

# **MMapXControl 说明文档**

作者：嘉博世纪

创建日期：2009年6月22日

更新日期：2009年6月22日

最新版本：3.0.0.0

版权© 2009 为嘉博世纪所有

## 文档管理

### ➤ 修改记录

日期	修改人	修改后版本	修改内容
2009-6-22	嘉博世纪	3.0.0.0	发行第一版,基础API说明

## 一、 控件安装与卸载

### 1) 控件的安装

下载安装程序包MMapXControl.exe（视各服务器[见附录1]不同可能为.exe/.zip/.rar等，.zip/.rar需解压缩后执行安装）。

双击启动安装程序，如下图：



单击“下一步”，进入安装目录选择：



选定安装目录后，单击“安装”，执行安装过程。安装完成后如下图：



单击“完成”结束安装。

## 2) 控件安装的文件

文件名	文件路径	文件大小(字节)	文件版本	文件功能
MMapXControl.ocx	安装路径	345088	3.0.0.0	MMapXControl 控件
Uninstall.exe	安装路径	66700		卸载安装程序
Uninstall.ini	安装路径	1491		卸载安装配置文件
borlndmm.dll	系统路径	18432	5.0.12.34	Borland 公司支持库
cc3250mt.dll	系统路径	983040	5.0.1.80	Borland 公司支持库

## 3) 控件的卸载

可进入windows的“添加或删除程序”管理器进行卸载，也可进入安装路径执行Uninstall.exe进行卸载。

## 二、 控件的API与事件（针对开发者）

MMapXControl.ocx控件在安装完之后，已默认注册到系统ActiveX控件中，名称为“**MMapXControl**”，开发者根据各自使用的开发工具（如Visual C++6.0, Visual Basic, Visual Studio .net, Delphi, C++ Builder等工具）进行加载。

MMapXControl.ocx控件包含丰富的API和Event支持，详细介绍如下（部分代码以Visual C++ 6.0环境为范例）。

目前，MMapXControl.ocx控件仅支持google地图，其余地图供应器（如MapABC、MapBar等）将在后续版本中开发升级。

### 1) API

请使用带“X”开头的API函数系列（“可用”一列为“是”），其余的函数使用将可能得到非您预期的结果。

函数名称	函数定义	可用
GetCenter	BOOL GetCenter(double* Lng, double* Lat);	否
SetCenter	BOOL SetCenter(double Lng, double Lat);	否
GetZoom	BOOL GetZoom(unsigned long* Zoom);	否
SetZoom	BOOL SetZoom(unsigned long Zoom);	否
GetBounds	BOOL GetBounds(double* S, double* W, double* N, double* E);	否
IsInside	BOOL IsInside(double Lng, double Lat, double S, double W, double N, double E);	否
GetEditMode	BOOL GetEditMode();	否
SetEditMode	BOOL SetEditMode(BOOL param0);	否
CreatePoint	BOOL CreatePoint(LPCTSTR Id, double Lng, double Lat, double Crs, LPCTSTR	否

	Icon, LPCTSTR Content, BOOL Visible);	
DestroyPoint	BOOL DestroyPoint(LPCTSTR Id);	否
CreatePolyline	BOOL CreatePolyline(LPCTSTR Id, double LngBuf, double LatBuf, unsigned long BufLen, LPCTSTR Content, unsigned long Color, unsigned long Weight, double Opacity, BOOL Geo);	否
DestroyPolyline	BOOL DestroyPolyline(LPCTSTR Id);	否
CreatePolygon	BOOL CreatePolygon(LPCTSTR Id, BOOL Closed, double LngBuf, double LatBuf, unsigned long BufLen, LPCTSTR Content, unsigned long LineColor, unsigned long LineWeight, double LineOpacity, unsigned long FillColor, double FillOpacity);	否
DestroyPolygon	BOOL DestroyPolygon(LPCTSTR Id);	否
Open	BOOL Open(LPCTSTR Host, LPCTSTR Type);	否
CollectGarbage	BOOL CollectGarbage();	否
WGS84_MARS	void WGS84_MARS(double WGS84_Lng, double WGS84_Lat, double* MARS_Lng, double* MARS_Lat);	否
MARS_WGS84	void MARS_WGS84(double MARS_Lng, double MARS_Lat, double* WGS84_Lng, double* WGS84_Lat);	否
XGetCenter	BOOL XGetCenter(long Lng, long Lat);	是
XSetCenter	BOOL XSetCenter(double Lng, double Lat);	是
XGetZoom	BOOL XGetZoom(long Zoom);	是
XSetZoom	BOOL XSetZoom(unsigned long Zoom);	是
XGetBounds	BOOL XGetBounds(long S, long W, long N, long E);	是
XIsInside	BOOL XIsInside(double Lng, double Lat, double S, double W, double N, double E);	是
XGetEditMode	BOOL XGetEditMode();	是
XSetEditMode	BOOL XSetEditMode(BOOL param0);	是
XCreatePoint	BOOL XCreatePoint(LPCTSTR Id, double Lng, double Lat, double Crs, LPCTSTR Icon, LPCTSTR Content, BOOL Visible);	是

XDestroyPoint	BOOL XDestroyPoint(LPCTSTR Id);	是
XCreatePolyline	BOOL XCreatePolyline(LPCTSTR Id, long LngBuf, long LatBuf, unsigned long BufLen, LPCTSTR Content, unsigned long Color, unsigned long Weight, double Opacity, BOOL Geo);	是
XDestroyPolyline	BOOL XDestroyPolyline(LPCTSTR Id);	是
XCreatePolygon	BOOL XCreatePolygon(LPCTSTR Id, BOOL Closed, long LngBuf, long LatBuf, unsigned long BufLen, LPCTSTR Content, unsigned long LineColor, unsigned long LineWeight, double LineOpacity, unsigned long FillColor, double FillOpacity);	是
XDestroyPolygon	BOOL XDestroyPolygon(LPCTSTR Id);	是
XOpen	BOOL XOpen(LPCTSTR Host, LPCTSTR Type);	是
XCollectGarbage	BOOL XCollectGarbage();	是
XWGS84_MARS	void XWGS84_MARS(double WGS84_Lng, double WGS84_Lat, long MARS_Lng, long MARS_Lat);	是
XMARS_WGS84	void XMARS_WGS84(double MARS_Lng, double MARS_Lat, long WGS84_Lng, long WGS84_Lat);	是（无效果）
DrawTextBiDiModeFlagsReadingOnly	long DrawTextBiDiModeFlagsReadingOnly();	否
InitiateAction	void InitiateAction();	否
IsRightToLeft	BOOL IsRightToLeft();	否
UseRightToLeftReading	BOOL UseRightToLeftReading();	否
UseRightToLeftScrollBar	BOOL UseRightToLeftScrollBar();	否
GetCursor	short GetCursor();	否
SetCursor	void SetCursor(short nNewValue);	否
AboutBox	void AboutBox();	是

■ 地图中心点坐标Get/Set

➤ BOOL XGetCenter(long Lng, long Lat);

功能：获取当前地图中心点坐标

输入参数：Lng为经度double变量的地址值，Lat为纬度double变量的地址值

返回参数：返回获取过程是否成功

示例代码：

```
double lng, lat;
```

```

if(map.XGetCenter((long)&lng, (long)&lat))
{
    ///获取成功, lng/lat中存放的数值为当前地图的中心点坐标
}
else
{
    ///获取失败, lng/lat中存放的数值未知
}

```

- `BOOL XSetCenter(double lng, double lat);`

功能: 设置地图中心点坐标

输入参数: lng为经度, lat为纬度

返回参数: 返回设置过程是否成功

示例代码:

```
bool value = map.XSetCenter(120.15608, 30.297658);
```

## ■ 地图缩放级别Get/Set

- `BOOL XGetZoom(long zoom);`

功能: 获取当前地图缩放级别

输入参数: zoom为缩放级别unsigned int变量的地址值

返回参数: 返回获取过程是否成功

示例代码:

```

unsigned int zoom;
if(map.XGetZoom((long)&zoom))
{
    ///获取成功, zoom中存放的数值为当前地图的缩放比例
}
else
{
    ///获取失败, zoom中存放的数值未知
}

```

- `BOOL XSetZoom(unsigned long zoom);`

功能: 设置地图缩放比例

输入参数: zoom为缩放比例

返回参数: 返回设置过程是否成功

示例代码:

```
bool value = map.XSetZoom(15);
```

- 关于缩放级别

Google地图目前支持的缩放级别为0~19, 控件支持与Google地图一致(详见

<http://code.google.com/intl/zh-CN/apis/maps/documentation/introduction.html#MapTypes>)。

其他地图供应器将随新发布的供应器对缩放级别定义进行更新。

## ■ 当前地图视野坐标Get

- `BOOL XGetBounds(long S, long W, long N, long E);`  
功能：获取当前地图视野坐标（南、西、北、东）  
输入参数：S为南端double类型变量的地址值，W为西端double类型变量的地址值，N为北端double类型变量的地址值，E为东端double类型变量的地址值

返回参数：返回获取过程是否成功

示例代码：

```
double S, W, N, E;  
if(map.XGetBounds((long)&S, (long)&W, (long)&N, (long)&E))  
{  
    ///获取成功，S/W/N/E中存放的数值为当前地图视野坐标  
}  
else  
{  
    ///获取失败，S/W/N/E中存放的数值未知  
}
```

#### ■ 坐标是否在某矩形内

- `BOOL XIsInside(double Lng, double Lat, double S, double W, double N, double E);`

功能：判断点（Lng/Lat）是否在矩形（S/W/N/E）内

输入参数：Lng/Lat为点坐标、S/W/N/E为矩形坐标

返回参数：判断的结果

示例代码：

```
bool value = map.XIsInside(120.15608, 30.297658, 10, 75, 50,  
130));
```

特别，当S/N值的绝对值大于90或W/E的绝对值大于180时，被判断矩形区域为当前地图视野区域。

```
bool value = map.XIsInside(120.15608, 30.297658, 360, 360, 360,  
360));
```

#### ■ 地图编辑模式Get/Set

- `BOOL XGetEditMode();`

功能：获取当前地图编辑模式

输入参数：

返回参数：是否为编辑模式

示例代码：

```
bool value = map.XGetEditMode();
```

- `BOOL XSetEditMode(BOOL param0);`

功能：设置地图编辑模式

输入参数：true为编辑模式，false为非编辑模式

返回参数：返回设置过程是否成功

示例代码：



```

if(map.XSetEditMode(!map.XGetEditMode()))
{
    ///设置成功
}
else
{
    ///设置失败
}

```

## ■ 点的创建与删除

➤ `BOOL XCreatePoint(LPCTSTR Id, double Lng, double Lat, double Crs, LPCTSTR Icon, LPCTSTR Content, BOOL Visible);`

功能：创建点

输入参数：Id为点的字符串标识，Lng/Lat为点的坐标，Crs为点的正北转角，Icon为点的图标（本地路径或网络URL，空字符为控件默认图片），Content为点的描述（显示在图标旁），Visible为点的可视属性

返回参数：创建是否成功

示例代码：

```

double lng = 120.15608;
double lat = 30.297658;
if(map.XCreatePoint("1", lng, lat, 45,
"..../images/markers/car1.png", "Point1", true))
{
    ///创建成功
}

```

➤ `BOOL XDestroyPoint(LPCTSTR Id);`

功能：删除点

输入参数：Id为点的字符串标识

返回参数：删除是否成功

示例代码：

```

if(map.XDestroyPoint("1"))
{
    ///删除成功
}

```

## ■ 线的创建与删除

➤ `BOOL XCreatePolyline(LPCTSTR Id, long LngBuf, long LatBuf, unsigned long BufLen, LPCTSTR Content, unsigned long Color, unsigned long Weight, double Opacity, BOOL Geo);`

功能：创建线

输入参数：Id为点的字符串标识，LngBuf/LatBuf为经纬度double串的首地址，BufLen为经纬度double串的长度，Content为线的描述（当鼠标移动到线上时显示），Color为线的颜色（0~65535），Weight为线的宽度（>1），Opacity为线的透明度（0~1），Geo为测地线属性（详见

<http://code.google.com/intl/zh-CN/apis/maps/documentation/reference.html#GPolylineOptions>)

返回参数：创建是否成功

示例代码：

```
double lng[2] = {130, 75};
double lat[2] = {50, 10};
if(map.XCreatePolyline("1", (long)&lng, (long)&lat, 2,
"Polyline1", 100, 4, 0.5, true))
{
    ///创建成功
}
```

➤ `BOOL XDestroyPolyline(LPCTSTR Id);`

功能：删除线

输入参数：Id为点的字符串标识

返回参数：删除是否成功

示例代码：

```
if(map.XDestroyPolyline("1"))
{
    ///删除成功
}
```

#### ■ 面的创建与删除

➤ `BOOL XCreatePolygon(LPCTSTR Id, BOOL Closed, long LngBuf, long LatBuf, unsigned long BufLen, LPCTSTR Content, unsigned long LineColor, unsigned long LineWeight, double LineOpacity, unsigned long FillColor, double FillOpacity);`

功能：创建面

输入参数：Id为点的字符串标识，Closed为面的封闭属性，

LngBuf/LatBuf为经纬度double串的首地址，BufLen为经纬度double串的长度，Content为面的描述（当鼠标移动到面上时显示），LineColor为面的外边界线颜色(0~65535)，LineWeight为面的外边界线宽度(>1)，LineOpacity为面的外边界线透明度(0~1)，FillColor为面的内部填充色(0~65535)，FillOpacity为面的内部填充透明度(0~1)

返回参数：创建是否成功

示例代码：

```
double Lng[4] = {130, 130, 75, 75};
double Lat[4] = {50, 10, 10, 50};
if(map.XCreatePolygon("1", true, (long)&Lng, (long)&Lat, 4,
"Polygon1", 200, 1, 1, 400, 0.3))
{
    ///创建成功
}
```

➤ `BOOL XDestroyPolygon(LPCTSTR Id);`

功能：删除面

输入参数：Id为点的字符串标识

返回参数：删除是否成功

示例代码：

```
if(map.XDestroyPolygon("1"))
{
    ///删除成功
}
```

## ■ 启动地图

➤ `BOOL XOpen(LPCTSTR Host, LPCTSTR Type);`

功能：启动地图

输入参数：Host为控件服务器地址字符串(IP或域名)，Type为启动的地图供应器类型（目前仅有“Google”可用）

返回参数：启动是否成功

示例代码：

```
if(map.XOpen("60.191.22.196", "Google"))
{
    ///启动成功
}
```

## ■ 内存回收

➤ `BOOL XCollectGarbage();`

功能：控件内存回收

输入参数：

返回参数：内存回收是否成功

示例代码：

```
if(map.XCollectGarbage())
{
    ///回收成功
}
```

➤ 请在地图操作不频繁的时候执行该函数，可减少进程的内存占用；**该函数的使用不当可能导致进程崩溃，请开发者慎用**（如果使用请多做测试）。

## ■ WGS84坐标系与Mars坐标系相互转换

➤ 火星坐标系概念见[附录2]

➤ `void XWGS84_MARS(double WGS84_Lng, double WGS84_Lat, long MARS_Lng, long MARS_Lat);`

功能：WGS84坐标系转Mars坐标系

输入参数：WGS84\_Lng为WGS84坐标经度，WGS84\_Lat为WGS84坐标纬度，MARS\_Lng为Mars坐标经度double类型变量的地址值，MARS\_Lat为Mars坐标纬度double类型变量的地址值

返回参数：

示例代码：

```
double lng = 120.15608;
```

```
double lat = 30.297658;
double vlng, vlat;
map.XWGS84_MARS(lng, lat, (long)&vlng, (long)&vlat);
```

- void XMARS\_WGS84(double MARS\_Lng, double MARS\_Lat, long WGS84\_Lng, long WGS84\_Lat);  
功能: Mars坐标系转WGS84坐标系(无效果)  
输入参数: MARS\_Lng为Mars坐标经度, MARS\_Lat为Mars坐标纬度, WGS84\_Lng为WGS84坐标经度double类型变量的地址值, WGS84\_Lat为WGS84坐标纬度double类型变量的地址值  
返回参数:  
示例代码:  
暂时不给出

## 2) Event

### ■ OnReady

- 功能: 当调用XOpen函数成功后, 地图加载完成时, 触发该事件
- 回调函数类型: void OnReady(long Buf, long Len)
- 参数: Buf始终为0, Len始终为0
- 示例代码:

```
void CDemoForVCDlg::OnOnReadyMmapx1(long Buf, long Len)
{
    ///地图加载完成时, 执行地图初始化操作
    map.XSetCenter(120.15608, 30.297658);
    map.XSetZoom(15);
    map.XSetEditMode(true);
}
```

### ■ OnMapType

- 功能: 当使用者切换地图类型时触发(主要针对google地图可切换地图类型而设计)

- 回调函数类型: void OnMapType(long Buf, long Len)
- 参数: Buf为字符串首地址, Len为字符串长度
- 示例代码:

```
void CDemoForVCDlg::OnOnMapTypeMmapx1(long Buf, long Len)
{
    char* p = new char[Len+1];
    memcpy(p, (char*)Buf, Len);
    p[Len] = '\0';
    MessageBox(p);
    delete p;
}
```

### ■ OnEditor

功能：当使用者在编辑模式下完成点/线/面的编辑之后，点击保存按钮时发生

回调函数类型：void OnEditor(long Buf, long Len)

参数：Buf为字符串首地址，Len为字符串长度

示例代码：

```
void CDemoForVCDlg::OnOnEditorMmapx1(long Buf, long Len)
{
    char* p = new char[Len+1];
    memcpy(p, (char*)Buf, Len);
    p[Len] = '\0';
    MessageBox(p);
    delete p;
}
```

Buf串的编码规则如下：

“类型;经纬度组数;(经纬度);...;(经纬度);名称”

类型：1为点，2为线，3为面

#### ■ OnSelObj

功能：当使用者在地图上单击由

XCreatePoint/XCreatePolyline/XCreatePolygon函数创建的可视对象时触发的事件

回调函数类型：void OnSelObj(long Buf, long Len)

参数：Buf为字符串首地址，Len为字符串长度

示例代码：

```
void CDemoForVCDlg::OnOnSelObjMmapx1(long Buf, long Len)
{
    char* p = new char[Len+1];
    memcpy(p, (char*)Buf, Len);
    p[Len] = '\0';
    MessageBox(p);
    delete p;
}
```

Buf串的编码规则如下：

“类型;Id”

类型：1为点，2为线，3为面

Id：该对象创建时，传入

XCreatePoint/XCreatePolyline/XCreatePolygon函数的Id值

### 三、 控件的使用与操作规范（针对使用者）

#### 1) 功能与特点

安装简便；支持远程更新（如控件升级，需要使用者重新安装控件包；如功能升级，则只需要服务器发布后，控件自动会下载更新）；画点支持角度参数，可体现实地物的旋转表示；画点支持缩放级别同步，在不同的缩放级别下都有良好的视觉效果；支持二次开发的地物对象的鼠标点选操作，

加强了控件的交互性；提供完善的地图编辑器功能；提供WGS84坐标到Mars坐标的转换。

## 2) 常用操作

### 1. 编辑器线/面编辑时局部点展开/删除



展开前



展开后

## 四、 FAQ

### a) 地图控件是否能在局域网中使用？

答：只要局域网支持HTTP代理，或者，只要您的IE浏览器可以访问一下网页 <http://ditu.google.cn/>，就可以使用本地图控件。

## 五、 附录

1. 目前有效服务器URL

服务器IP	下载服务页	XOpen中的Host
60.191.22.196	60.191.22.196/MTL/Index.html	60.191.22.196

2. 关于火星坐标系的相关信息

<http://www.unistrong.com/bbs/topic.aspx?topicid=2774>