idMap:%d\r\n"), pCustomMap->idMap);
    strInfo.AppendFormat(TEXT("idLayerBig:%d\r\n"), pCustomMap->idLayerBig);
    strInfo.AppendFormat(TEXT("idLayerSmall:%d\r\n"), pCustomMap->idLayerSmall);
    strInfo.AppendFormat(TEXT("iLayerBigFlag:%u\r\n"), pCustomMap->iLayerBigFlag);
    strInfo.AppendFormat(TEXT("iLayerSmallFlag:%u\r\n"), pCustomMap->iLayerSmallFlag);
    strInfo.AppendFormat(TEXT("iMinLevel:%d\r\n"), pCustomMap->iMinLevel);
    strInfo.AppendFormat(TEXT("iMaxLevel:%d\r\n"), pCustomMap->iMaxLevel);
    strInfo.AppendFormat(TEXT("idLayerSmall:%d\r\n"), pCustomMap->iCoordType);
    strInfo.AppendFormat(TEXT("iTileFormat:%d\r\n"), pCustomMap->iTileFormat);
    strInfo.AppendFormat(TEXT("iTileSize:%d\r\n"), pCustomMap->iTileSize);
    strInfo.AppendFormat(TEXT("iVersion:%u\r\n"), pCustomMap->iVersion);
    strInfo.AppendFormat(TEXT("bUseLayer:%d\r\n"), pCustomMap->bUseLayer);
    strInfo.AppendFormat(TEXT("bIsSatellite:%d\r\n"), pCustomMap->bIsSatellite);
```

```
    strInfo.AppendFormat(TEXT("strName:%s\r\n"), Utf8ToCString(pCustomMap->strName));
    strInfo.AppendFormat(TEXT("strHost:%s\r\n"), Utf8ToCString(pCustomMap->strHost));
    strInfo.AppendFormat(TEXT("wPort:%u\r\n"), pCustomMap->wPort);
    strInfo.AppendFormat(TEXT("iHostStart:%d\r\n"), pCustomMap->iHostStart);
    strInfo.AppendFormat(TEXT("iHostEnd:%d\r\n"), pCustomMap->iHostEnd);
    strInfo.AppendFormat(TEXT("strUrl:%s\r\n"), Utf8ToCString(pCustomMap->strUrl));
    m_strResult = strInfo;
    UpdateData(FALSE);
}

}
```

2.5.35. 获取奥维 WebSocket 配置信息

```
void COmapInterfaceDlg::OnBnClickedBtnGetWebSocketCon()
{
    g_omapApi.SendMessageTo0map(OMAP_API_CMD_GET_WEB_SOCKET_CONF, NULL, 0)
}
```

处理消息:

```
if (nCmdId == OMAP_API_CMD_GET_WEB_SOCKET_CONF)
{
    if (pcd->cbData != sizeof(tag0ApiWebSocketConf))
    {
        return TRUE;
    }
    tag0ApiWebSocketConf* pWebSocketConf = (tag0ApiWebSocketConf*)pcd->lpData;
    CString strInfo;
    strInfo.AppendFormat(TEXT("dwWssUserLimitIP:%u\r\n"), pWebSocketConf->dwWssUserLimitIP);
    strInfo.AppendFormat(TEXT("wPort:%u\r\n"), pWebSocketConf->wPort);
    strInfo.AppendFormat(TEXT("strPsk:%s\r\n"), Utf8ToCString(pWebSocketConf->strPsk));
    strInfo.AppendFormat(TEXT("strSslCaFn:%s\r\n"),
    Utf8ToString(pWebSocketConf->strSslCaFn));
    strInfo.AppendFormat(TEXT("strSslCertFn:%s\r\n"),
    Utf8ToString(pWebSocketConf->strSslCertFn));
    strInfo.AppendFormat(TEXT("strSslKeyFn:%s\r\n"), pWebSocketConf->strSslKeyFn);
    strInfo.AppendFormat(TEXT("bHttpTileService:%d\r\n"), pWebSocketConf->bHttpTileService);
    m_strResult = strInfo;
    UpdateData(FALSE);
}
```

2.5.36. 移动奥维对象

```
void COmapInterfaceDlg::OnBnClickedBtnCmdMoveObj()
{
```

```
DWORD idParent = 738224536; //需要插入的位置节点id
int nIndexInParent = 0; //插入父节点对象内顺序位置 0表示插入到第一条
int nId = 7; //移动对象数量
DWORD pidList[7] = { 1633817490, 1606192095, 746456053, 2140764178, 1270509728, 691868557,
28138214 }; //需要移动的对象id数组

int iRet= g_omapApi.MoveOmapObject(idParent, nIndexInParent, pidList, nId);
AlertErrSendOmapMsg(iRet);

}
```