



TarMap 二次开发手册

(ver1.0)

广东智维科技有限公司

2005年12月

目 录

第一章 前言	3
第二章 安装	4
1、 运行环境	4
2、 安装	4
2.1 安装组件	4
2.2 安装地图	4
第三章 功能简介	5
1、 GIS功能	5
1.1 显示地图	5
1.2 地图缩放	5
1.3 地图平移	6
1.4 测量距离	6
1.5 比例尺	6
1.6 动态注记	6
1.7 平滑浏览	7
1.8 地物查询	7
1.9 注记自动避让	9
1.10 地图自动切换	9
2、 GPS功能	9
2.1 显示目标	9
2.2 锁定跟踪	10
2.3 轨迹管理	10
2.4 目标选择	10
2.5 多窗口多区域监控	11
第四章 开发指南	11
1、 COM技术概述	11
2、 TARMAP体系结构	11
3、 如何开始	12
3.1 注册组件	12
3.2 引入组件	12
3.3 创建对象	13
4、 初始化	14
第五章 用户参考手册	16
1、 ITARMANAGER对象	16
1.1 属性	16
1.2 方法	16
2、 ITARMAP对象(可视控件)	17
2.1 属性	17
2.2 方法	18
2.3 事件	21
3、 ITARSELECTIONS对象	23
3.1 属性	23
3.2 方法	23

本手册主要介绍基于 TarMap 二次的开发方法。

预期阅读者：GIS、GPS 监控系统开发人员。

限制：不得擅自修改本文档的内容

版本：V1.0

撰写人：肖钢 13112215331 MSN: hb Xiaogang@hotmail.com

第一章 前言

地理信息系统（GIS）是一种以地理信息为主要内容的综合性管理系统，是地理学、地图学与计算机技术相结合的产物。GIS应用于城市管理，是以电子地图的形式直观地表现城市基础地理信息，并可作图文互查，综合分析等。

位置服务系统（LBS）是利用先进的计算机技术、GIS和全球定位系统（GPS）来实现对动态目标（车辆、船舶、飞机）进行实时监控和调度的综合信息管理系统。

目前国内大部分和位置服务有关的系统（如：GPS监控系统）的开发是在MAPINFO，ARC/INFO及一些国内GIS平台上进行的二次开发，但这些GIS开发平台往往侧重于GIS管理、分析功能，而忽略了GPS监控系统所要求的实时性与准确性。GIS的很多强大的分析功能在GPS监控系统中根本用不上，而对于大多数的GPS监控系统用户及系统集成商来说，也并不想了解过多的GIS算法、概念。

我们在长期从事GPS监控系统项目的开发、实施过程中，总结了大量的经验，推出此专门针对GPS应用的二次开发平台。通过广泛收集用户需求，将多年的行业经验融于该产品。您在开发过程中不用再面对复杂的GIS概念，地图的显示、编辑、查询全部封装在控件内部，您不需要为如何将监控目标正确地显示在地图上而绞尽脑汁。

该产品的设计以高效、简洁、实用为基本原则，二次开发过程简单方便。该组件产品提供3个接口，共3个属性、16个方法、8个事件，因此，对于二次开发而言十分简单，充分展现了TarMap的高度封装和优秀的图形技术。

该产品基于 Windows COM (Component Object Model) 技术设计，通过 COM 技术实现了整个 GIS (Geographic Information System) 的底层，并在该底层的基础上完成了 GPS 监控系统开发控件。提供用于 GIS 管理与 GPS 监控功能的 API，您可以使用 VB、VC++、Delphi 等流行开发工具，轻松地将空间信息管理和 GPS 监控融入到应用系统中。

另外，我们将向二次开发商免费提供用 VC++ 开发出的完整 GPS 监控系统的源代码和开发指导。

第二章 安装

1、运行环境

本系统的运行环境如下表所示：

	硬件	一般配置	推荐配置
硬件环境	CPU	Pentium MMX 以上	Pentium 1G 以上
	内存	128M	256M 以上
	硬盘	50M	1G 以上
软件环境	简体中文 Windows 桌面操作系统(98/Me/2000/XP/Server)		
二次开发语言平台	VC++/VB/Delphi 等		

2、安装

安装前确保您的电脑系统符合上述条件。

2.1 安装组件

- 1、将 TarMapCtrl.dll 文件放入 system32 目录。
- 2、在 dos 命令行提示符下或“开始”=>“运行”窗口中运行指令：
Regsvr32 TarMapCtrl.dll 即可。

2.2 安装地图

- 1、将 TarMap 格式的地图文件(*.map)地图文件放入任何一个目录中（如果有多个地图文件，则必须放入同一目录）；
- 2、建立一个地图库文件（index.tap），文本格式。格式如下：

[START]

全中国.map //第一幅地图文件名

广东省.map //第二幅地图文件名
深圳市.map //第三幅地图文件名
.....
[END]

第三章 功能简介

1、GIS 功能

1.1 显示地图

显示 TarMap 格式的地图，整体效果美观、简洁。



1.2 地图缩放

- 1、通过鼠标画矩形或鼠标滚轮，对地图进行快速放大或缩小(不需额外编程)；
- 2、用 PgUp 或 PgDown 对地图显示比例进行微调(不需额外编程)。

1.3 地图平移

通过鼠标或键盘方向键对地图进行全方位平移(不需额外编程)。

1.4 测量距离

根据鼠标所画的折线，计算距离。



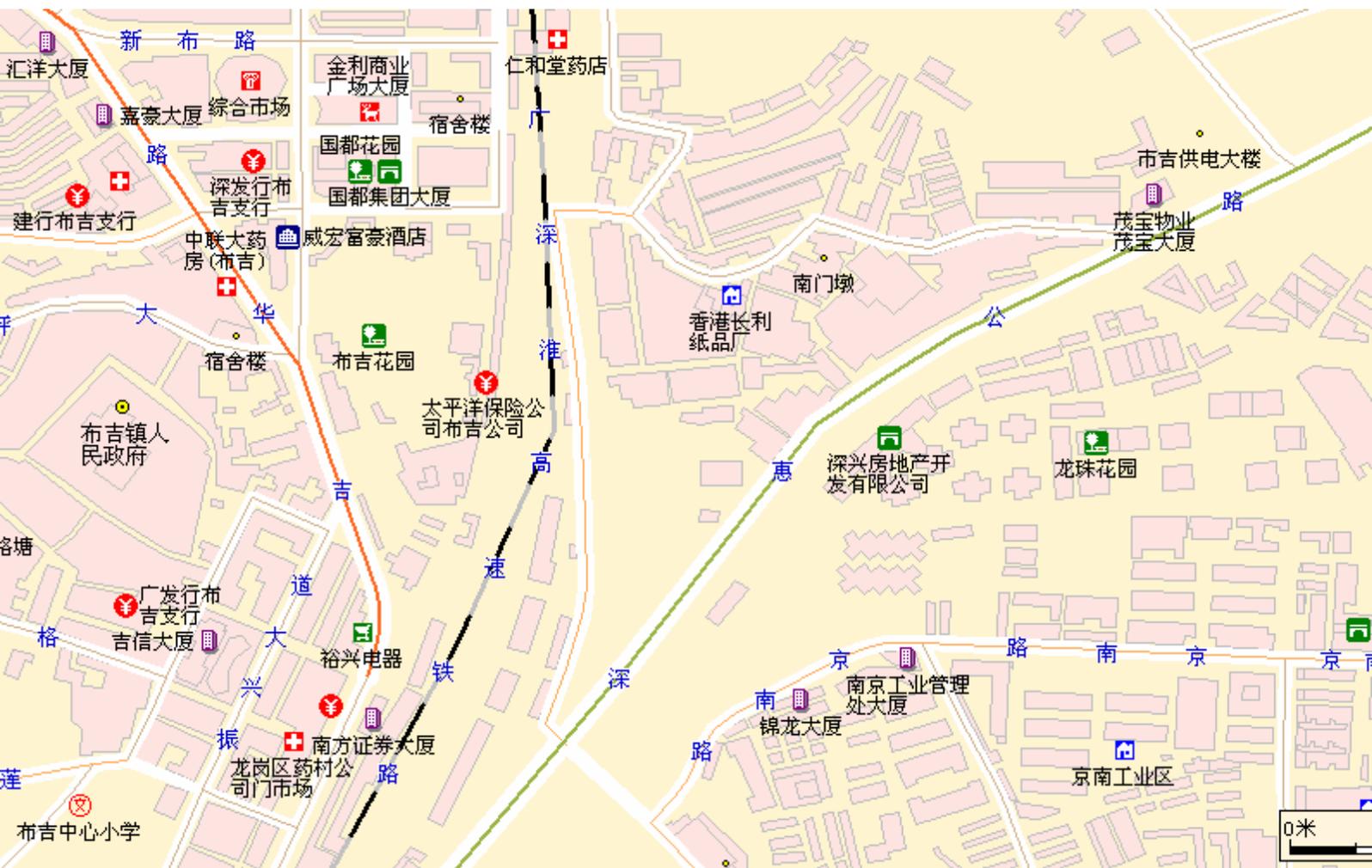
1.5 比例尺

在监控窗口右下角显示当前地图的当前比例尺。



1.6 动态注记

实现道路、水系注记流动显示，即窗口中的道路名称始终显示全名，不会出现道路名称不全的情况。



1.7 平滑浏览

将地图操作设为平移模式，结合鼠标滚轮的缩放操作，可实现快速平滑浏览地图。不会出现“反白”现象。

1.8 地物查询

- 1、点选查询
- 2、查询矩形范围内的所有地物；
- 3、按关键字模糊查询。





1.9 注记自动避让

在图上点位很密集的情况下，不会出现点位互相压盖，注记重叠一起，系统会自动地调整点位图标的注记的显示位置和尺寸，保证地图显示的美观和图面的整洁。

1.10 地图自动切换

当有多幅地图时，系统会自动根据锁定目标所属地理坐标范围切换到相应的地图。

2、GPS 功能

2.1 显示目标

- 1、在地图窗口中显示同时显示 N 个监控目标(N 值取决于电脑硬件配置)。
- 2、根据目标的当前状态，用不同的颜色显示目标符号，以区分目标的正常和异常。

2.2 锁定跟踪

可以对目标设定锁定跟踪模式，此时，目标外围有红色矩形框提示为锁定状态。处于锁定跟踪模式的目标始终在当前窗口范围内。



2.3 轨迹管理

- 1、设定轨迹是否显示；
- 2、不同目标轨迹点的颜色不同，以方便同时查看多个目标的轨迹。

2.4 目标选择

可以直接在地图上用鼠标选择监控目标，查看该目标的属性(是否显示轨迹、是否被锁定等)，同时通知应用程序当所选择的目标 ID,以方便实现该目标状态、其它属性的同步显示。

2.5 多窗口多区域监控

可以在多个窗口中显示不同的地图。

第四章 开发指南

1、COM 技术概述

组件对象模型（Component Object Model, COM）是创建 Windows 应用程序时所选用的进程间通信标准，即软件组件之间的通信标准。组件实际上是一些小的二进制可执行程序，它可以给应用程序、操作系统以及其他组件提供一些服务。开发定制的 COM 组件就如同开发动态的、面向对象的 API。多个 COM 对象可以连接起来以形成应用程序或组件系统。并且组件可以在运行时、在不重新连接或编译应用程序的情况下被卸下或替换掉。Microsoft 的许多技术，如 ActiveX、DirectX 以及 OLE 等都是基于 COM 而建立起来的。并且 Microsoft 的开发人员也大量的使用 COM 组件来定制它们的应用程序及操作系统。

尽管 COM 是 Microsoft 推广的 Windows 操作系统下编程技术，但 COM 技术所蕴含的概念是不会受到操作系统或开发平台的限制的。与结构化编程和面向对象编程相似，COM 实际上是提供了一种平台无关的编程方法，任何编程人员都可能从这种编程方法中受益，尽管 Windows 程序员运用这种编程方法可能会快捷方便许多。

COM 技术具有以下几个特点：

- 1、与开发平台无关
- 2、安全的对象封装与接口设计
- 3、支持自动化对象接口
- 4、动态连接
- 5、优秀的版本控制

2、TarMap 体系结构

TarMap 组件平台目前提供 4 个接口，如下表所示：

TarMap 组件平台				
接口名称	ITarManager	ITarMap	ITarSelection	ITarSelections
是否可视	否	是	否	否
功能简介	管理地图数据和各接口间的通信	监控窗口. GIS 底层接口, 显示地图	保留	地物选择集合

3、如何开始

3.1 注册组件

平台提供的组件文件名为 `TramapCtrl.dll`，使用前必须注册。系统安装时自动注册该组件库，另外也可以用 `Regsrv32.exe` 或 `Tstcon32.exe` 注册。或者也可以在开发平台的集成环境中注册。

3.2 引入组件

注册组件后，可以将其加入开发工具中（如 VB）。下面以当前几种流行的开发工具（Visual C++、Visual Basic、Delphi）为例进行详细介绍。

1、Visual C++ 6.0

新建一个工程，在 `StdAfx.h` 中加入下面代码：

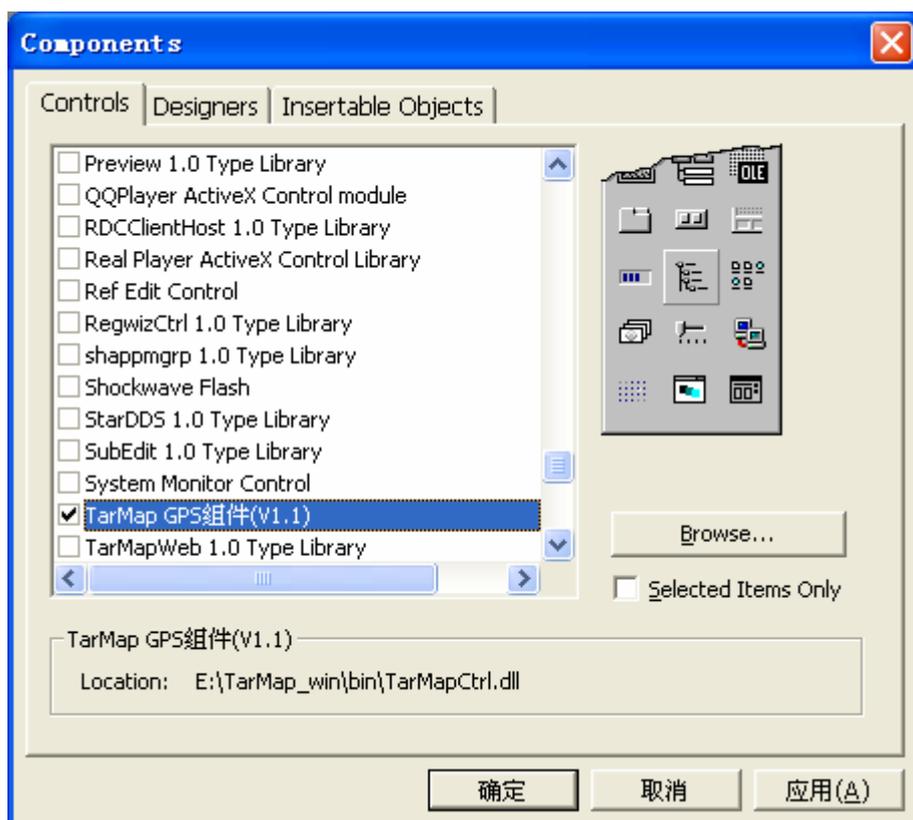
```
#import "..\bin\TarMapCtrl.dll" no_namespace
```

注意：

上面代码中引号内的部分是组件的注册相对路径，一般位于 Windows 系统目录。如果是手工通过 `Regsrv32.exe` 注册，请不要将 `TarMapCtrl.dll` 文件放入含中文的路径中。

2、Visual Basic 6.0

新建一个工程，选择 `Project` 菜单下的 `Components` 命令，在组件对话框的组件列表选中“`TarMap GPS 组件 (V1.1)`”，如下图所示：



3、Delphi

选择 Component 菜单的” Import ActiveX Control” 命令, 在 Import ActiveX 对话框的组件列表中选择 “TarMap GPS 组件 (V1.1)”, 然后单击 Install 按钮, 输入要安装的组件包的文件名及描述, 单击确定按钮。

3.3 创建对象

控件是可视化对象, 可以在设计时创建, 也可以在运行时创建。另外系统还提供了一些非可视对象, 它们必须在运行时创建。下面以 VC 为例介绍:

1. 创建 ITarManager 对象:

```

CLSID clsid;
HRESULT hr;
ITarManager* m_pTarManager;//申明ITarManager对象
hr=CLSIDFromProgID(OLESTR("TarMapCtrl.TarManager"),&clsid);
if(FAILED(hr))
{
    ::MessageBox(0,_T("没有检测到组件
    TarMapCtrl.TarManager, 请确认该组件是否注册安装! "),

```

```

        _T("错误"),
        MB_ICONHAND|MB_OK|MB_DEFBUTTON1);
    return;
}

hr=CoCreateInstance(clsid,NULL,CLSCTX_INPROC_SERVER,
    __uuidof(ITarManager),(LPVOID *)&m_pTarManager);

if(FAILED(hr))
{
    AfxMessageBox("TarManager 组建创建失败");
    return;
}

```

2. 创建 ITarMap 对象：

ITarMap 是可视对象，用于显示地图。对象一般在 CView 中创建，在 CView 类中声明一个 CWnd 对象，如：CWnd m_wndTarMap 对象，再声明 ITarMap* m_pTarMapCtrl。

创建方法：

```

m_wndTarMap.CreateControl(__uuidof(TarMap),
    NULL,
    WS_VISIBLE | WS_CHILD,
    ctrlrect,
    this,
    IDC_TARMAP));

LPUNKNOWN lpknow=m_wndTarMap.GetControlUnknown();
if(!lpknow)
    return FALSE;
LPDISPATCH lpDisp=NULL;
HRESULT hr;
hr=lpknow->QueryInterface(IID_IDispatch,(void**>(&lpDisp));
if(FAILED(hr)||NULL==lpDisp)
    return FALSE;
m_pTarMapCtrl= (ITarMap*)lpDisp;

```

4、初始化

第一步，创建 TarMap 管理控件：ITarManager，该控件为不可见

第二步，创建 TarMap 监控窗口控件：ITarMap。

第三步，调用 ITarMap.Init

第四步，打开图库，调用 TarManager.LoadMaps，将带路径的图库文件名作为参数传给该函数，即可载入图库中所定义的地图，图库文件为一文本文件，

文件格式如下：

```
[START]
全中国.map           //第一幅地图文件名
广东省.map          //第二幅地图文件名
深圳市.map          //第三幅地图文件名
.....
[END]
```

第五章 用户参考手册

1、ITarManager 对象

1.1 属性

无。

1.2 方法

1、LoadMaps

用途：装载地图图库。

定义：HRESULT LoadMaps (_bstr_t strMapPath)

输入参数：_bstr_t strMapPath，图库文件路径和文件名。

输出参数：无。

返回：HRESULT

VC 示例：

```
hr = m_pTarManager->LoadMaps("D:\\MyMap\\tap\\index.tap");  
if(FAILED(hr))  
{  
    AfxMessageBox("LoadMaps()失败");  
    return false;  
}
```

VB 示例：

```
Dim GpsManager As New TarManager '全局变量  
GpsManager.LoadMaps("D:\\MyMap\\tap\\index.tap");
```

2、GetGeoMapsName

用途：获取已载入的地图名称列表。

定义：string GetGeoMapsName(long* nCount)

输入参数：nCount：地图个数

输出参数: nCount: 地图个数。

返回: 由地图名称组成的字符串。地图名称间用空隔为间隔符。一般在接收到 MapLoadedComplete 事件后才调用该函数。

VC 示例:

```
_bstr_t bstrNames("");  
long nCount=0;  
bstrNames= m_pTarManager->GetGeoMapsName(&nCount);
```

VB 示例:

```
sMapNames As String  
Dim nMapCount As Long  
sMapNames = GpsManager.GetGeoMapsName(nMapCount)
```

2、ITarMap 对象(可视控件)

2.1 属性

1、TarManager

该属性值应为事先创好的 TarManager 对象。

2、ActionType

地图操作类型, 类型值如下:

ACTIONTYPE_NONE: 无操作

ACTIONTYPE_ZOOMIN: 放大

ACTIONTYPE_ZOOMOUT : 缩小

ACTIONTYPE_PAN: 平移

ACTIONTYPE_MAPPOINTSELECT: 点选地物

ACTIONTYPE_RECT : 矩形选择地物

ACTIONTYPE_MEASURELENGTH : 测量距离

3、TarSelections

地物集对象。

2.2 方法

1、Init

用途：初始化。在调用 ITarMap 对象的其他方法前调用该方法。

定义：HRESULT Init()

输入参数：无。

输出参数：无。

返回：HRESULT

VC 示例：

```
HRESULT hr =pTarMap->Init();
if(FAILED(hr))
{
    AfxMessageBox("Init ()失败");
    .....
}
```

VB 示例：

```
TarMap1.Init
```

2、ChangeMapTo

用途：切换地图

定义：HRESULT ChangeMapTo([in] int nIndex)

输入参数：nIndex，表示地图序号，从 0 开始，即第一幅地图的序号为 0

输出参数：无。

返回：HRESULT

VC 示例：

```
int n=0;
HRESULT hr =pTarMap-> ChangeMapTo (n);
if(FAILED(hr))
```

```
{  
    AfxMessageBox("ChangeMapTo ()失败");  
    .....  
}
```

VB 示例:

```
TarMap1.ChangeMapTo 2'切换到第 3 幅地图
```

3、SetActive

用途：激活地图窗口，使之重画，当地图窗口失去焦点后，可调用此方法

定义：HRESULT SetActive()

输入参数：无。

输出参数：无。

返回：HRESULT

4、ZoomFull

用途：显示全图。使当前地图显示于当前窗口中

定义：HRESULT ZoomFull()

输入参数：无。

输出参数：无。

返回：HRESULT

5、QuerObsure

用途：模糊查询地物

定义：HRESULT QuerObsure([in]BSTR strKeyword,[in]ITarSelections* pSelections)

输入参数：strKeyword：关键字，pSelections：查询结果集。

输出参数：无。

返回：HRESULT

VC 示例:

```
HRESULT hr;
```

```

ITarSelectionsPtr pMapSelections;
hr = pMapSelections.CreateInstance(__uuidof(TarSelections));
if(FAILED(hr))
{
    AfxMessageBox("Create TarSelections Instace Fail");
    return false;
}
pTarMap->QuerObsure("学校", pMapSelections);

```

VB 示例:

```

Dim pTarSelections As New TarSelections
TarMap1.QuerObsure "市", pTarSelections '查询当前地图中所有名称中含“市”的地物,
'并存储到 pTarSelections 对象中

```

6、SelectGeoObject

用途: 选中 TarSelections 集中的地物

定义: HRESULT SelectGeoObject([in] int nIndex)

输入参数: nIndex: TarSelections 集中地物的序号, 从 0 开始。

输出参数: 无。

返回: HRESULT

7、ShowTarget

用途: 显示或更新监控目标的位置

定义: int ShowTarget([in]BSTR strTragetName,[in]int x,[in]int y, [in]float fDir,[in]BOOL bAlarming ,[in]BOOL bPutInView)

输入参数: strTragetName: 目标名称

x: 经度, 单位: 分

y: 纬度, 单位: 分

fDir: 方向

bAlarming: 是否报警

bPutInView: 保留

输出参数: 无。

返回: 0: 成功, -1: 失败

8、SelectTrarget

用途：锁定监控目标

定义：HRESULT SelectTrarget([in]BSTR bstrTargetName,[out,retval]int *nRet)

输入参数：bstrTargetName-目标名称

输出参数：nRet: 0-成功，-1-失败

返回：nRet

VC 示例：

```
SelectTrarget("粤 A-0001");
```

VB 示例：

```
Dim nRet As Integer  
nRet = TarMap1.SelectTrarget("粤 A-0001")  
If nRet = -1 Then  
    MsgBox "粤 A-0001 不存在，无法选中该目标"  
End If
```

9、DeleteTarget

用途：将监控目标从地图上删除

定义：DeleteTarget([in]BSTR bstrTargetName,[out,retval]int* nRet)

输入参数：bstrTargetName-目标名称

输出参数：nRet: 0-成功，-1-失败

返回：nRet

2.3 事件

1、MapLoadedComplete

当调用 ITarManager.LoadMaps()后，该方法会立即返回，TarMap 将在后台以异步方式装载地图。当所有地图数据都装载完成后，TarMap 组件会发出该事件，已通知应用程序地图已经装载完毕。一般应用程序在收到该事件后，开始调用 ITarManager.GetGeoMapsName（）方法获取以装载的地图名称列表。

2、*MapChanged*

当地图切换后，会发出该事件，并将当前新的地图名称作为参数传出。

3、*MouseMoveOnMap*

当鼠标在地图上移动时，发出该事件，并将当前鼠标所在的经纬度信息作为参数传出。

4、*RButtonUpTarmap*

当在地图上按鼠标右键，松开鼠标右键后会发出该事件，并将当前鼠标所在的经纬度信息作为参数传出，如果当前位置上有 GPS 目标，则会传出该目标的名称 (strTargetID)，否则名称 ID 会为空。

5、*RButtonDown*

当在地图上按下鼠标右键后会发出该事件，并将当前鼠标所在的经纬度信息作为参数传出。

6、*LButtonDown*

当在地图上按下鼠标左键后会发出该事件，并将当前鼠标所在的经纬度信息作为参数传出。

7、*LButtonUp*

当在地图上按鼠标左键，松开鼠标左键后会发出该事件，并将当前鼠标所在的经纬度信息作为参数传出，如果当前位置上有 GPS 目标，则会传出该目标的名称 (strTargetID)，否则名称 ID 会为空。

8、*EndGeoSelection*

矩形选择地物结束后，发出该事件，并将该矩形内的地物集合 pSelections 作为参数传出。

3、ITarSelections 对象

3.1 属性

无。

3.2 方法

1、Reset

用途：重置/清空 ITarSelections 集合中的内容

定义：HRESULT Reset()

输入参数：无

输出参数：无

返回：HRESULT

2、GetCount

用途：获取 ITarSelections 集合中地物的个数

定义：HRESULT GetCount([out,retval]int* nCount);

输入参数：nCount

输出参数：nCount

返回：nCount

3、Begin

用途：将 ITarSelections 集合的指针指向第一个地物。

定义：HRESULT Begin()

输入参数：无

输出参数：无

返回：无

4、*GetNext*

用途：获取下一个地物的名称

定义：BSTR GetNext();

输入参数：无

输出参数：无

返回：pName