

Qplanetosm 二次开发手册

(ActiveX 接口部分)

QPlanetOSM 是基于 Qt 的 OpenStreetMap 地理信息控件，其主要接口如下：

目录

Qplanetosm 二次开发手册	1
(ActiveX 接口部分)	1
1、 远程服务配置	4
1.1 osm_get_remote_address 获取服务器地址	4
1.2 osm_set_remote_address 设置服务器地址	4
2、 本地缓存配置	6
2.1 osm_get_local_cache 获取本地缓存文件夹	6
2.2 osm_set_local_cache 设置本地缓存文件夹	6
2.3 osm_get_cache_expire_days 获取瓦片保质期	7
2.4 osm_set_cache_expire_days 设置瓦片保质期	7
2.5 osm_get_auto_download 获取自动下载状态	8
2.6 osm_set_auto_download 设置自动下载状态	8
3、 漫游	10
3.1 osm_get_level 获得当前缩放比例	10
3.2 osm_set_level 设置当前缩放比例	10
3.3 获得当前纬度/经度	11
3.3.1 osm_get_center_lat 函数原型	11
3.3.2 osm_get_center_lon 函数原型	11
3.3.3 对应界面设置	11
3.4 osm_set_center_pos 设置当前纬度/经度	11
3.5 osm_save_view 截图	11
4、 图层控制	13
4.1 枚举图层名称	13
4.1.1 osm_layer_get_count 获得图层个数	13
4.1.2 osm_layer_get_name 获得图层名称	13
4.2 图层可见性	13
4.2.1 osm_layer_get_visiable 获取可见性	13
4.2.2 osm_layer_set_visiable 设置可见性	13
4.3 图层活动性	14
4.3.1 osm_layer_get_active 获取活动性	14
4.3.2 osm_layer_set_active 设置活动性	14
4.4 图层顺序	14
4.4.1 osm_layer_move_up 图层上移	14

4.4.2	osm_layer_move_down 图层下移	14
4.4.3	osm_layer_move_top 图层置顶	14
4.4.4	osm_layer_move_bottom 图层到底	14
4.5	对应界面设置	15
5、	图层脚本呼叫与事件响应	16
5.1	osm_layer_call_function 脚本呼叫	16
5.2	evt_Message 图层事件响应	16
5.3	从字符串到词典的转换	16
5.3.1	C#转换函数	16
5.3.2	C++/Qt 转换函数	17
5.3.3	例子	18
6、	插件图层功能调用	19
6.1	量测插件 grid 功能	19
6.1.1	get_ruler_status 获取工具状态	19
6.1.2	set_ruler_status 设置工具状态	19
6.1.3	get_polygon 获取当前框选区域	20
6.2	标图插件 geomarker 功能	21
6.2.1	update_point 添加/更新点标记	21
6.2.2	update_line 添加/更新线标记	22
6.2.3	update_polygon 添加/更新多边形标记	23
6.2.4	update_icon 添加/更新图标标记	25
6.2.5	update_props 更新标签显示功能	26
6.2.6	exists 查找标记是否存在	27
6.2.7	delete_marks 删除标记	28
6.2.8	delete_props 删除某个标记的属性	29
6.2.9	mark_names 返回所有的标记名	30
6.2.10	mark 返回特定标记的所有信息	30
6.2.11	props 返回特定标记的标签	31
6.2.12	add_resource 添加资源（图标）	31
6.2.13	save_resources 保存资源（图标）	32
6.2.14	load_resources 读取资源（图标）	33
6.2.15	save_xml 保存标记	34
6.2.16	load_xml 读取标记	34
6.2.17	show_props 展开/隐藏所选图元的详细描述	34
6.2.18	props_vis 得到各个图元的详细信息展开状态	35
7、	控件事件参考	36
7.1	地图状态事件	36
7.1.1	WINDOW_CREATE 地图初始化事件	36
7.1.2	MAP_INITED 地图初始化完成事件	36
7.1.3	CONNECTION 服务连接事件	36
7.1.4	MAP_RESIZED 地图尺寸变化	36
7.2	用户漫游事件	37
7.2.1	MOUSE_XBUTTON_? 鼠标点击事件	37
7.2.2	MOUSE_WHELL 滚轮事件	37

7.2.3	MOUSE_MOVE 滚轮事件	38
7.2.4	CENTER_CHANGED 中心变更事件	38
7.3	LAYER_MOVED_XXX 图层顺序变更事件	38
7.4	ITEM_XBUTTON_XXXCLICKED geomarker 插件图元鼠标事件	39

1、 远程服务配置

1.1 osm_get_remote_address 获取服务器地址

函数原型:

方法名称:	osm_get_remote_address	
参数名	参数类型	说明
layerName	String	被询问的图层名称, 为界面选项卡的名称 (一般为 OSM)
返回值	String	当前的地址

范例:

```
string address = axqtaxviewer_planetosm.osm_get_remote_address("OSM");
```

对应界面设置:

1.2 osm_set_remote_address 设置服务器地址

函数原型:

方法名称:	osm_set_remote_address	
参数名	参数类型	说明
layerName	String	被设置的图层名称, 为界面选项卡的名称 (一般为 OSM)
addr	String	新的地址 使用字符串参数替换, %1 代表比例尺, %2 代表瓦片列, %3 代表行 (http://c.tile.openstreetmap.org/%1/%2/%3.png)
返回值	Void	

范例:

```
axqtaxviewer_planetosm.osm_set_remote_address(  
"OSM", " http://c.tile.openstreetmap.org/%1/%2/%3.png");
```

对应界面设置:

瓦片服务地址

☒ 1 联接并按需下载

本地缓存保质期 天

本地缓存文件夹

2



2、 本地缓存配置

2.1 osm_get_local_cache 获取本地缓存文件夹

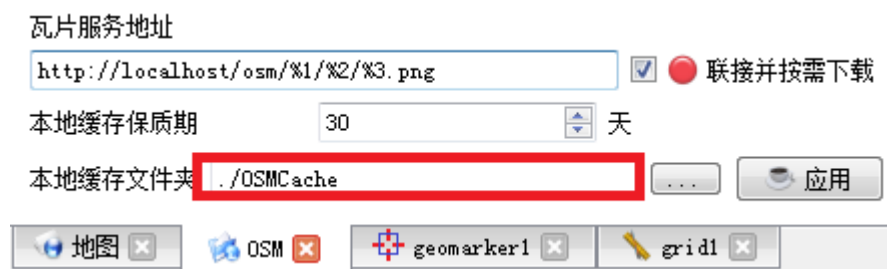
函数原型:

方法名称:	osm_get_local_cache	
参数名	参数类型	说明
layerName	String	被获取的图层名称, 为界面选项卡的名称 (一般为 OSM)
返回值	String	当前缓存文件夹位置

范例:

```
string address = axqtaxviewer_planetosm.osm_get_local_cache("OSM");
```

对应界面设置:



2.2 osm_set_local_cache 设置本地缓存文件夹

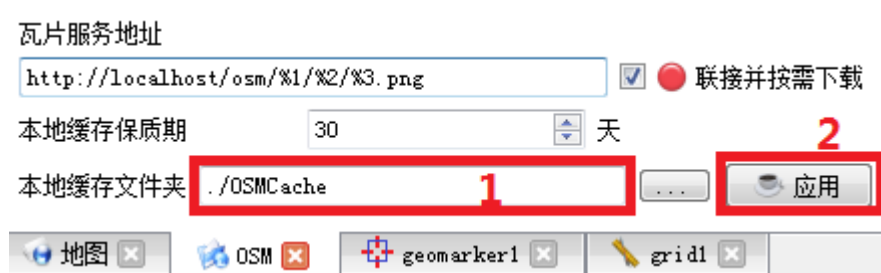
函数原型:

方法名称:	osm_set_local_cache	
参数名	参数类型	说明
layerName	String	被设置的图层名称, 为界面选项卡的名称 (一般为 OSM)
addr	String	新的地址
返回值	Void	

范例:

```
axqtaxviewer_planetosm.osm_set_local_cache("OSM", "/home/user/OSMCache");
```

对应界面设置:



2.3 osm_get_cache_expire_days 获取瓦片保质期

*瓦片保质期一旦越过，对应的瓦片会重新下载

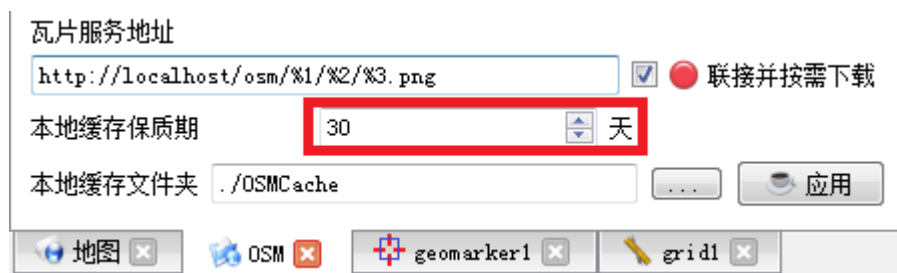
函数原型：

方法名称：	osm_get_cache_expire_days	
参数名	参数类型	说明
layerName	String	被获取的图层名称，为界面选项卡的名称（一般为 OSM）
返回值	int	当前保质期

范例：

```
int expdays = axqtaxviewer_planetosm.osm_get_cache_expire_days("OSM");
```

对应界面设置：



2.4 osm_set_cache_expire_days 设置瓦片保质期

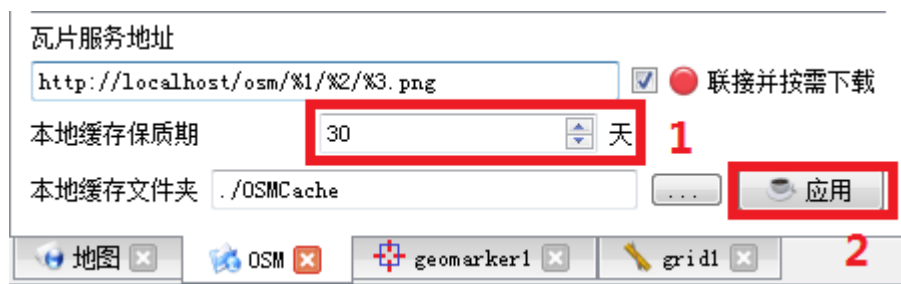
函数原型：

方法名称：	osm_set_cache_expire_days	
参数名	参数类型	说明
layerName	String	被设置的图层名称，为界面选项卡的名称（一般为 OSM）
days	int	新的保质期, 0 = 永不过期
返回值	Void	

范例：

```
axqtaxviewer_planetosm.osm_set_cache_expire_days("OSM", 120);
```

对应界面设置：



2.5 osm_get_auto_download 获取自动下载状态

* 自动下载选中时，如果本地缓存没有瓦片，会从服务器上下载。

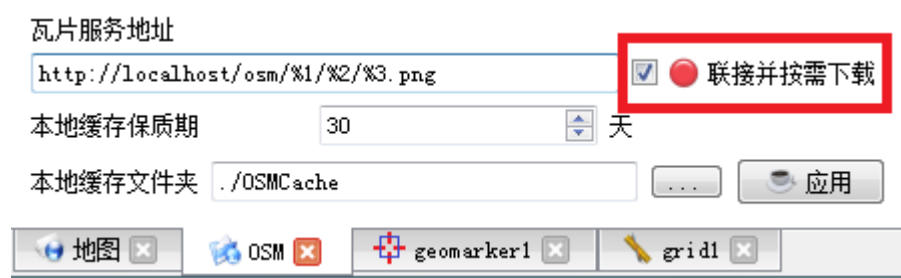
函数原型：

方法名称：	osm_get_auto_download	
参数名	参数类型	说明
layerName	String	被获取的图层名称，为界面选项卡的名称（一般为 OSM）
返回值	int	当前状态，0=不下载，-1=下载

范例：

```
int d = axqtaxviewer_planetosm.osm_get_auto_download("OSM");
```

对应界面设置：



2.6 osm_set_auto_download 设置自动下载状态

函数原型：

方法名称：	osm_set_auto_download	
参数名	参数类型	说明
layerName	String	被设置的图层名称，为界面选项卡的名称（一般为 OSM）
days	int	状态，0=不下载，-1=下载
返回值	Void	

范例：

```
axqtaxviewer_planetosm.osm_set_auto_download("OSM", -1);
```

对应界面设置：

瓦片服务地址

http://localhost/osm/%1/%2/%3.png

1

☒ ☐ 联接并按需下载

本地缓存保质期

30



天

本地缓存文件夹

./OSMCache



2

应用

地图

OSM

geomarker1

grid1

3、 漫游

3.1 osm_get_level 获得当前缩放比例

函数原型:

方法名称:	osm_get_level	
参数名	参数类型	说明
返回值	int	当前缩放比例(0~18)

范例:

```
int level = axqtaxviewer_planetosm.osm_get_level();
```

对应界面设置:



3.2 osm_set_level 设置当前缩放比例

函数原型:

方法名称:	osm_set_level	
参数名	参数类型	说明
P1	int	比例(0~18)
返回值	int	旧的比例

范例:

```
axqtaxviewer_planetosm.osm_set_level(12);
```

对应界面设置:



3.3 获得当前纬度/经度

3.3.1 osm_get_center_lat 函数原型

方法名称:	osm_get_center_lat	
参数名	参数类型	说明
返回值	double	单位为度

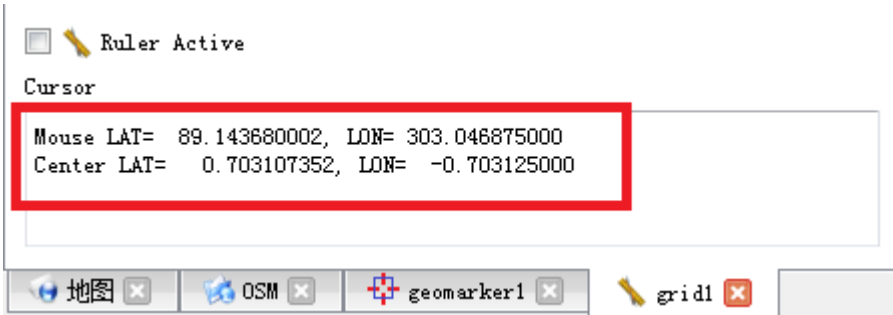
3.3.2 osm_get_center_lon 函数原型

方法名称:	osm_get_center_lat	
参数名	参数类型	说明
返回值	double	单位为度

范例:

```
double lat = axqtaxviewer_planetosm.osm_get_center_lat();  
double lon = axqtaxviewer_planetosm.osm_get_center_lon();
```

3.3.3 对应界面设置



3.4 osm_set_center_pos 设置当前纬度/经度

函数原型:

方法名称:	osm_set_center_pos	
参数名	参数类型	说明
Lat	Double	纬度, 单位为度
Lon	Double	经度, 单位为度
返回值	int	=1 表示成功, =0 失败

范例:

```
axqtaxviewer_planetosm.osm_set_center_pos(40.28236734, 101.3836464);
```

3.5 osm_save_view 截图

函数原型:

方法名称:	osm_save_view	
参数名	参数类型	说明

Filename	String	文件名，支持 BMP, PNG, JPEG
返回值	Int	0=失败，-1 为成功

范例：

```
int ret = axqtaxviewer_planetosm.osm_save_view("D:/1.jpg");
```

对应界面设置：



4、 图层控制

4.1 枚举图层名称

4.1.1 osm_layer_get_count 获得图层个数

方法名称:	osm_layer_get_count	
参数名	参数类型	说明
layName	String	图层名称, 为界面选项卡的名称
返回值	Int	图层个数

4.1.2 osm_layer_get_name 获得图层名称

方法名称:	osm_layer_get_count	
参数名	参数类型	说明
layName	String	图层名称, 为界面选项卡的名称
P1	Int	图层编号, 0~图层个数-1
返回值	String	图层名称

范例:

```
int layers = axqtaxviewer_planetosm.osm_layer_get_count();
for (int i = 0; i < layers; ++i)
{
    string layrname = axqtaxviewer_planetosm.osm_layer_get_name(i);
}
```

4.2 图层可见性

4.2.1 osm_layer_get_visiable 获取可见性

方法名称:	osm_layer_get_visiable	
参数名	参数类型	说明
layName	String	图层名称, 为界面选项卡的名称
返回值	Int	图层可见性,0=不可见, -1=可见

4.2.2 osm_layer_set_visiable 设置可见性

方法名称:	osm_layer_set_visiable	
参数名	参数类型	说明
layName	String	图层名称, 为界面选项卡的名称
v	Int	可见性,0=不可见, -1=可见
返回值	Int	设置后的可见性,0=不可见, -1=可见

4.3 图层活动性

4.3.1 osm_layer_get_active 获取活动性

方法名称:	osm_layer_get_active	
参数名	参数类型	说明
layName	String	图层名称, 为界面选项卡的名称
返回值	Int	图层活动性,0=不接收消息, -1=接收键鼠消息

4.3.2 osm_layer_set_active 设置活动性

方法名称:	osm_layer_set_active	
参数名	参数类型	说明
layName	String	图层名称, 为界面选项卡的名称
v	Int	活动性,0=不接收消息, -1=接收键鼠消息
返回值	Int	设置后的活动性,0=不接收消息, -1=接收键鼠消息

4.4 图层顺序

4.4.1 osm_layer_move_up 图层上移

方法名称:	osm_layer_move_up	
参数名	参数类型	说明
layName	String	图层名称, 为界面选项卡的名称
返回值	Int	-1=成功, 0=失败 (找不到对应的图层)

4.4.2 osm_layer_move_down 图层下移

方法名称:	osm_layer_move_down	
参数名	参数类型	说明
layName	String	图层名称, 为界面选项卡的名称
返回值	Int	-1=成功, 0=失败 (找不到对应的图层)

4.4.3 osm_layer_move_top 图层置顶

方法名称:	osm_layer_move_top	
参数名	参数类型	说明
layName	String	图层名称, 为界面选项卡的名称
返回值	Int	-1=成功, 0=失败 (找不到对应的图层)

4.4.4 osm_layer_move_bottom 图层到底

方法名称:	osm_layer_move_bottom	
参数名	参数类型	说明
layName	String	图层名称, 为界面选项卡的名称
返回值	Int	-1=成功, 0=失败 (找不到对应的图层)

4.5 对应界面设置



5、 图层脚本呼叫与事件响应

5.1 osm_layer_call_function 脚本呼叫

函数原型:

方法名称:	osm_layer_call_function	
参数名	参数类型	说明
layName	String	图层名称, 为界面选项卡的名称
args	String	输入参数集合
返回值	String	结果参数集合

*参数集合是使用分号(;)分割的 “键=值” 集合, 类似:

“function=XXX; lat=XXX; lon=XXX;...;”

各个键值对的顺序无关

调用注意事项:

- 1、调用者应该在 UI 线程
- 2、调用行为是阻塞的
- 3、各个图层/插件支持不同的方法, 详见各个图层的方法说明

范例:

```
string cmd = "function=set_ruler_status;status=-1;";  
res = axqtaxviewer_planetosm.osm_layer_call_function("grid1", cmd);
```

5.2 evt_Message 图层事件响应

方法名称:	evt_Message	
参数名	参数类型	说明
args	String	结果参数集合

*参数集合是使用分号(;)分割的 “键=值” 集合, 类似:

“name=XXX; destin=XXX;source=XXX;...;”

5.3 从字符串到词典的转换

为了在外部程序中方便的使用键-值集合, 我们可以使用一段小代码, 把字符串中的键值集合转换为词典, 提供两个函数, 当然, 用户也可自己写。

5.3.1 C#转换函数

```
Dictionary<string, string> string2map(string s)  
{
```

```

Dictionary<string, string> map = new Dictionary<string, string>();
char[] charSeparators = new char[] { ';' };
string[] result = s.Split(charSeparators, StringSplitOptions.RemoveEmptyEntries);
foreach (string p in result)
{
    char[] chars2 = new char[] { '=' };
    string[] kv = p.Split(chars2, StringSplitOptions.None);
    if (kv.Count() > 1)
    {
        try
        {
            map.Add(kv[0], kv[1]);
        }
        catch (ArgumentException)
        {
            addMsg(4, "An element with Key = \"" + kv[0] + "\" already exists.");
        }
    }
}
return map;
}

```

5.3.2 C++/Qt 转换函数

```

#include <QMap>
#include <QString>
#include < QStringList >
QMap<QString, QVariant> string_to_map(const QString & s)
{
    QMap<QString, QVariant> res;
    QStringList lst = s.split(";");
    foreach (QString s, lst)
    {
        int t = s.indexOf("=");
        if (t>0 && t< s.size())
        {
            QString name = s.left(t).trimmed();
            QString value = s.mid(t+1).trimmed();
            res[name] = value;
        }
    }
    return res;
}

```

5.3.3 例子

```
string res = axqtaxviewer_planetosm.osm_layer_call_function("geomarker1",  
                                                            "function=mark_names;");  
  
Dictionary<string, string> dp = string2map(res);
```

6、 插件图层功能调用

6.1 量测插件 grid 功能

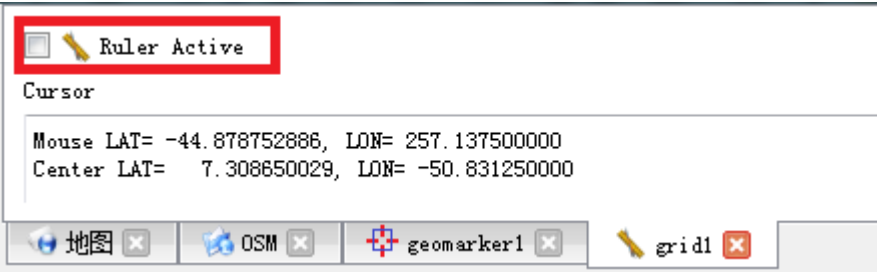
6.1.1 get_ruler_status 获取工具状态

范例呼叫:

```
string res =  
axqtaxviewer_planetosm.osm_layer_call_function  
(  
"grid1", "function=get_ruler_status;"  
);
```

呼叫参数表			
键名	键意义	取值	说明
function	函数名称	get_ruler_status	获取当前量测状态
返回参数表			
键名	键意义	取值	说明
status	状态值	0 或者-1	0 为关闭, 1 为打开

对应界面元素



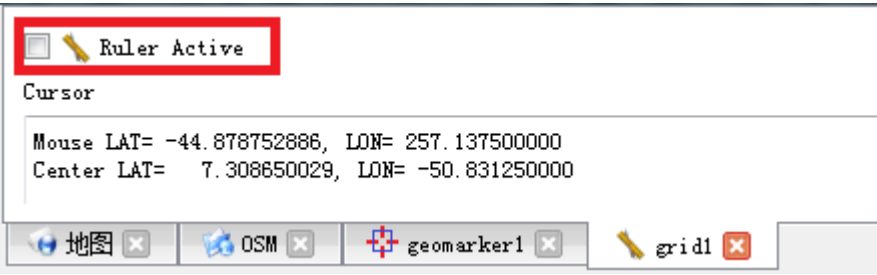
6.1.2 set_ruler_status 设置工具状态

范例呼叫:

```
string cmd = "function=set_ruler_status;status=-1;";  
res = axqtaxviewer_planetosm.osm_layer_call_function("grid1", cmd);
```

呼叫参数表			
键名	键意义	取值	说明
function	函数名称	set_ruler_status	获取当前量测状态
status	新状态	0, -1	0=关闭, -1 为打开
返回参数表			
键名	键意义	取值	说明
status	当前状态值	0 或者-1	0 为关闭, 1 为打开

对应界面元素



6.1.3 get_polygon 获取当前框选区域

范例呼叫:

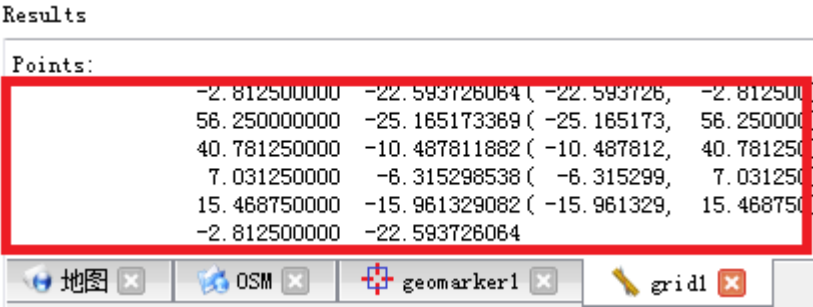
```
string res = axqtaxviewer_planetosm.osm_layer_call_function  
("grid1", "function=get_polygon;");
```

呼叫参数表			
键名	键意义	取值	说明
function	函数名称	get_polygon	获取当前量测状态
返回参数表			
键名	键意义	取值	说明
size	顶点数目	0~N	0 表示空
lat0	第一个顶点纬度		
lon0	第一个顶点经度		
lat1	第二个顶点纬度		
lon1	第二个顶点经度		
.....			
lat< size-1>	第 size 个顶点纬度		
lon< size-1>	第 size 个顶点经度		

范例返回:

```
"lat0=-22.5937260639293;lat1=-25.1651733686639;lat2=-10.4878118820567;lat3=-6.315298538  
33003;lat4=-15.9613290815966;lat5=-22.5937260639293;lon0=-2.8125;lon1=56.25;lon2=40.781  
25;lon3=7.03125;lon4=15.46875;lon5=-2.8125;size=6;"
```

对应界面元素



6.2 标图插件 geomarker 功能

6.2.1 update_point 添加/更新点标记

范例呼叫：

```
string res = axqtaxviewer_planetosm.osm_layer_call_function("geomarker1",  
  
    "function=update_point;name=哈哈;type=1;" +  
  
    "lat=35.2;lon=101.783;" +  
  
    "style_pen=2;color_pen=0,0,255,128;width_pen=3;" +  
  
    "style_brush=1;color_brush=0,255,0,128;" +  
  
    "color_label=0,0,255,96;weight_label=99;size_label=12;" +  
  
    "width=16;height=20;");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	update_point	添加点标记
name*	点 ID	自定义	必须唯一，否则会覆盖同名点
type	标记类型	1,2	1=椭圆点(default) 2=矩形点
lat*	点的经度		
lon*	点的纬度		
style_pen	线类型	0~6	参考界面的线选项
color_pen	线色	字符串	r,g,b,alpha 红,绿,蓝,透明度
width_pen	线宽	>=0	为像素宽
style_brush	填充类型	0~14	参考界面的填充选项
color_brush	填充色	字符串	r,g,b,alpha 红,绿,蓝,透明度
color_label	字体颜色	字符串	r,g,b,alpha 红,绿,蓝,透明度
weight_label	字体粗细	0~99	0~99, 99 最粗
size_label	字号	0~N	字的大小
width	符号宽度	0~N	像素宽度
height	符号高度	0~N	像素高度
返回参数表			
键名	键意义	取值	说明
error	错误信息		如果不存在，说明成功

对应界面元素

标记

	Name	
1	PTO	RECT

纬度 12 **lat**

经度 34 **lon**

像素宽 8

像素高 8

width, height

☒ 矩形标记点 **type=1**

☐ 椭圆标记点 **type=2**

属性

width pen

笔像素宽 3

style pen

笔风格 DashLine

color pen

笔色彩 201, 3, 255, 128

color brush

填充色彩 81, 255, 7, 233

color label

文本色彩 244, 2, 67, 25

weight label

文本粗细 99

style brush

填充风格 DenselfPattern

size label

字号 9

Name	Value

点标记 删除

连线标记 **更新**

区域标记 保存XML

图标标记 加载XML

ID PTO **name**

最后

属性名 LABEL 删除

属性值 please input label here 更新

地图

OSM

geomarker1

grid1

6.2.2 update_line 添加/更新线标记

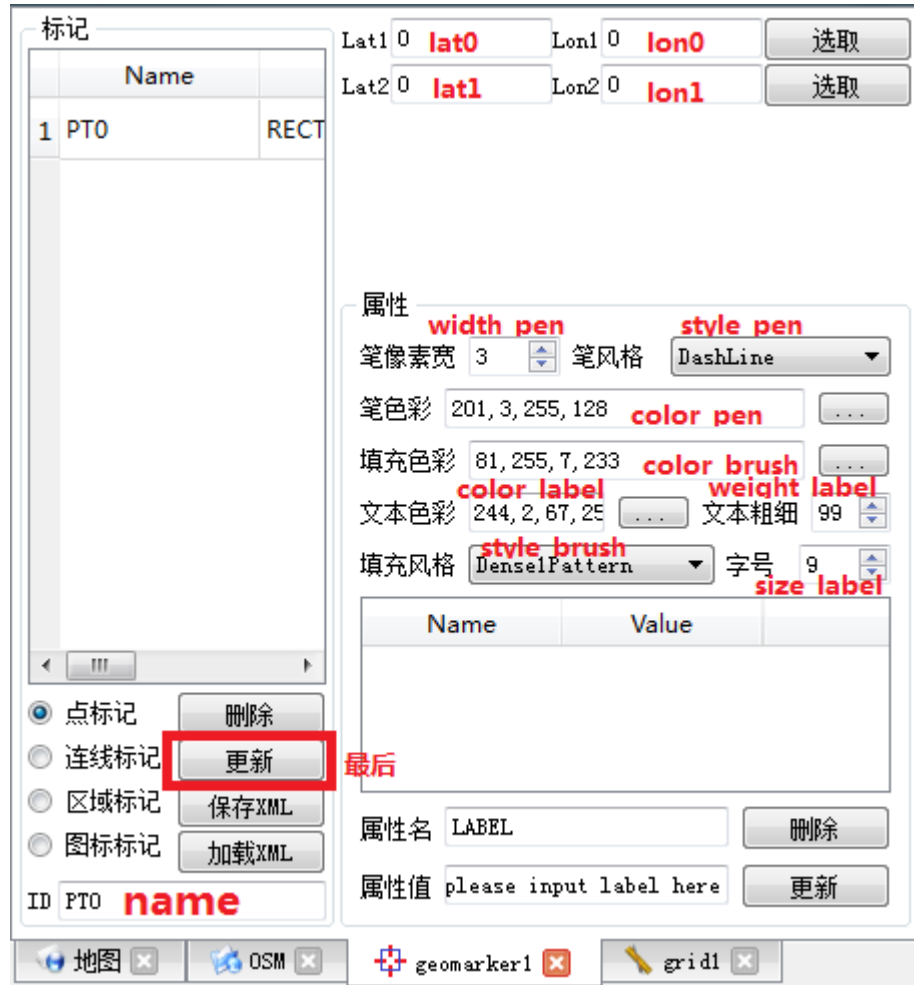
范例呼叫：

```
string res = axqtaxviewer_planetosm.osm_layer_call_function("geomarker1",
    "function=update_line;name=ID3;type=3;" +
    "lat0=40;lon0=98;lat1=12;lon1=144;" +
    "style_pen=4;color_pen=255,0,0,96;width_pen=2;");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	update_line	添加线标记
name*	点 ID	自定义	必须唯一，否则会覆盖同名点
type	标记类型	3	3=线(default)
lat0*	起点的经度		
lon0*	起点的纬度		
lat1*	终点的经度		
lon1*	终点的纬度		
style_pen	线类型	0~6	参考界面的线选项
color_pen	线色	字符串	r,g,b,alpha 红,绿,蓝,透明度

width_pen	线宽	>=0	为像素宽
color_label	字体颜色	字符串	r,g,b,alpha 红,绿,蓝,透明度
weight_label	字体粗细	0~99	0~99, 99 最粗
size_label	字号	0~N	字的大小
返回参数表			
键名	键意义	取值	说明
error	错误信息		如果不存在, 说明成功

对应界面元素



6.2.3 update_polygon 添加/更新多边形标记

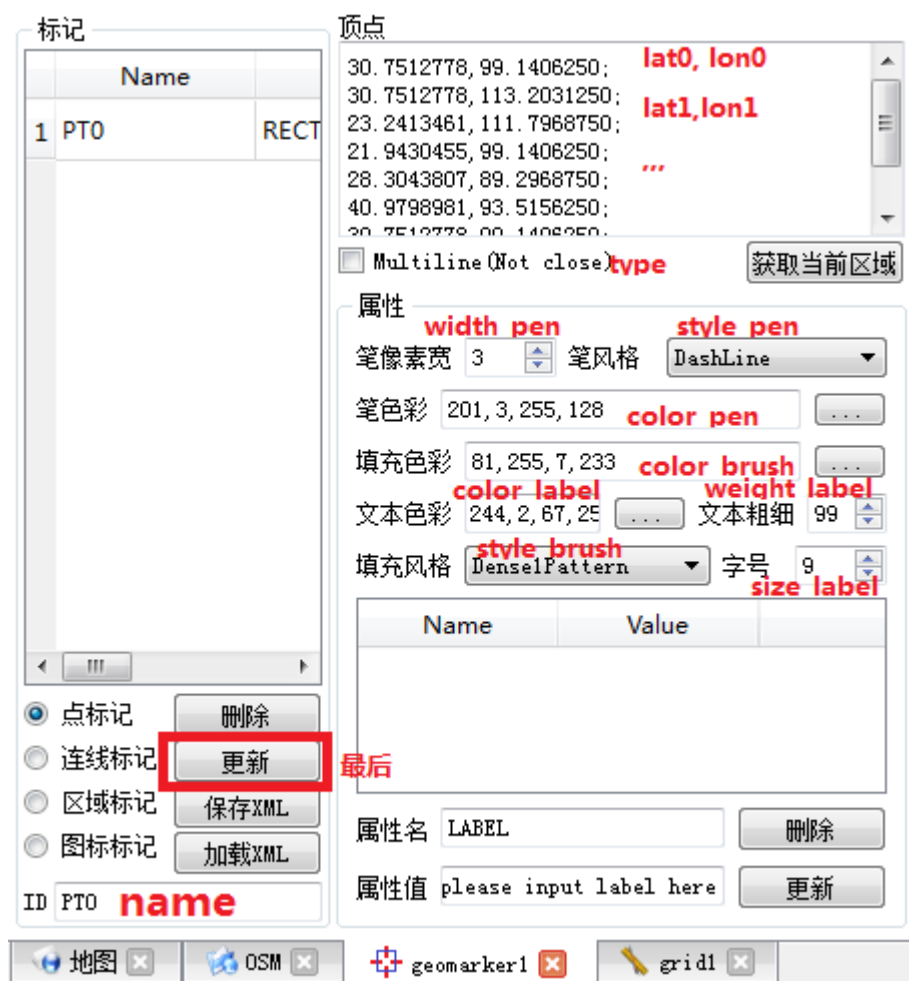
范例呼叫:

```
string res = axqtaxviewer_planetosm.osm_layer_call_function("geomarker1",  
    "function=update_polygon;name=ID4;type=4;"+  
    "lat0=12.2;lon0=67.3;"+  
    "lat1=14.3;lon1=62.8;"+  
    "lat2=22.7;lon2=66.5;"+  
    "lat3=11.5;lon3=72.2;"+  
    "lat4=10.8;lon4=69.4;"+
```

```
"style_pen=2;color_pen=0,0,255,128;" +
"width_pen=3;style_brush=1;color_brush=0,255,0,128;");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	update_polygon	添加多边形标记
name*	点 ID	自定义	必须唯一，否则会覆盖同名点
type	标记类型	4,6	4 = 多边形（封闭） 6 = 分段连线（开放）
lat0*	顶点 1 的经度		
lon0*	顶点 1 的纬度		
lat1*	顶点 2 的经度		
lon1*	顶点 2 的纬度		
lat2*	顶点 3 的经度		
lon2*	顶点 3 的纬度		
...			
style_pen	线类型	0~6	参考界面的线选项
color_pen	线色	字符串	r,g,b,alpha 红,绿,蓝,透明度
width_pen	线宽	>=0	为像素宽
style_brush	填充类型	0~14	参考界面的填充选项
color_brush	填充色	字符串	r,g,b,alpha 红,绿,蓝,透明度
color_label	字体颜色	字符串	r,g,b,alpha 红,绿,蓝,透明度
weight_label	字体粗细	0~99	0~99, 99 最粗
size_label	字号	0~N	字的大小
返回参数表			
键名	键意义	取值	说明
error	错误信息		如果不存在，说明成功

对应界面元素



6.2.4 update_icon 添加/更新图标标记

范例呼叫：

```
res = axqtaxviewer_planetosm.osm_layer_call_function("geomarker1",
                                                    "function=update_icon;name=lenna;" +
                                                    "lat=12.347364;lon=107.3736438;" +
                                                    "icon=lenna;scale=1.2;rotate=12;smooth=1;" +
                                                    +"color_label=0,0,255,96;weight_label=99;size_label=9;");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	update_icon	添加图标标记
name*	点 ID	自定义	必须唯一，否则会覆盖同名点
type	标记类型	5	5= 图标 6 = 分段连线（开放）
lat*	经度		
lon*	纬度		
icon	图标名	自定	需要与已有图标名对应
scale	缩放因子	>0	=1 为原始大小，<1 缩小 >1 放大

rotate	旋转因子	0-360	旋转（顺时针）
smooth	平滑边缘	0,1	0=不平滑，1=平滑
color_label	字体颜色	字符串	r,g,b,alpha 红,绿,蓝,透明度
weight_label	字体粗细	0~99	0~99, 99 最粗
size_label	字号	0~N	字的大小
返回参数表			
键名	键意义	取值	说明
error	错误信息		如果不存在，说明成功

对应界面元素

6.2.5 update_props 更新标签显示功能

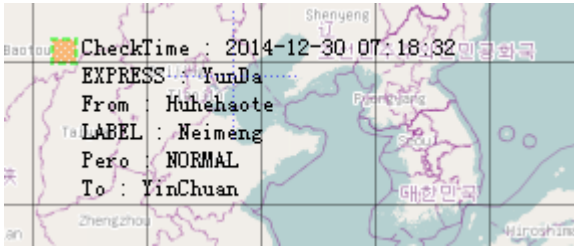
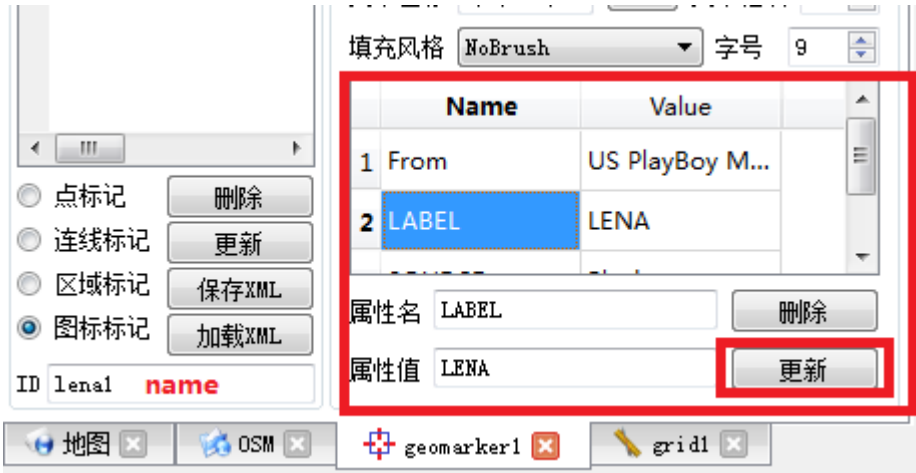
范例呼叫：

```
res = axqtaxviewer_planetosm.osm_layer_call_function("geomarker1",  
"function=update_props;name=ID2;" +  
"LABEL=Neimeng;EXPRESS=YunDa;Pero=NORMAL;" +  
"CheckTime=2014-12-30 07:18:32;" +  
"From=Huhehaote;To=YinChuan");
```

呼叫参数表(*为可选)

键名	键意义	取值	说明
function *	函数名称	update_props	更新标签显示
name*	点 ID	自定义	必须唯一，否则会覆盖同名点
其他键值自定义	自定义	自定义	不要与关键词 function, name 重名
返回参数表			
键名	键意义	取值	说明
error	错误信息		如果不存在，说明成功

对应界面元素



6.2.6 exists 查找标记是否存在

范例呼叫:

```
res = axqtaxviewer_planetosm.osm_layer_call_function("geomarker1",
"function=exists;name=ID2;");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	exists	更新标签显示
name*	点 ID	自定义	必须唯一，否则会覆盖同名点
返回参数表			
键名	键意义	取值	说明
return	结果	0,1	0=不存在 1=存在

对应界面元素

标记		
	Name	
1	ID1	ELLIP
2	ID2	RECT
3	ID4	POLY
4	PT0	RECT
5	lena1	ICON

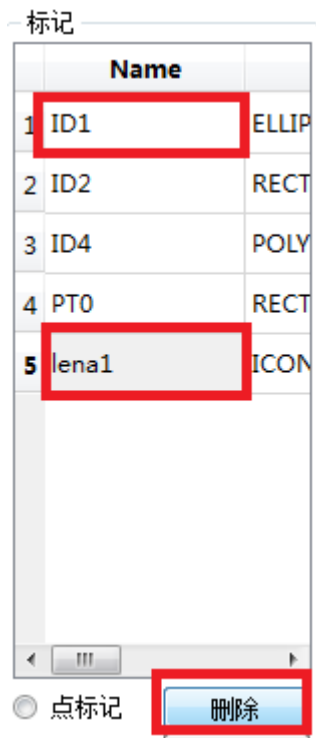
6.2.7 delete_marks 删除标记

范例呼叫：

```
string res = axqtaxviewer_planetasm.osm_layer_call_function("geomarker1",
"function=delete_marks;name0=ID1;name1=ID2;name2=lena1");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	delete_marks	删除标记
name0	第一个待删 ID		如果不存在，则全删除
name1	第二个待删 ID		
.....			
返回参数表			
键名	键意义	取值	说明
name0	错误信息	0,1	0=未删除，1=删除
name1	错误信息	0,1	0=未删除，1=删除
...*如果在呼叫参数里没有指定 name,则会删除全部，返回 ALL=1			

对应界面元素



6.2.8 delete_props 删除某个标记的属性

范例呼叫：

```
string res = axqtaxviewer_planetasm.osm_layer_call_function("geomarker1",  
"function=delete_props;name=ID1;prop0=TIME;prop1=Profit;Prop2=Address;Pr  
op3=tel;");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	delete_props	删除标记
name0	第一个待删 ID		如果不存在，则全删除
name1	第二个待删 ID		
.....			
返回参数表			
键名	键意义	取值	说明
name0	成功删除	0,1	0=未删除，1=删除
name1	成功删除	0,1	0=未删除，1=删除
... *如果在呼叫参数里没有指定 name,则会删除全部，返回 ALL=1			

对应界面元素

Name	Value
1 From	US Playboy M...
SOURCE	Playboy

属性名

属性值

6.2.9 mark_names 返回所有的标记名

范例呼叫：

```
string res = axqtaxviewer_planetasm.osm_layer_call_function("geomarker1",
    "function=mark_names;");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	mark_names	删除标记
返回参数表			
键名	键意义	取值	说明
name0	第一个标记名		
name1	第二个标记名		
...			
name<N-1>	第 N 个标记名		

6.2.10 mark 返回特定标记的所有信息

范例呼叫：

```
string res = axqtaxviewer_planetasm.osm_layer_call_function
("geomarker1", "function=mark;name=LENA;");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	mark	删除标记
name *	查询的标记名		
返回参数表			
键名	键意义	取值	说明
name	点 ID		
type	标记类型	1,2	1=椭圆点(default) 2=矩形点
lat	点的经度		
lon	点的纬度		
lat0	经度 1		
lon0	纬度 1		

lat1	经度 2		
lon1	纬度 2		
...			
style_pen	线类型	0~6	参考界面的线选项
color_pen	线色	字符串	r,g,b,alpha 红,绿,蓝,透明度
width_pen	线宽	>=0	为像素宽
style_brush	填充类型	0~14	参考界面的填充选项
color_brush	填充色	字符串	r,g,b,alpha 红,绿,蓝,透明度
color_label	字体颜色	字符串	r,g,b,alpha 红,绿,蓝,透明度
weight_label	字体粗细	0~99	0~99, 99 最粗
size_label	字号	0~N	字的大小
width	符号宽度	0~N	像素宽度
height	符号高度	0~N	像素高度
error	错误信息		如果不存在, 说明成功

6.2.11 props 返回特定标记的标签

范例呼叫:

```
string res = axqtaxviewer_planetosm.osm_layer_call_function
("geomarker1", "function=props;name=LENA;");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	mark	删除标记
name *	查询的标记名		
返回参数表			
键名	键意义	取值	说明
name	点 ID		
error	错误信息		如果不存在, 说明成功
...			
其他为自定义的标签名、值集合			

6.2.12 add_resource 添加资源 (图标)

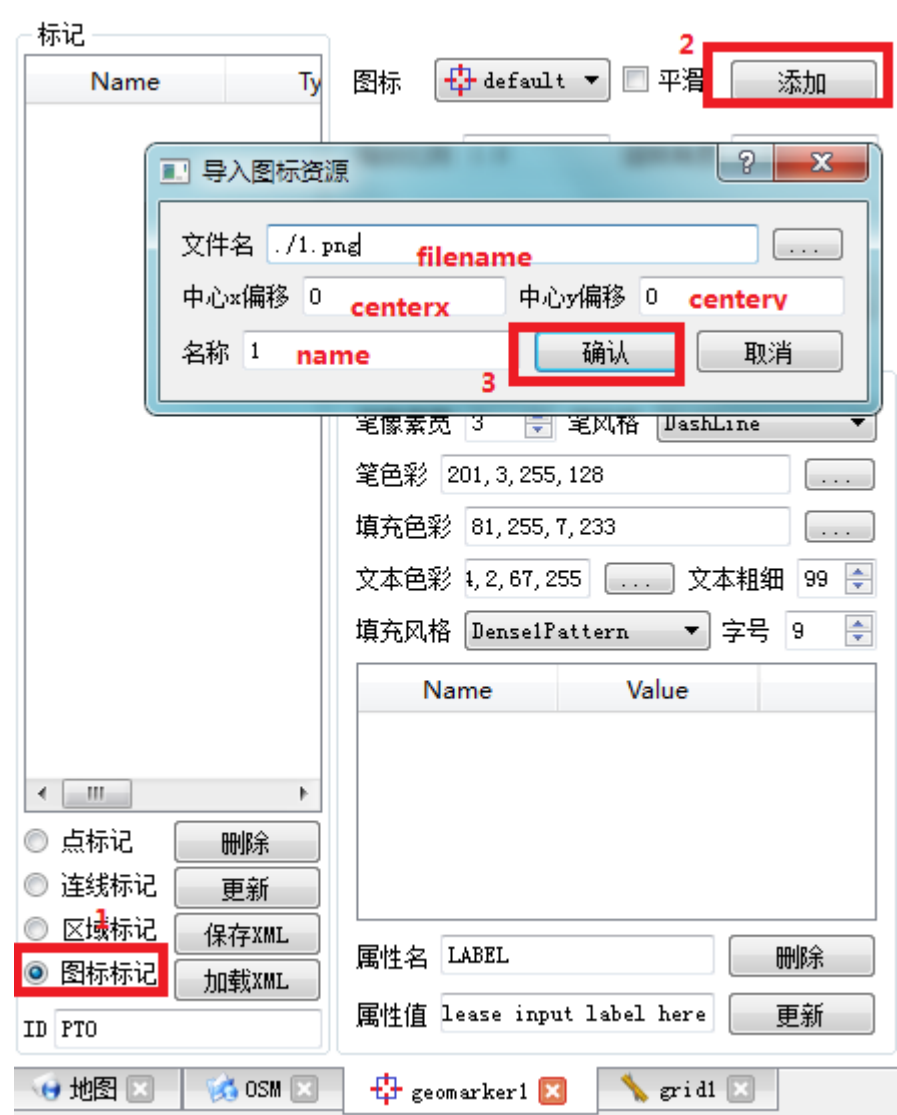
范例呼叫:

```
res = axqtaxviewer_planetosm.osm_layer_call_function("geomarker1",
"function=add_resource;name=lena;" +
"filename=d:/lena_good.png;" +
"centerx=32;centery=32;");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明

function *	函数名称	add_resource	删除标记
name *	图标名	自定义	
filename *	图标文件名	文件位置	注意\\和/的区别
centerx *	图标的中心点偏移	中心点位置	如一个手型图标的中心在手指，一个旗子的中心在旗杆底部
centerx *			
返回参数表			
键名	键意义	取值	说明
error	错误信息		如果不存在，说明成功

对应界面元素



6.2.13 save_resources 保存资源（图标）

```
string ret = axqtaxviewer_planetosm.osm_layer_call_function("geomarker1",  
"function=save_resources;xml=d:/1.xml");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	save_resources	保存图标

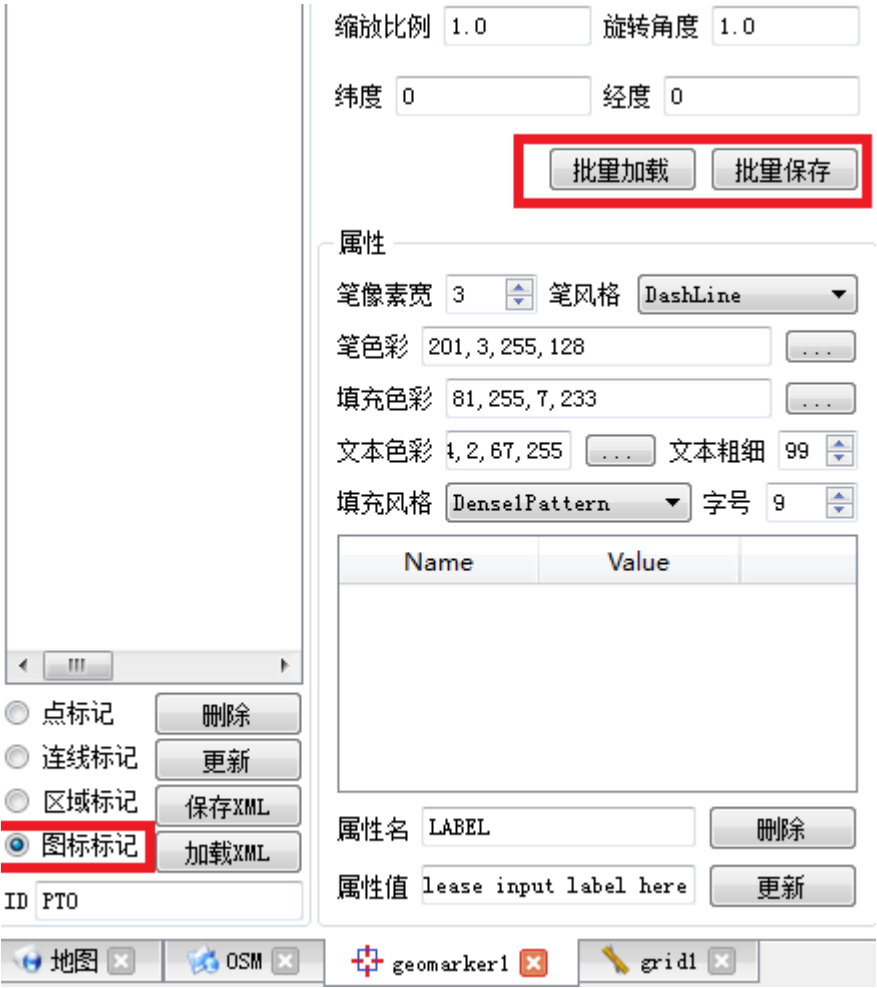
xml *	xml 文件名	文件位置	注意\\和/的区别
返回参数表			
键名	键意义	取值	说明
error	错误信息		如果不存在，说明成功

6.2.14 load_resources 读取资源（图标）

```
string ret = axqtaxviewer_planetosm.osm_layer_call_function("geomarker1",
"function=load_resources;xml=d:/1.xml");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	load_resources	读取图标
xml *	xml 文件名	文件位置	注意\\和/的区别
返回参数表			
键名	键意义	取值	说明
error	错误信息		如果不存在，说明成功

对应界面元素



6.2.15 save_xml 保存标记

```
string ret = axqtaxviewer_planetosm.osm_layer_call_function("geomarker1",  
"function=save_xml;xml=d:/mark.xml");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	save_xml	保存所有
xml *	xml 文件名	文件位置	注意\\和/的区别
返回参数表			
键名	键意义	取值	说明
error	错误信息		如果不存在，说明成功



6.2.16 load_xml 读取标记

```
string ret = axqtaxviewer_planetosm.osm_layer_call_function("geomarker1",  
"function=load_xml;xml=d:/mark.xml");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	load_xml	读取标记
xml *	xml 文件名	文件位置	注意\\和/的区别
返回参数表			
键名	键意义	取值	说明
error	错误信息		如果不存在，说明成功

6.2.17 show_props 展开/隐藏所选图元的详细描述

```
string ret = axqtaxviewer_planetosm.osm_layer_call_function("geomarker1",  
"function=show_props;YunNam=0;Dali=1;Shandong=0;qujing=1;");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	show_props	读取标记
<ID0> *	被展开/隐藏的图元 ID	0,1	0=隐藏， 1=展开
[ID1]...[Idn-1]			
没有列出的图元将被自动关闭展开状态			
返回参数表			
键名	键意义	取值	说明
error	错误信息		如果不存在，说明成功

[ID1]...[Idn-1]	操作后的状态		
-----------------	--------	--	--

6.2.18 props_vis 得到各个图元的详细信息展开状态

```
string ret = axqtaxviewer_planetasm.osm_layer_call_function("geomarker1",
"function=props_vis;name0=YunNam;name1=Shandong;");
```

呼叫参数表(*为必选)			
键名	键意义	取值	说明
function *	函数名称	props_vis	读取标记
空	要询问的图元 ID		
Name0...nameN-1			
没有列出的图元将被自动关闭展开状态			
返回参数表			
键名	键意义	取值	说明
error	错误信息		如果不存在，说明成功
[ID1]...[Idn-1]	状态	0,1	0=隐藏，1=展开

7、 控件事件参考

控件事件会通过 `evt_Message` 发往容器（如 C#程序等），容器即可获知发生在控件中的一切。

事件是以字符串形式封装的“键-值”集合，所有事件都包含三个共有键：

- 1、来源 “source”
- 2、目的 “destin”
- 3、名称 “name”

此外，根据 name 不同，还会跟随若干专有属性名、属性值。

7.1 地图状态事件

7.1.1 WINDOW_CREATE 地图初始化事件

事件体举例：`destin=ALL;name=WINDOW_CREATE;source=MAIN_MAP;`

键名	键意义	取值	说明
source	事件来源	MAIN_MAP	MAIN_MAP 表示主控件
destin	事件目的	ALL	表示发给所有接收者
name	事件名称	WINDOW_CREATE	窗口正在创建

7.1.2 MAP_INITED 地图初始化完成事件

事件体举例：`destin=ALL;nLevel=0;name=MAP_INITED;source=MAIN_MAP;`

键名	键意义	取值	说明
source	事件来源	MAIN_MAP	OSM 表示 OSM 背景图层
destin	事件目的	ALL	表示发给所有接收者
name	事件名称	MAP_INITED	地图初始化完成

7.1.3 CONNECTION 服务连接事件

事件体举例：

`destin=ALL;name=CONNECTION;source=OSM;status=CONNECTED;`

`destin=ALL;name=CONNECTION;source=OSM;status=CLOSED;`

键名	键意义	取值	说明
source	事件来源	OSM	OSM 表示 OSM 背景图层
destin	事件目的	ALL	表示发给所有接收者
name	事件名称	CONNECTION	表示该图层会按需试图从服务器下载瓦片
status	状态	CONNECTED CLOSED	表示“已设置”或者“未设置”

7.1.4 MAP_RESIZED 地图尺寸变化

`destin=OUTER;height=453;name=MAP_RESIZED;source=MAIN_MAP;width=445;`

键名	键意义	取值	说明
source	事件来源	MAIN_MAP	OSM 表示 OSM 背景图层

destin	事件目的	OUTER	表示发给控件外部的容器(C#)
name	事件名称	MAP_RESIZED	地图尺寸变化
width	宽度	像素数	
height	高度	像素数	

7.2 用户漫游事件

7.2.1 MOUSE_XBUTTON_? 鼠标点击事件

事件体举例：

```
destin=OUTER;lat=7.71099165543323;lon=14.0625;nLevel=0;name=MOUSE_LBUTTONDOWN;s
ource=MAIN_MAP;
```

键名	键意义	取值	说明
source	事件来源	MAIN_MAP	MAIN_MAP 表示控件
destin	事件目的	OUTER	表示发给外部控件容器(如 C#)
name	事件名称	MOUSE_LBUTTONDOWN MOUSE_RBUTTONDOWN MOUSE_MBUTTONDOWN MOUSE_BUTTONDOWN MOUSE_LBUTTONUP MOUSE_RBUTTONUP MOUSE_MBUTTONUP MOUSE_BUTTONUP MOUSE_LBUTTONDBLCLK MOUSE_RBUTTONDBLCLK MOUSE_MBUTTONDBLCLK MOUSE_BUTTONDBLCLK	左键按下 右键按下 中键按下 键按下 左键弹起 右键弹起 中键弹起 键弹起 左键双击 右键双击 中键双击 键双击
nLevel	比例尺	当前比例尺	0~18
lat	当前纬度		单位为度
lon	当前经度		单位为度

7.2.2 MOUSE_WHELL 滚轮事件

事件体举例：

```
aDeltaX=0;aDeltaY=120;destin=OUTER;lat=-58.0778762678752;lon=97.03125;nLevel=0;name=M
OUSE_WHELL;pDeltaX=0;pDeltaY=0;source=MAIN_MAP;
```

键名	键意义	取值	说明
source	事件来源	MAIN_MAP	MAIN_MAP 表示控件
destin	事件目的	OUTER	表示发给外部控件容器(如 C#)
name	事件名称	MOUSE_WHELL	滚轮事件
nLevel	比例尺	当前比例尺	0~18
lat	当前纬度		单位为度
lon	当前经度		单位为度

pDeltaX	X 点偏移	1 单位为 pixel	pDelta 是对专业二维轨迹球而言的。对一般的鼠标，为 0
pDeltaY	Y 点偏移	1 单位为 pixel	
aDeltaX	X 角偏移	1 单位为“八分之一度”	aDeltax 是对专业二维轨迹球而言的。对一般的鼠标，为 0
aDeltaY	Y 角偏移	1 单位为“八分之一度”	

7.2.3 MOUSE_MOVE 滚轮事件

事件体举例：

```
destin=OUTER;lat=7.71099165543323;lon=14.0625;mask=0;nLevel=0;name=MOUSE_MOVE;source=MAIN_MAP;
```

键名	键意义	取值	说明
source	事件来源	MAIN_MAP	MAIN_MAP 表示控件
destin	事件目的	OUTER	表示发给外部控件容器(如 C#)
name	事件名称	MOUSE_MOVE	鼠标移动
nLevel	比例尺	当前比例尺	0~18
lat	当前纬度		单位为度
lon	当前经度		单位为度
Mask	按键状态	0x01 = 左键按下 0x02 = 右键按下 0x04 = 中键按下	为掩码，可直接使用&运算符测试

7.2.4 CENTER_CHANGED 中心变更事件

事件体举例：

```
destin=OUTER;lat=31.5785354264734;lon=120.76171875;nLevel=3;name=CENTER_CHANGED;source=MAIN_MAP;
```

键名	键意义	取值	说明
source	事件来源	MAIN_MAP	MAIN_MAP 表示控件
destin	事件目的	OUTER	表示发给外部控件容器(如 C#)
name	事件名称	CENTER_CHANGED	中心改变
nLevel	比例尺	当前比例尺	0~18
lat	当前纬度		单位为度
lon	当前经度		单位为度

7.3 LAYER_MOVED_XXX 图层顺序变更事件

事件体举例：

```
destin=ALL;layerName=geomarker1;name=LAYER_MOVED_DOWN;source=MAIN_MAP;
```

键名	键意义	取值	说明
source	事件来源	MAIN_MAP	MAIN_MAP 表示控件
destin	事件目的	ALL	所有接收者
name	事件名称	LAYER_MOVED_UP	图层已经上移
		LAYER_MOVED_DOWN	图层已经下移

		LAYER_MOVED_TOP LAYER_MOVED_BOTTOM	图层已经置顶 图层已经置底
layerName	图层名称		

7.4 ITEM_XBUTTON_XXXCLICKED geomarker 插件图元鼠标事件

事件体举例：

destin=ALL;id=ID7;name=ITEM_LBUTTON_CLICKED;source=geomarker1;

键名	键意义	取值	说明
source	事件来源	geomarker<n>	Geomark1 表示第一个 geomarker 插件 Geomark2 表示第二个 geomarker 插件
destin	事件目的	ALL	表示发给所有部位
name	事件名称	ITEM_LBUTTON_CLICKED ITEM_RBUTTON_CLICKED ITEM_MBUTTON_CLICKED ITEM_BUTTON_CLICKED ITEM_LBUTTON_DBLCLICKED ITEM_RBUTTON_DBLCLICKED ITEM_MBUTTON_DBLCLICKED ITEM_BUTTON_DBLCLICKED	左键单击 右键单击 中键单击 键单击 左键双击 右键双击 中键双击 键双击
id	图元 ID	为被单击的图元 ID	可以调用相应方法获取该 ID 的详细信息