

中地软件系列丛书

# MAPGIS

## 组件开发手册

武汉中地信息工程有限公司

北京中地时代软件工程有限公司

二 二年一月 武汉



## 内容简介

中地软件丛书之《MAPGIS 组件开发手册》是根据最新推出的 MAPGIS 软件平台编写而成，主要介绍组件的基本概念、MAPGIS 组件的体系结构以及利用组件进行 MAPGIS 二次开发的基本方法和必须具备的基础知识。

本书作为 MAPGIS 地理信息系统系列产品配套使用手册，供使用 MAPGIS 地理信息系统并利用组件进行二次开发的用户参考。

版权所有 武汉中地信息工程有限公司

警告： 未经武汉中地信息工程有限公司书面许可，任何单位和个人不得以任何形式或手段复制或传播本书的任何部分。

联系电话： 027-87561962，87561963，87561910

传真： 027-87561708，87483128

网址： [www.mapgis.com.cn](http://www.mapgis.com.cn)

电子信箱： [technic@mapgis.com.cn](mailto:technic@mapgis.com.cn)  
[market@mapgis.com.cn](mailto:market@mapgis.com.cn)



## 前 言

MAPGIS 是武汉中地信息工程有限公司研制的具有自主版权的大型基础激励信息系统平台。它是一个集当代最先进的图形、图像、地质、地理、遥感、测绘、人工智能、计算机科学于一体的大型智能软件系统,是集数字制图、数据库管理及空间分析为一体的空间信息系统,是进行现代化管理和决策的先进工具。MAPGIS 连续四年在全国 GIS 测评中名列第一,并成为唯一连续四年获得国家科技部向社会推荐的国产 GIS 软件平台。

中地软件丛书之《MAPGIS 组件开发手册》是根据最新推出的 MAPGIS 软件平台编写而成,主要介绍组件的基本概念、MAPGIS 组件的体系结构以及利用组件进行 MAPGIS 二次开发的基本方法和必须具备的基础知识。本书分为三个部分,共有十个章节和三个附录。其中:

第一部分(即第一章和第二章):介绍 MAPGIS 组件的特点、层次结构以及如何在常见的软件开发环境(主要是 VB 和 C++)中用 MAPGIS 组件进行二次开发。

第二部分(即第三章至第十章):详细介绍了 MAPGIS 组件和控件的方法、属性及事件。

第三部分(即附录部分):提供本书所有 MAPGIS 组件对象和控件的索引,方便读者迅速的查找。

参加本书编写的人员主要是本系统的软件开发人员以及长期从事 MAPCAD 和 MAPGIS 培训的教师。由于时间仓促,书中难免存在错误和不当之处,敬请广大用户及读者提出宝贵意见和建议,以利改进。

中地软件丛书编委会  
2002 年 1 月



# 目 录

<b>第一章 概述 .....</b>	<b>1</b>
1.1、MAPGIS 组件对象模型特性.....	1
1.2、MAPGIS 组件开发平台的层次结构.....	2
1.3、MAPGIS 组件开发平台的组件对象关系略图.....	3
1.4、MAPGIS 组件文件 .....	3
1.5、MAPGIS 组件对象关系图 .....	5
1.6、MAPGIS 组件对象接口图 .....	12
<b>第二章 MAPGIS 组件使用方法 .....</b>	<b>18</b>
2.1、安装 .....	18
2.2、使用 MAPGIS 对象 .....	18
2.3、嵌入 MAPGIS 对象 .....	19
2.4、初始化 COM 库 .....	21
2.5、建立 MAPGIS 对象 .....	22
2.6、声明和分配对象变量.....	23
2.7、使用属性 .....	23
2.8、调用方法 .....	25
2.9、使用集合 .....	26
2.10、释放 MAPGIS 对象 .....	31
2.11、解除 COM 库.....	31
2.12、Visual Basic 对象 .....	31
2.13、C++对象 .....	32
2.14、MAPGIS 组件使用示例代码（VB） .....	36
<b>第三章 MAPGIS 对象说明 .....</b>	<b>37</b>
3.1、D_Dot 对象.....	37
3.2、D_DotSet 对象.....	37
3.3、L_Dot 对象 .....	40
3.4、L_DotSet 对象.....	40
3.5、D_3Dot 对象.....	43
3.6、D_3DotSet 对象.....	43
3.7、L_3Dot 对象.....	46
3.8、L_3DotSet 对象.....	47
3.9、D_Rect 对象 .....	49
3.10、D_RectSet 对象 .....	50

3.11、Win_Org 对象.....	52
3.12、Date_Stru 对象.....	52
3.13、Time_Stru 对象.....	53
3.14、TimeStamp_Stru 对象.....	53
3.15、Field_ExtHead 对象.....	54
3.16、Field_Head 对象.....	58
3.17、Record_Head 对象.....	62
3.18、Field 对象.....	65
3.19、Record 对象.....	65
3.20、RecordSet 对象.....	67
3.21、GisEnv 对象.....	70
3.22、Tran_Par 对象.....	70
3.23、Map_Para 对象.....	71
3.24、Pnt_Info_Sub 对象.....	74
3.25、Pnt_Info_Note 对象.....	75
3.26、Pnt_Info_Cir 对象.....	76
3.27、Pnt_Info_Arc 对象.....	76
3.28、Pnt_Info_Image 对象.....	77
3.29、Pnt_Info_Text 对象.....	77
3.30、Pnt_Info 对象.....	79
3.31、Pnt_Pad_Note 对象.....	80
3.32、Pnt_Pad_Sub 对象.....	81
3.33、Pnt_Pad_Cir 对象.....	82
3.34、Pnt_Pad_Arc 对象.....	83
3.35、Pnt_Pad_Image 对象.....	83
3.36、Pnt_Pad_Text 对象.....	84
3.37、Pnt_Pad 对象.....	85
3.38、Lin_Info 对象.....	87
3.39、Lin_Pad 对象.....	89
3.40、Lin_Top 对象.....	90
3.41、Arc_Alloc_Info 对象.....	91
3.42、Reg_Info 对象.....	92
3.43、Reg_Pad 对象.....	94
3.44、Net_Info 对象.....	95
3.45、Net_Pad 对象.....	97
3.46、Nod_Center_Info 对象.....	99
3.47、Nod_Stop_Info 对象.....	100
3.48、Nod_Barrier_Info 对象.....	100



3.49、Pnt 对象 .....	101
3.50、PntSet 对象 .....	101
3.51、Lin 对象 .....	103
3.52、LinSet 对象 .....	104
3.53、Lin3D 对象 .....	105
3.54、Lin3DSet 对象 .....	106
3.55、Reg 对象 .....	107
3.56、RegSet 对象 .....	109
3.57、Net 对象 .....	111
3.58、NetSet 对象 .....	112
3.59、Path 对象 .....	114
3.60、PathSet 对象 .....	117
3.61、GeoPnt 对象 .....	118
3.62、GeoPntSet 对象 .....	119
3.63、GeoLin 对象 .....	121
3.64、GeoLinSet 对象 .....	121
3.65、GeoLin3D 对象 .....	123
3.66、GeoLin3DSet 对象 .....	124
3.67、GeoReg 对象 .....	125
3.68、GeoRegSet 对象 .....	127
3.69、GeoNet 对象 .....	129
3.70、GeoNetSet 对象 .....	130
3.71、GeoPath 对象 .....	132
3.72、GeoPathSet 对象 .....	135
3.73、LayerOnOffPad 对象 .....	136
3.74、LayerPad 对象 .....	137
3.75、IDList 对象 .....	138
3.76、LONGList 对象 .....	140
3.77、RepFldFlag 对象 .....	142
3.78、Tic_Dot 对象 .....	143
3.79、Tic_DotSet 对象 .....	144
3.80、Ask_Db_Rcd 对象 .....	145
3.81、Ext_Link_Db_Info 对象 .....	146
3.82、Rad 对象 .....	146
3.83、RadSet 对象 .....	147
3.84、ExtDbTool 对象 .....	148
3.85、Layer 对象 .....	157
3.86、PntAtt 对象 .....	163

3.87、LinAtt 对象.....	167
3.88、RegAtt 对象.....	171
3.89、NodAtt 对象 .....	176
3.90、NetAtt 对象 .....	180
3.91、NodData 对象.....	184
3.92、LinData 对象 .....	202
3.93、PathData 对象.....	211
3.94、WorkArea 对象.....	214
3.95、VectWorkArea 对象.....	218
3.96、PntArea 对象 .....	220
3.97、LinArea 对象 .....	237
3.98、RegArea 对象 .....	253
3.99、NetArea 对象.....	273
3.100、TblArea 对象.....	290
3.101、ExpObj 对象.....	301
3.102、Query 对象 .....	301
3.103、Analysis 对象 .....	319
3.104、TraceDEM 对象 .....	326
3.105、MapGis 对象 .....	330

#### **第四章 MAPGIS 控件说明.....364**

4.1、Arc_Alloc_InfoEdit 控件.....	364
4.2、Lin_InfoEdit 控件 .....	365
4.3、Lin_PadEdit 控件 .....	367
4.4、Lin_TopEdit 控件.....	369
4.5、Net_InfoEdit 控件 .....	371
4.6、Net_PadEdit 控件.....	373
4.7、Nod_Center_InfoEdit 控件 .....	375
4.8、Pnt_Info_ArcEdit 控件.....	377
4.9、Pnt_Info_CirEdit 控件.....	379
4.10、Pnt_Info_ImageEdit 控件.....	381
4.11、Pnt_Info_NoteEdit 控件 .....	382
4.12、Pnt_Info_SubEdit 控件 .....	384
4.13、Pnt_Info_TextEdit 控件.....	386
4.14、Pnt_Pad_ArcEdit 控件 .....	388
4.15、Pnt_Pad_CirEdit 控件 .....	390
4.16、Pnt_Pad_ImageEdit 控件 .....	391
4.17、Pnt_Pad_NoteEdit 控件.....	393

4.18、Pnt_Pad_SubEdit 控件 .....	395
4.19、Pnt_Pad_TextEdit 控件.....	397
4.20、Reg_InfoEdit 控件.....	399
4.21、Reg_PadEdit 控件 .....	401
<b>第五章 显示对象说明 .....</b>	<b>403</b>
5.1、MapGisDC 对象 .....	403
<b>第六章 Layer 和 Map 说明 .....</b>	<b>426</b>
6.1、MapLayer 对象.....	426
6.2、GroupLayer 对象 .....	427
6.3、VectorLayer 对象.....	433
6.4、LinLayer 对象.....	434
6.5、NetLayer 对象 .....	436
6.6、PntLayer 对象.....	436
6.7、RegLayer 对象.....	436
6.8、Map 对象 .....	437
6.9、LegendItem 对象 .....	442
6.10、PntLegendItem 对象 .....	443
6.11、LinLegendItem 对象 .....	444
6.12、RegLegendItem 对象.....	445
6.13、MapLegend 对象 .....	446
6.14、MapPrintParam 对象 .....	450
6.15、MapNorthArrow 对象 .....	451
6.16、MapScaleBar 对象.....	451
6.17、MapScaleText 对象.....	451
6.18、MapSurround 对象 .....	451
6.19、SelectItem 对象 .....	451
6.20、SelectSet 对象.....	452
<b>第七章 MSI 对象说明.....</b>	<b>454</b>
7.1、RasterArea 对象.....	454
7.2、RasterLayer 对象 .....	483
7.3、VectorToMsi 对象.....	484
7.4、MsiToVector 对象.....	485
<b>第八章 DBS 对象说明.....</b>	<b>489</b>
8.1、DBS_FramInfo 对象 .....	489

8.2、MSystemTime 对象.....	490
8.3、DBS_FramFile 对象.....	491
8.4、DBS_FramFileSet 对象.....	492
8.5、DBS_GraIndex 对象.....	495
8.6、DBS_GraIndexSet 对象.....	495
8.7、DBSArea 对象.....	498
8.8、DBSLayer 对象.....	528
<b>第九章 数字高程模型对象说明.....</b>	<b>534</b>
9.1、DemInfoStru 对象.....	534
9.2、DemInsObj 对象.....	535
9.3、DistPowGridStru 对象.....	535
9.4、KringGridStru 对象.....	536
9.5、LabelFontStru 对象.....	536
9.6、LabelFormatStru 对象.....	537
9.7、SearchDataStru 对象.....	538
9.8、SlopeLineStru 对象.....	538
9.9、TinNetDispStru 对象.....	539
9.10、TinPnt 对象.....	540
9.11、TinPntSet 对象.....	541
9.12、TinNet 对象.....	541
9.13、TinNetSet 对象.....	543
9.14、ZLevelStru 对象.....	543
9.15、ZLevelStruSet 对象.....	544
9.16、TinArea 对象.....	545
9.17、TinModel 对象.....	566
9.18、GridModel 对象.....	577
9.19、AppModel 对象.....	591
9.20、TinLayer 对象.....	601
<b>第十章 可视化控件说明.....</b>	<b>604</b>
10.1、EditView 控件.....	604
10.2、MapTreeCtrl 控件.....	641
10.3、GisAttEdit 控件.....	642
10.4、AttStruEdit 控件.....	648
<b>附 录 MAPGIS 组件对象索引.....</b>	<b>652</b>

# MAPGIS 组件开发手册

## 第一章 概述

组件对象模型 (COM) 技术给程序设计理念带来了新的突破, 并且标志着未来应用程序的发展趋势。

组件技术给 GIS 的发展注入了新的活力, 从用户的角度来看, 组件就是一系列软件模块, 你可以轻而易举的将这些模块组装起来构建满足自己需求的、功能强大的应用程序。传统的 API 函数提供的是生鲜超市中的一条鱼一根葱一棵白菜, 彼此之间没什么联系, 掌勺的你必须选材自己调理, 而组件提供给你的是火锅拼盘, 你要的是白菜火锅鱼头火锅或麻辣火锅, 带回家统统下锅就好。

程序设计领域里, 每个人都想飞。MAPGIS 组件为您的开发插上新技术的翅膀, 让每个应用程序员在 GIS 程序设计的天空中自由翱翔。

### 1.1、MAPGIS 组件对象模型特性

组件化 MAPGIS 集成了更多的先进技术, 功能更加强大, 使用更加便利。具有二次开发方便、易于集成、无限扩展等特点。

#### 1.1.1 功能完备

MAPGIS 以组件的方式提供了完善的 GIS 功能: 包括高性能的空间数据库管理、完备的空间分析工具、属性数据分析、网络分析功能、多源图像分析与处理、三维模型显示、地图编辑等。

#### 1.1.2 可扩展的体系结构

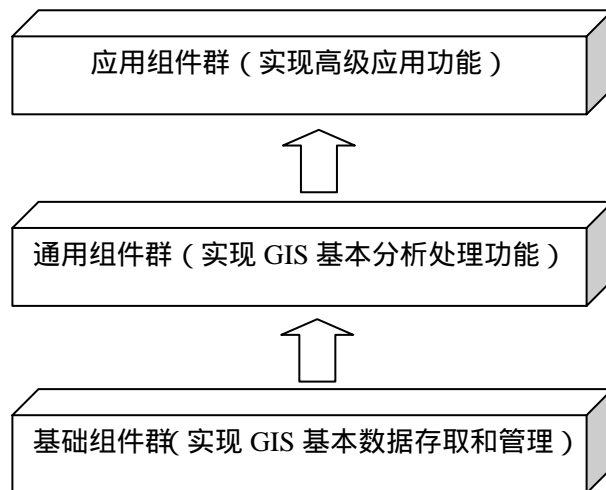
系统基于的标准是开放的, 这就为组件化 MAPGIS 的进一步扩展提供了极大的发挥空间, 使得组件化 MAPGIS 很容易与诸如 Web 等其他信息服务进行无缝集成, 建立功能丰富的具体 GIS 应用。通过编程语言用户可以定制产品建立客户化的用户界面。

#### 1.1.3 方便的二次开发

作为组件式 GIS 软件开发工具, 以“控件+对象”的形式, MAPGIS 支持多种开发软件, 包括 Visual C++、Visual Basic、Delphi、Script、InterDev 等, 开发者可以通过 MAPGIS 的软件开发包开发特定的应用。软件开发包使开发者可以多层次的访问组件的数据层、表现层和界面层。

## 1.2、MAPGIS 组件开发平台的层次结构

组件式 MAPGIS 开发平台设计为三级结构：



基础组件群：面向空间数据的管理，提供基本的数据交换和组织。

通用组件群：面向通用功能，提供 GIS 的通用处理。

应用组件群：抽象出行业应用的特定算法，固化到组件中，进一步加速开发过程。

### 1.2.1 基础组件

处于平台最低层，是整个系统的基础和核心，主要进行空间数据的存取和管理，提供基本的交互过程，并能以灵活的方式与数据库系统连接。

### 1.2.2 通用组件

通用组件由基础组件构造而成。它们面向通用功能，简化用户开发过程，如图形显示和编辑组件、属性浏览组件、图像分析组件等等。它们之间的协同控制消息都被封装起来。这级组件经过封装后，使二次开发更为简单。

### 1.2.3 应用组件

应用组件的设计是以实际工程应用为目标，它除了需要通用组件所拥有的一般 GIS 功能外，还需要针对不同行业的通用特点，进行分析和定制，把不同的应用功能按照组件的模式封装组合起来。在这一层次进行二次开发，工作量就可以大大简化。

### 1.3、MAPGIS 组件开发平台的组件对象关系略图

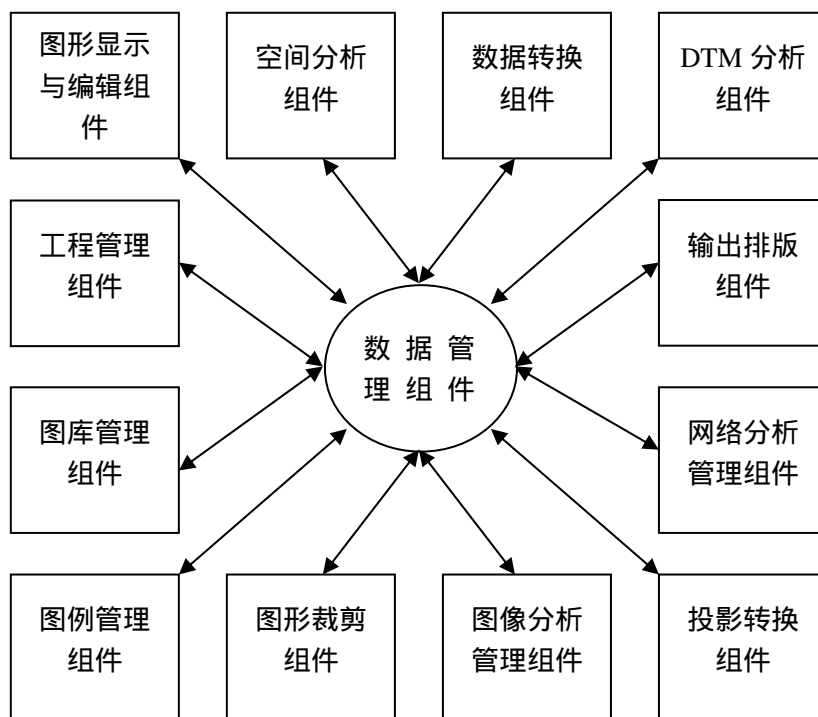


图 1-1 组件对象关系略图

MAPGIS 组件不依赖于某一种开发语言，可以直接嵌入到某些通用的开发环境(如：Visual Basic 或 Delphi)中实现 GIS 功能。而其他的专业模型则可以使用这些通用开发环境来实现，也可以插入其它的专业性模型的分析控件，各个模块之间既可相互关联共同处理数据，又可在维护修改时独立操作互不影响，因此，利用 MAPGIS 组件进行 GIS 应用系统的开发可以实现高效、无缝的系统集成（集成方式见图 1-2）。

### 1.4、MAPGIS 组件文件

基于 COM 技术的 MAPGIS 组件被包含在一系列 Windows 动态连接库（dynamic link library, DLL）文件里，ActiveX 控件则被包含在一系列 OCX（.ocx）文件里。使用时要对这些组件文件进行注册，具体过程详述见第二章，以下是 MAPGIS 组件文件：

MapGisBasCom1.dll

MapGisBasCtl.dll

MapGisBasCtl1.dll

MapGisBasCtl2.dll  
MapGisDspCom.dll  
MapGisLayerCom.dll  
MapGisMapCom.dll  
MapGisDBSCom.dll  
MapGisRasterCom.dll  
EditView.ocx  
MapTree.ocx  
AttStruEdit.ocx  
GisAttEdit.ocx

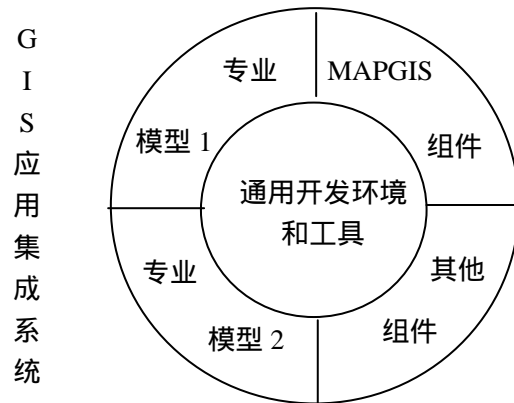
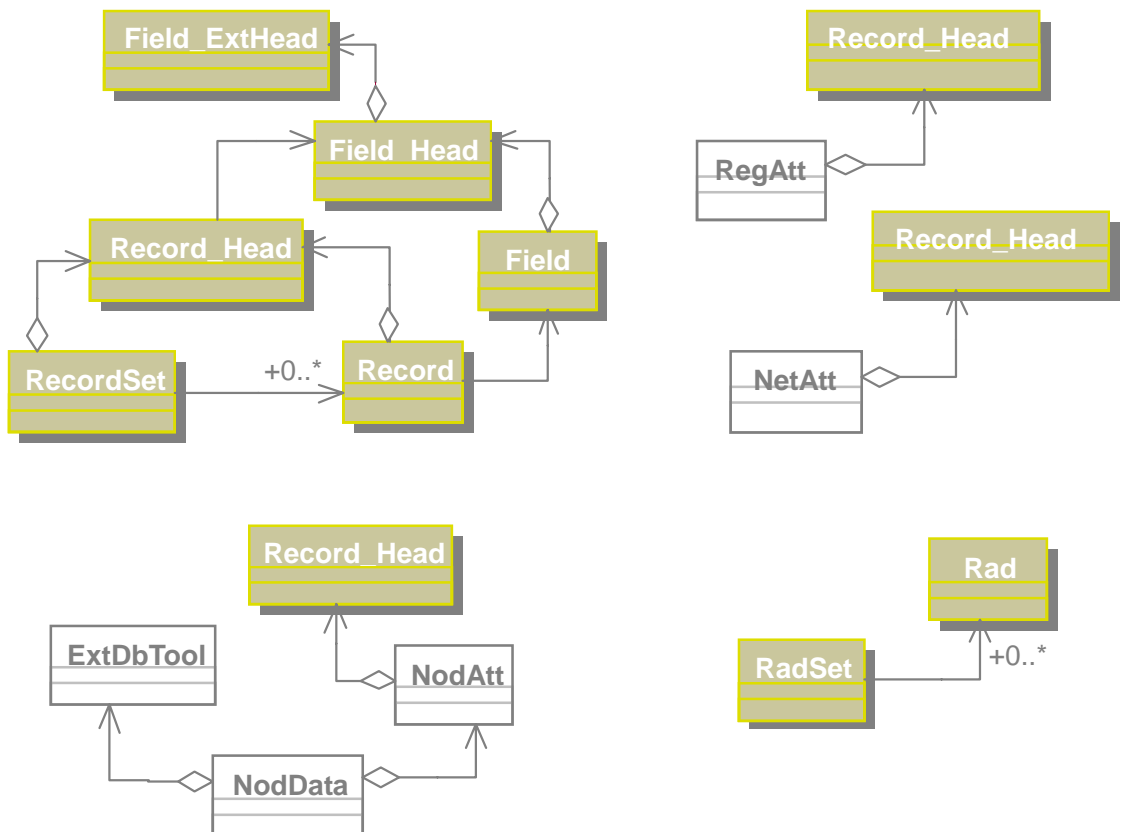
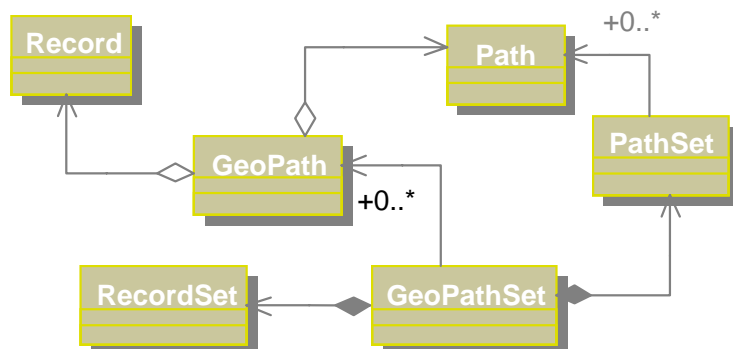
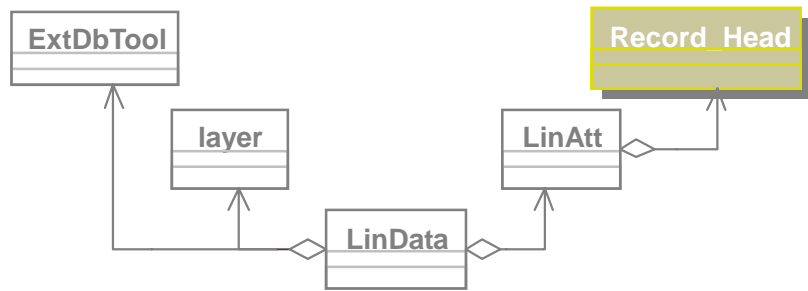
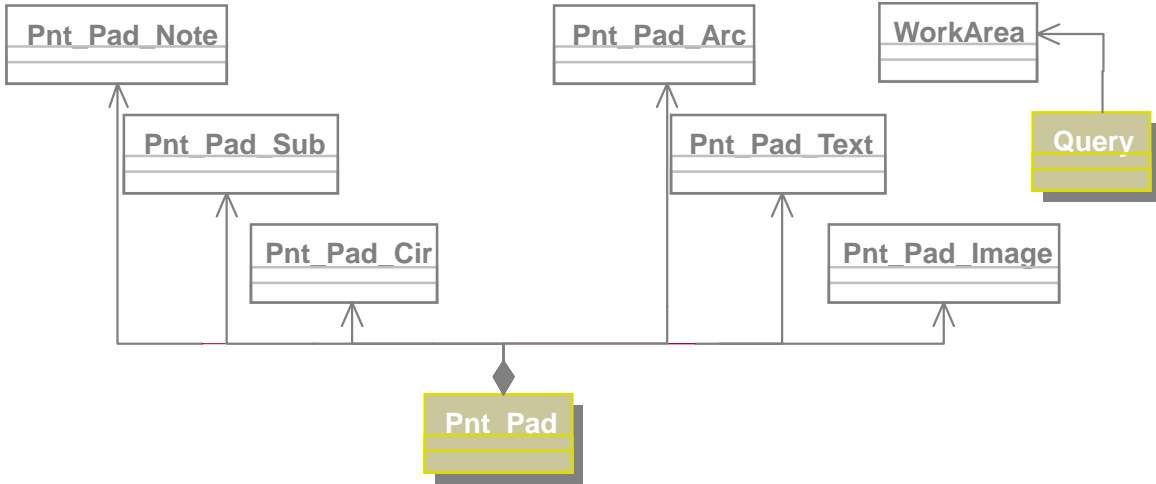


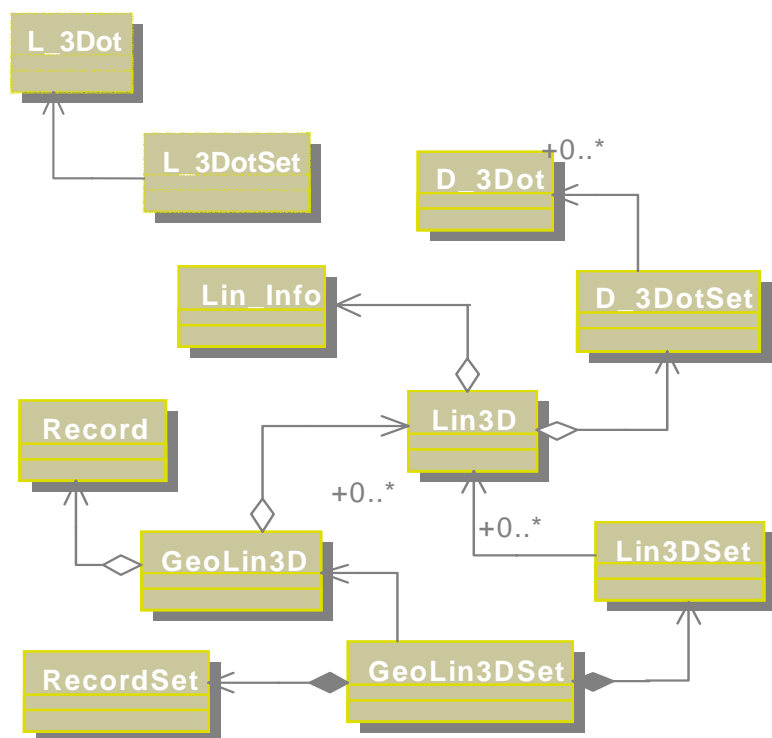
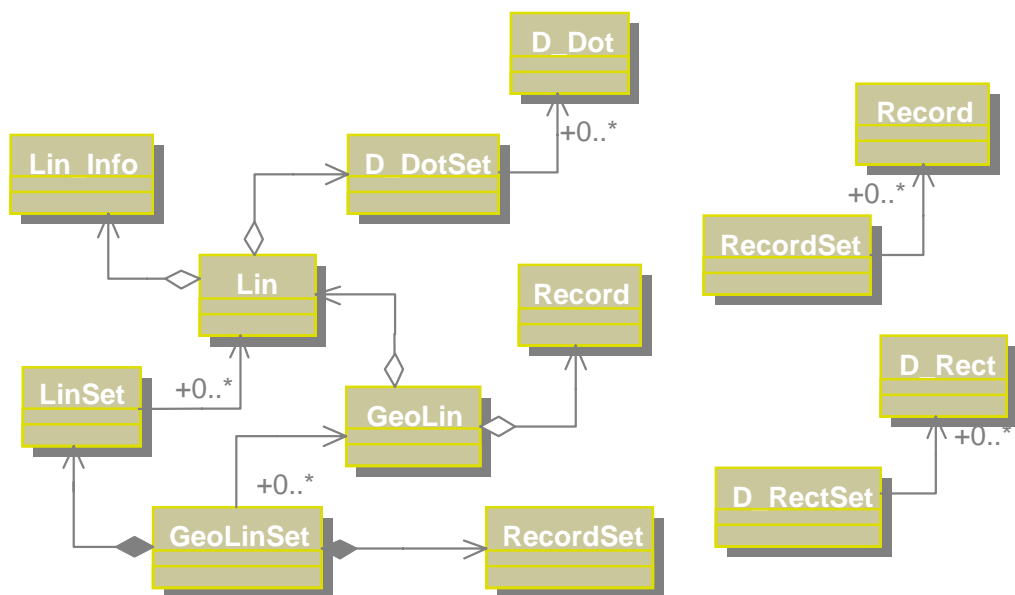
图 1-2 MAPGIS 组件与其他功能模块集成关系

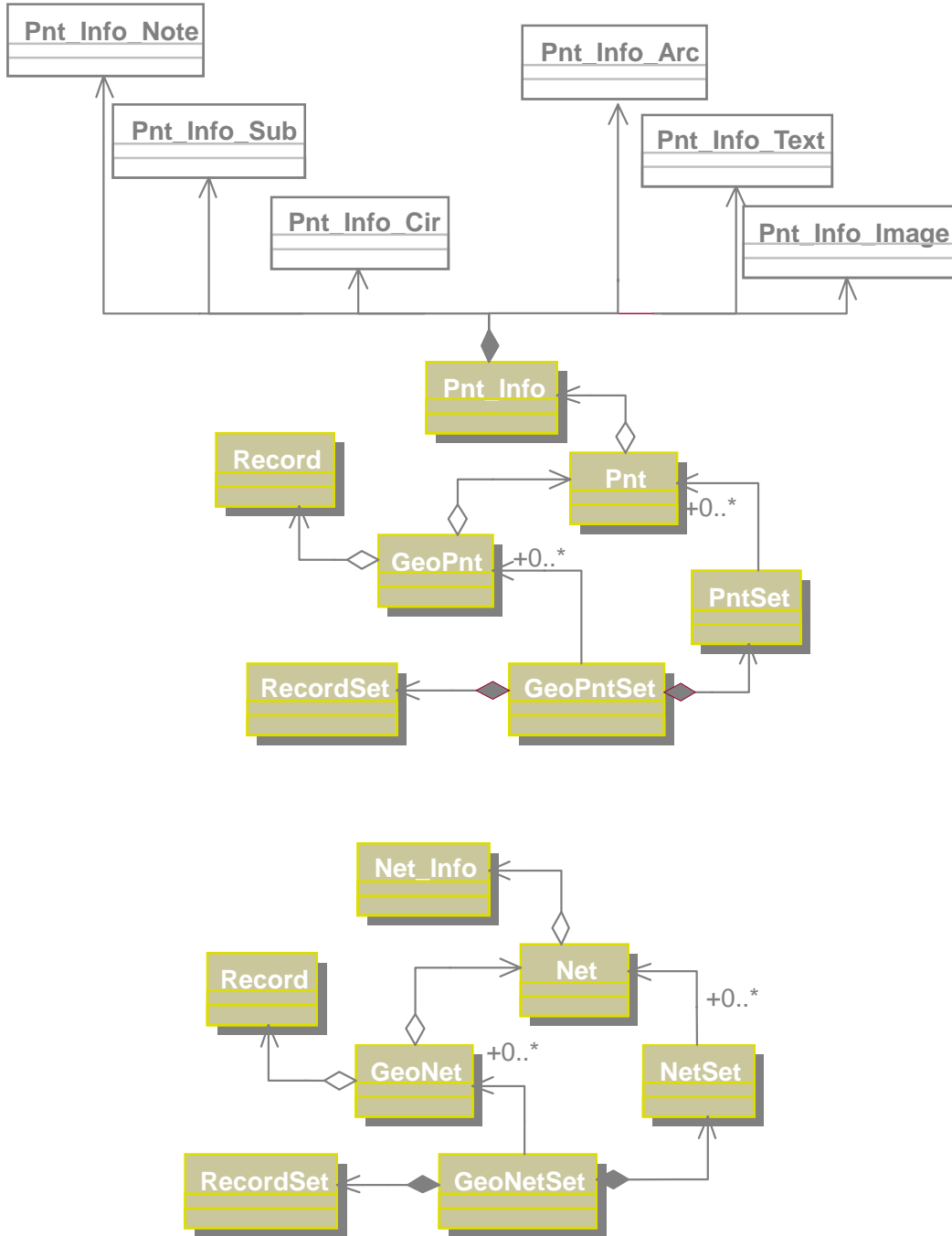


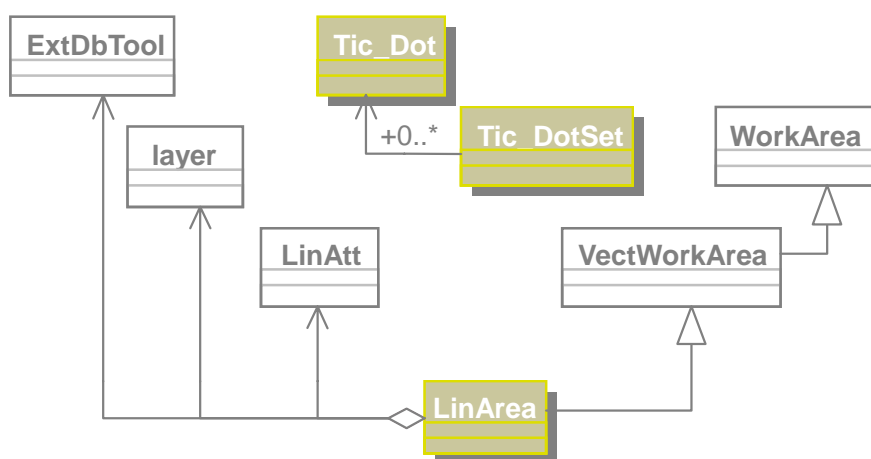
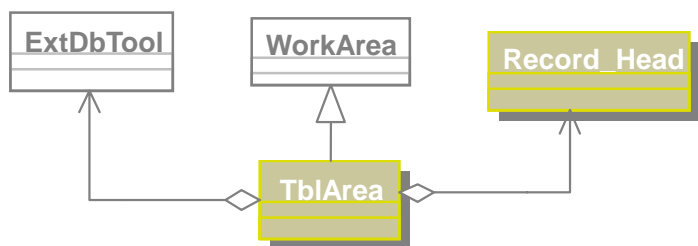
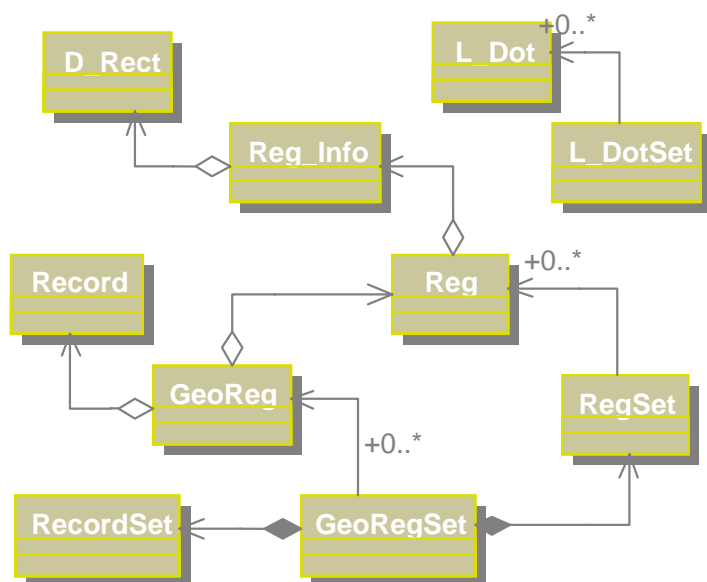
## 1.5、MAPGIS 组件对象关系图

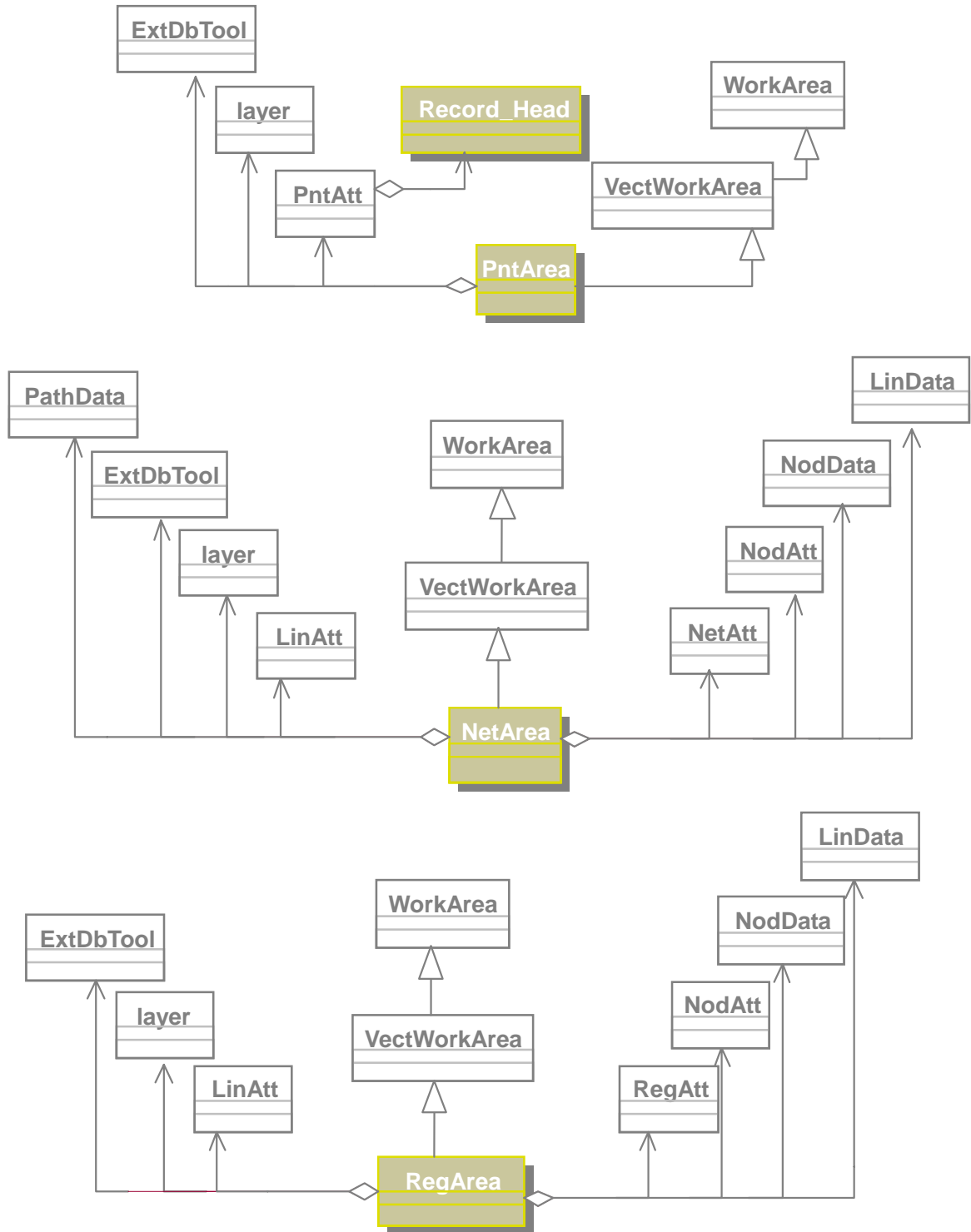


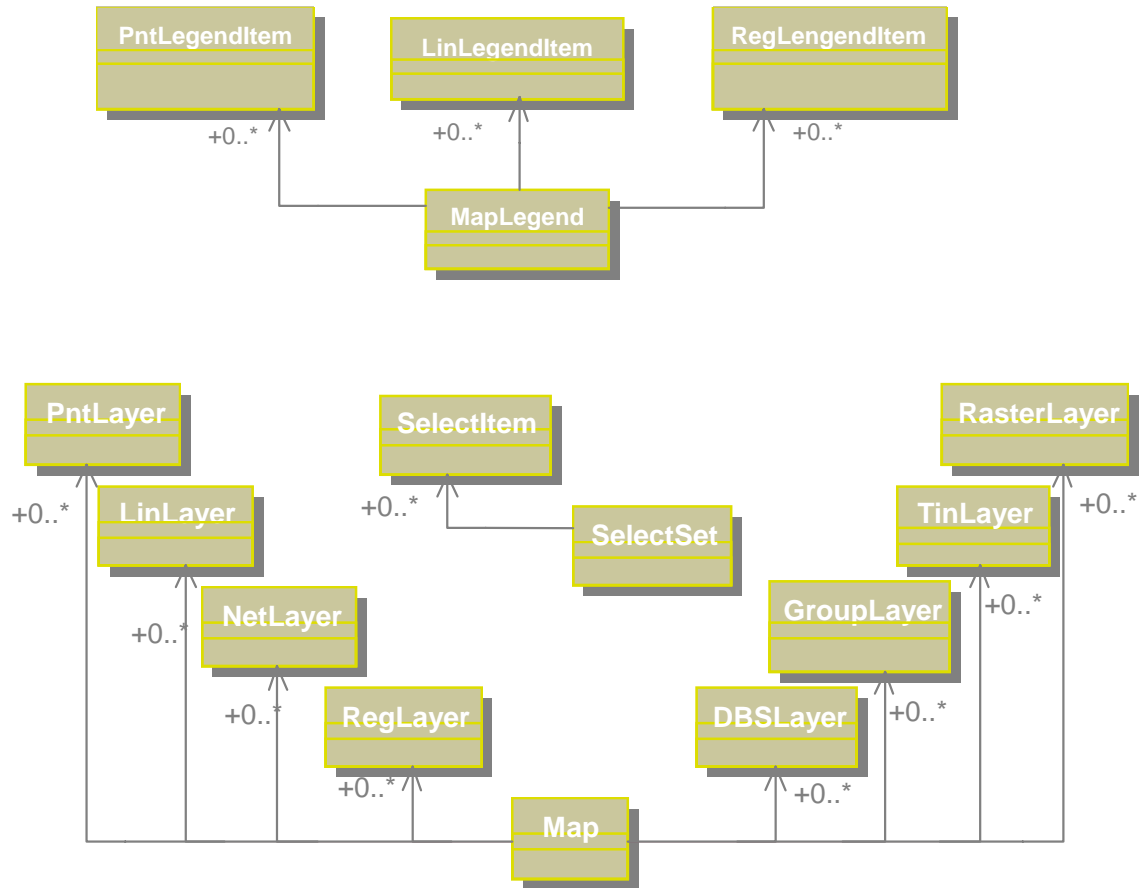




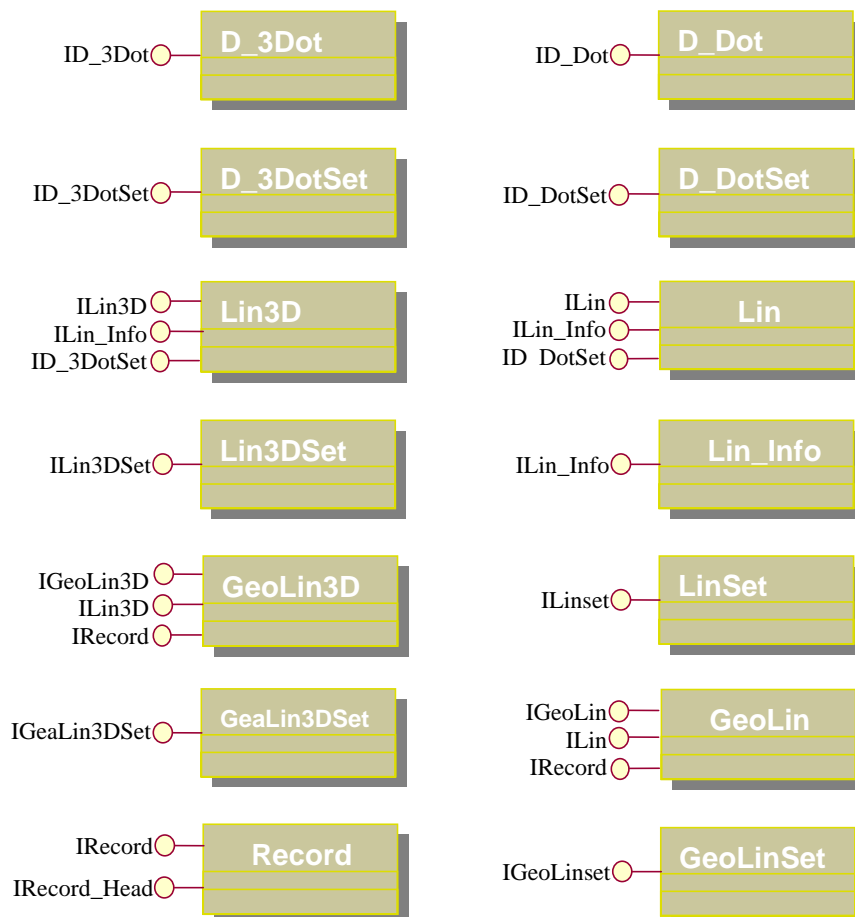
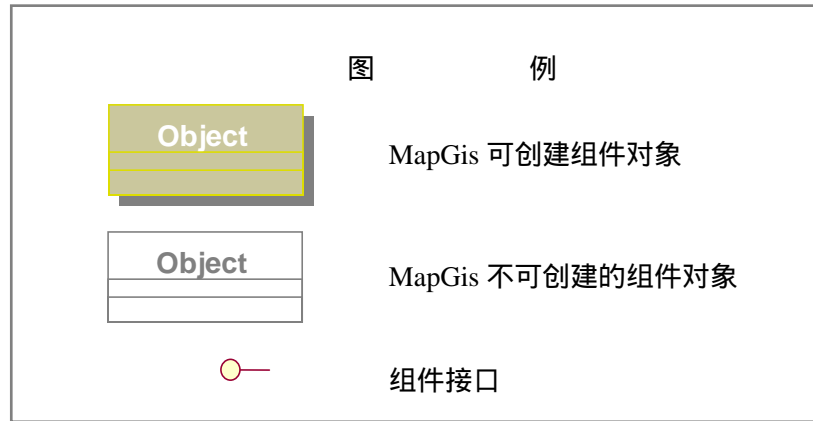




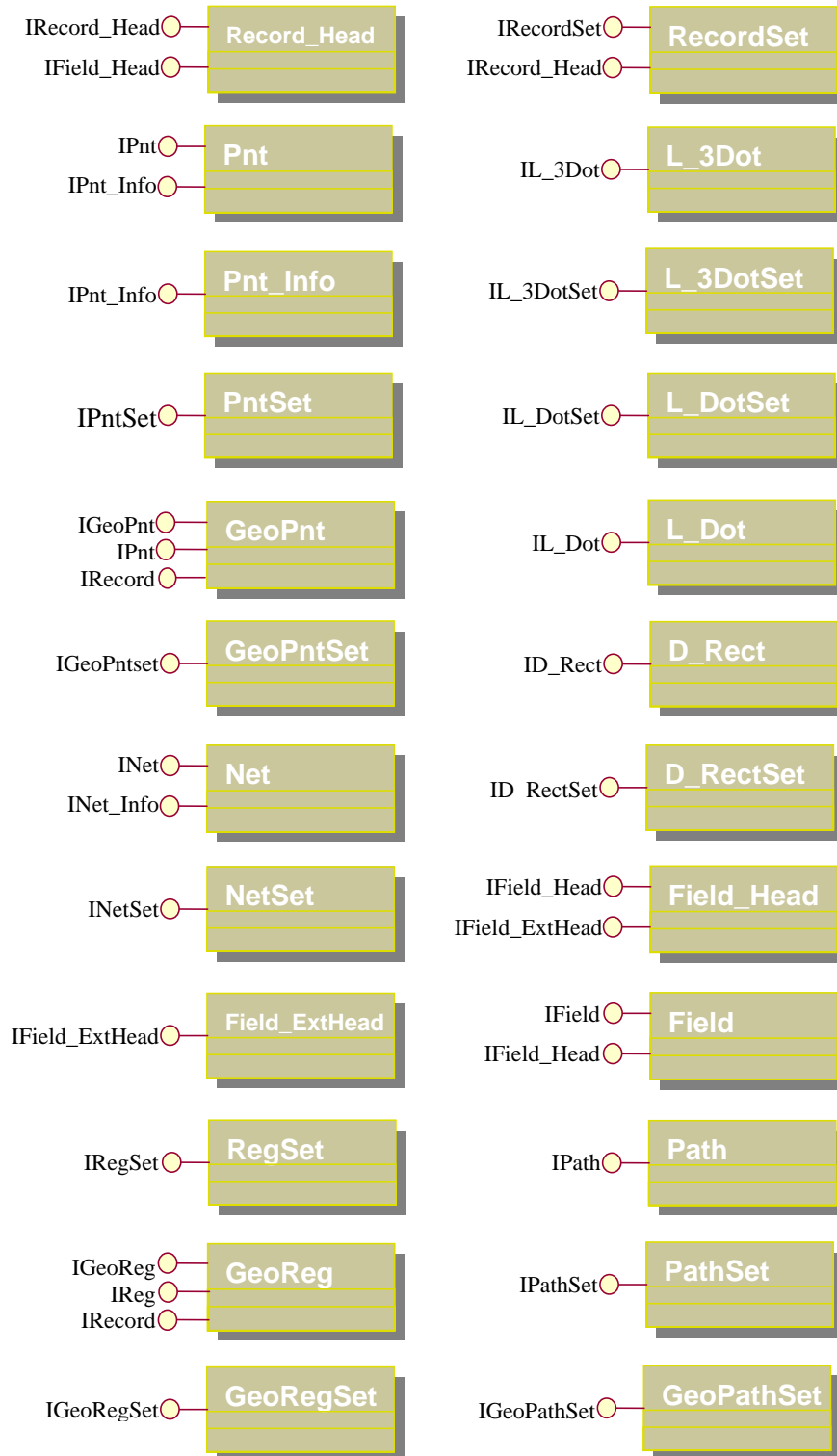


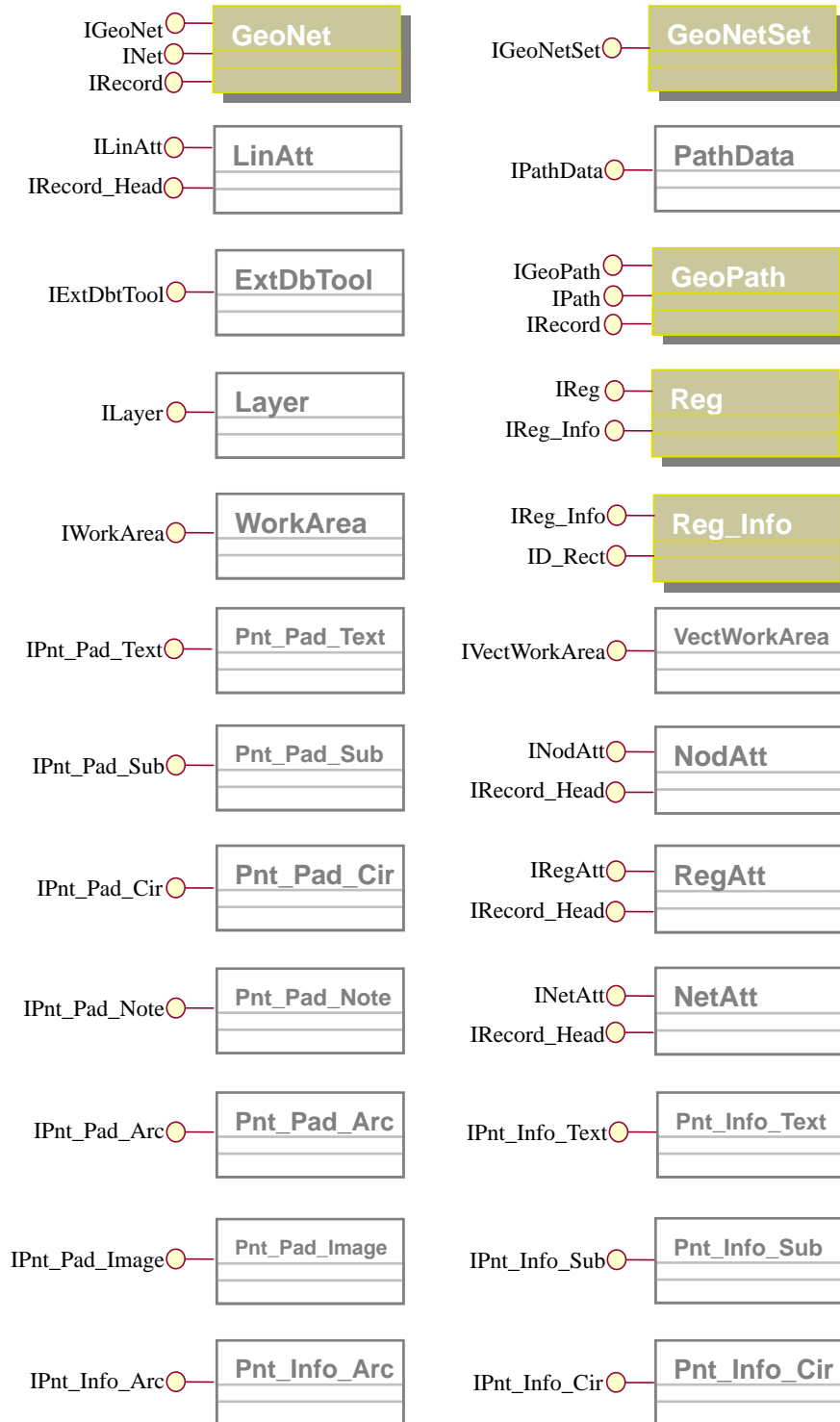


## 1.6、MAPGIS 组件对象接口图

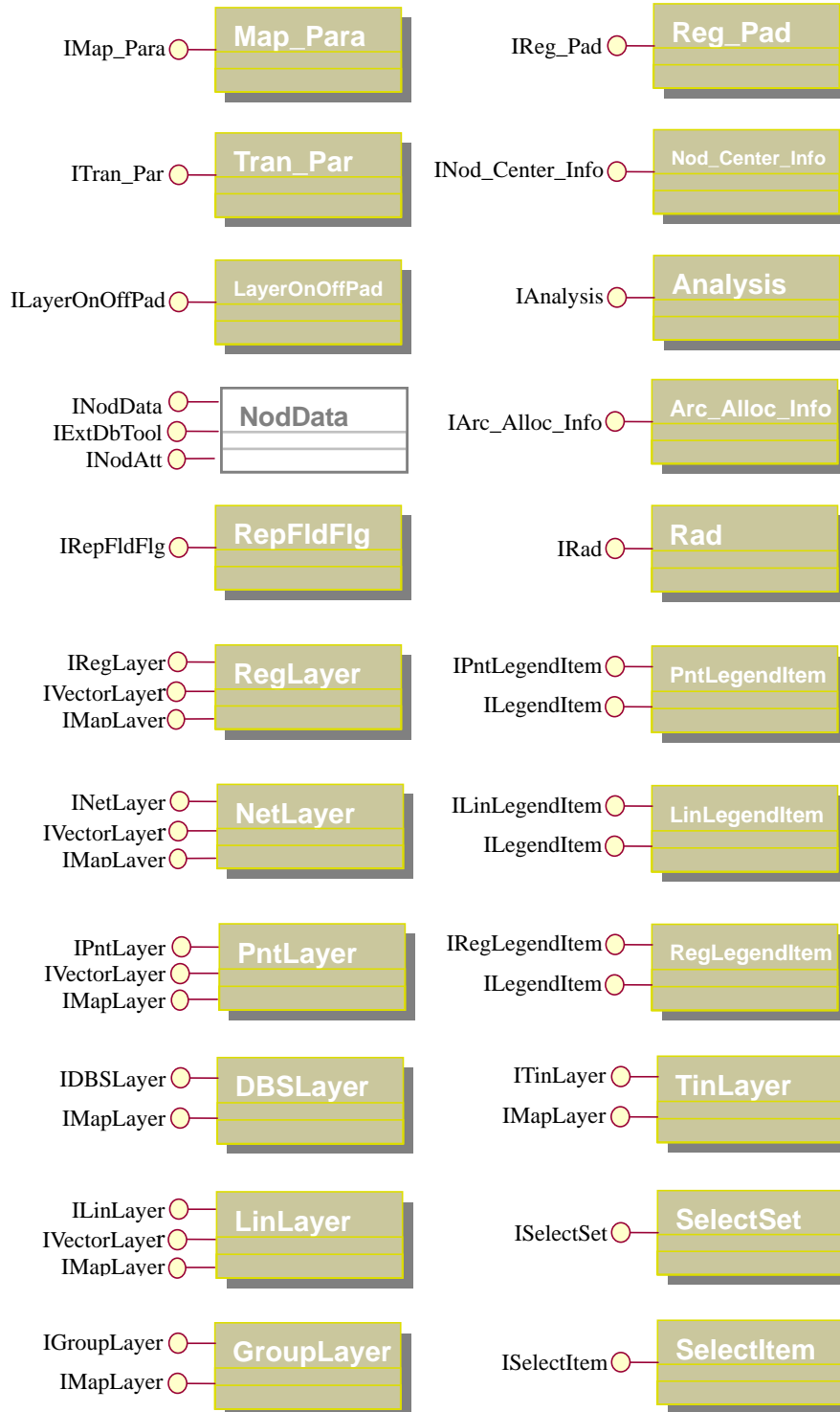


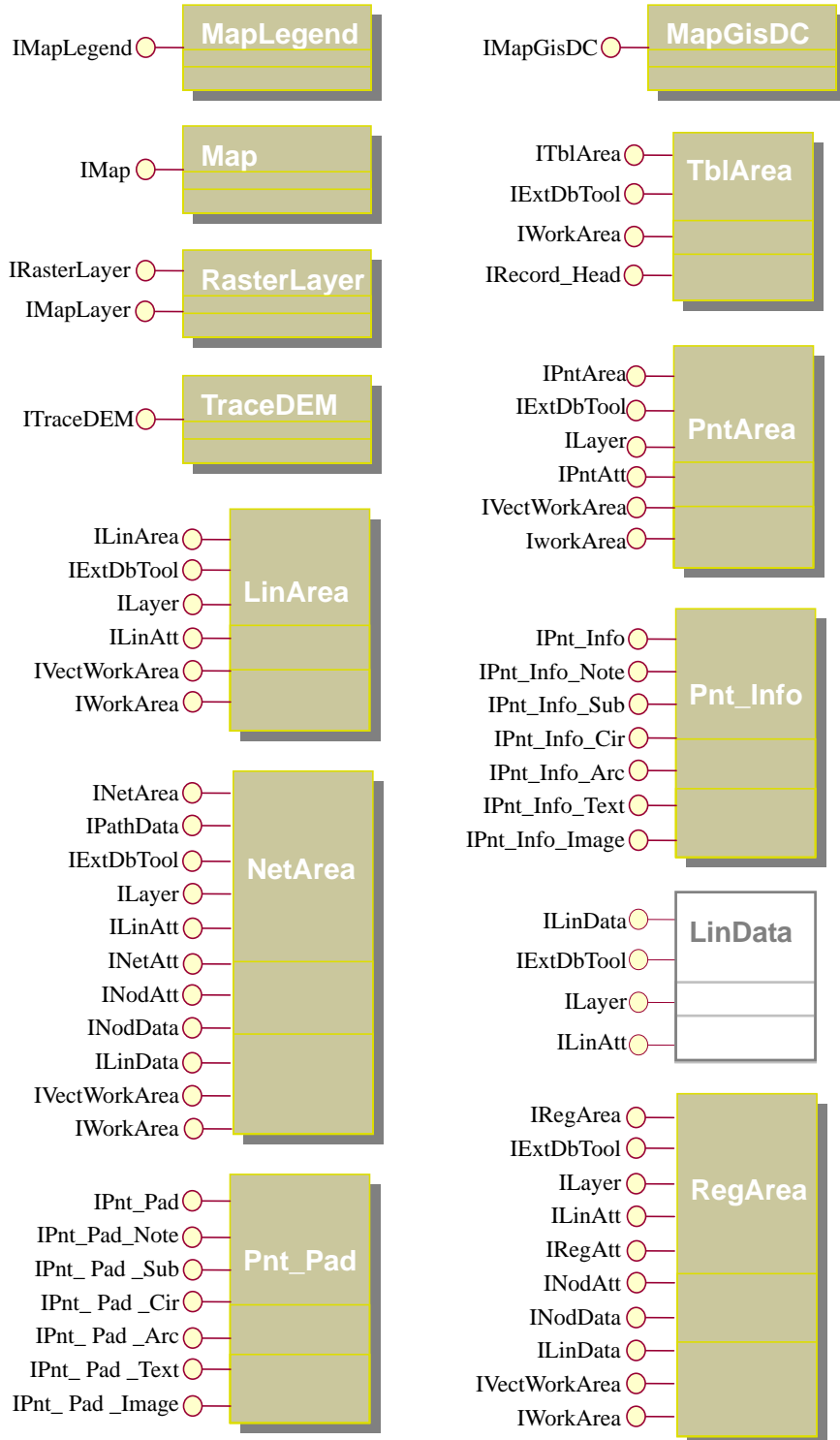












## 第二章 MAPGIS 组件使用方法

本章介绍如何在 Visual Basic 和 C++ 程序中使用MAPGIS组件对象。MAPGIS 组件提供32位 COM 对象，从32位 Visual Basic 程序和32位 C++ 程序可以使用这些 COM 对象。

为了简单明了，本章包括的范例代码不检查成功或处理错误。

### 2.1、安装

要使用 MapGis 的组件，必须对其安装并在 Windows 注册表里注册。运行 MapGis 二次开发包中的 Setup.exe 程序，可以自动安装组件并注册。二次开发包在光盘目录 SetUpSDK??中，??表示版本号。

在某些情况下，用户需要对单个组件文件进行注册或注销，则按以下步骤进行：(可注册的组件请参阅 1.4 MAPGIS 组件文件)。

1. 将组件文件拷贝到 MAPGIS 的 PROGRAM 目录下，假设路径为：

D:\mapgis61\program\

2. 注册组件方法举例，Windows 开始菜单->运行：

regsvr32 D:\mapgis61\program\MapGisBasCom1.dll

注册成功与否将有对话框弹出。

3. 注销组件方法举例，Windows 开始菜单->运行：

regsvr32 /u D:\mapgis61\program\MapGisBasCom1.dll

### 2.2、使用 MAPGIS 对象

#### Visual Basic

可以使用 Visual Basic 编写使用 MAPGIS 对象的程序。在 Visual Basic 程序中使用 MAPGIS 对象所要求的步骤是：

1. 嵌入 MAPGIS 对象类型库。
2. 建立 MAPGIS 对象。
3. 使用 MAPGIS 对象和其他 VB 对象。
4. 释放 MAPGIS 对象。

使用一个句点(.)把对象变量与该对象的属性、方法和集合分开。

#### C++

可以使用C++和COM函数编写使用MAPGIS对象的程序。在C++程序中使用MAPGIS对象所要求的步骤是：

1. 嵌入MAPGIS对象和COM头文件，并链接到COM库。

2. 初始化COM库。
3. 建立MAPGIS对象。
4. 使用MAPGIS对象和其他对象。
5. 释放MAPGIS对象。
6. 解除COM库的初始化。

使用成员选择运算符(->)把“对象指针”变量与属性、方法和集合函数分开。

## 2.3、嵌入 MAPGIS 对象

对MAPGIS对象编程的第一步是在项目中嵌入合适的类型库或头文件。

Visual Basic

为了使Visual Basic能够访问MAPGIS对象，应该在Visual Basic项目或模块中嵌入MAPGIS对象类型库。如果想把一个Visual Basic对象声明为MAPGIS对象，要求嵌入MAPGIS对象类型库，如果想把一个Visual Basic对象声明为一般Object类型，就不必嵌入MAPGIS对象。

为了在项目中嵌入 MAPGIS 对象类型库，通常执行下列步骤。从 Project(工程)菜单中选择References(引用)，References 对话框出现后，选中复选框中所需的类型库，如 MapGisBasCom1 1.0 Type Library(如下图所示)。这要求 Visual Basic 环境或代码模块编辑器是活动的，正在运行。

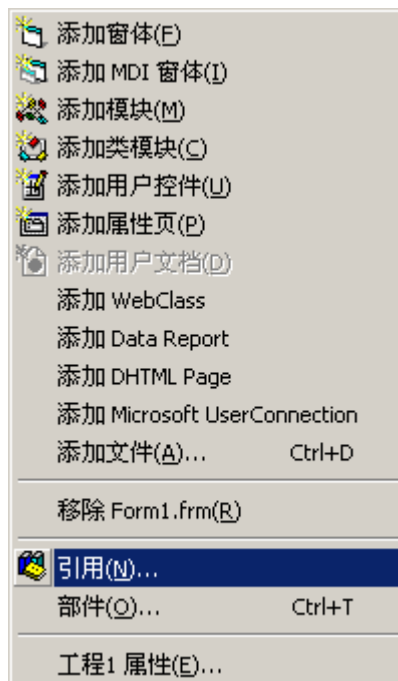


图 2-1 Visual Basic Project(工程)菜单

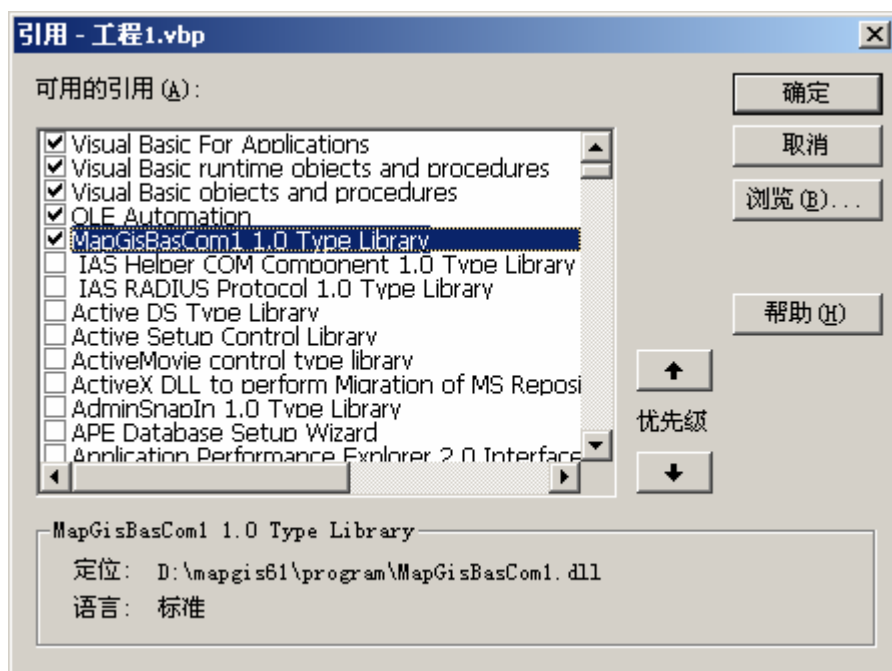


图 2-2 Visual Basic References (引用)对话框

## C++

MAPGIS对象完整的C语言说明包含在下列\*.h文件中,对应的组件类ID和接口ID则包含在\*.c文件中:

基础组件: MapGisBasCom1.h、 MapGisBasCom\_i.c  
 显示组件: MapGisDspCom.h、 MapGisDspCom\_i.c  
 图层组件: MapGisLayerCom.h、 MapGisLayerCom\_i.c  
 Map组件: MapGisMapCom.h、 MapGisMapCom\_i.c  
 底图库组件: MapGisDBSCom.h、 MapGisDBSCom\_i.c  
 栅格组件: MapGisRasterCom.h、 MapGisRasterCom\_i.c

为了使C++程序能够访问MAPGIS对象定义,除了嵌入Windows和OLE头文件之外,还必须嵌入MAPGIS???H和MAPGIS???C文件。默认情况下,MAPGIS头文件安装在MapGis的INCLUDE目录中。Visual C++项目的Include设置或INCLUDE环境变量应指向含有这些文件的目录。例如,C++程序中要使用MAPGIS基础组件,则必须包含以下语句:

```
#include "MapGisBasCom1.h"
#include "MapGisBasCom1_i.c"
```

Visual C++项目的 Library 设置或 LIB 环境变量应指向含有这些文件的目录(如下图所示)。



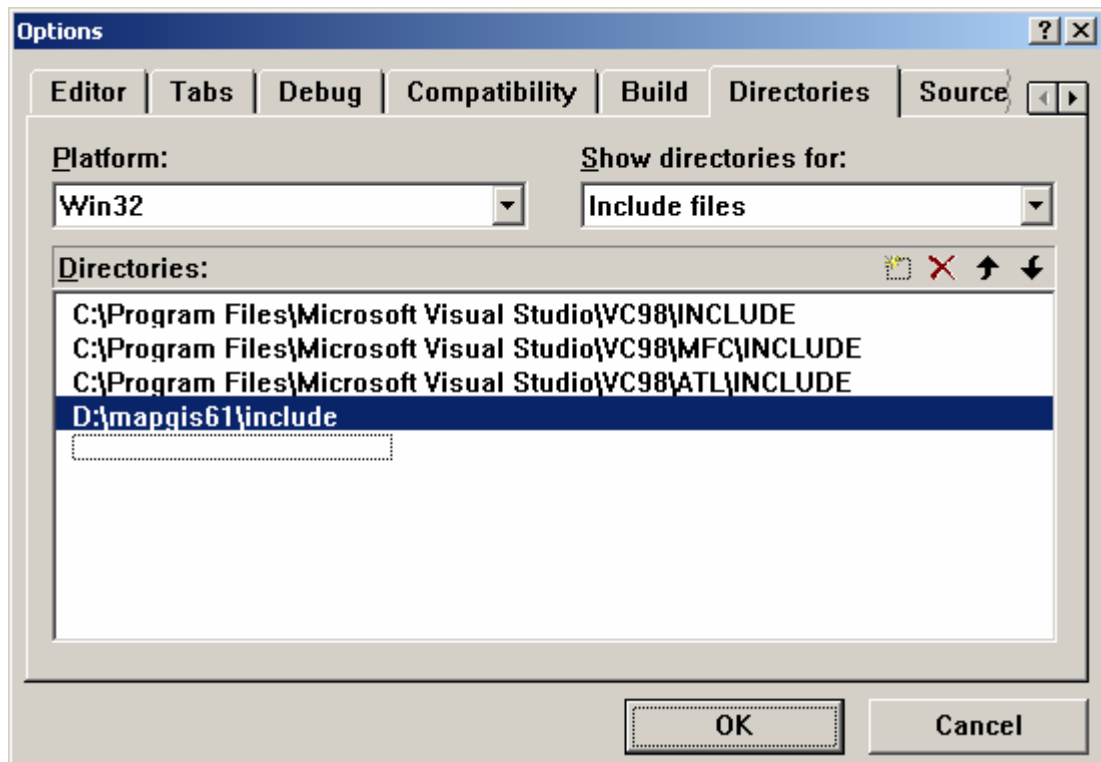


图 2-3 Visual C++ Options 对话框

## 2.4、初始化 COM 库

在程序能使用MAPGIS对象之前，必须初始化COM库。

Visual Basic

MAPGIS应用程序不必初始化COM，Visual Basic会自动初始化COM库。

C++

所有MAPGIS COM函数都返回一个HRESULT值，这个值含有关于MAPGIS COM函数的成功或失败的信息。可以声明单个HRESULT变量并重复利用它。例如：

```
HRESULT hr;
```

在调用任何MAPGIS组件对象之前，MAPGIS应用程序必须使用CoInitialize函数初始化COM库。

例如：

```
hr=CoInitialize(NULL);
```

必须把NULL传递给CoInitialize。

## 2.5、建立 MAPGIS 对象

MAPGIS组件的核心是MAPGIS对象。

Visual Basic

在Visual Basic中有两个方法用于建立MAPGIS对象,这两个方法都涉及使用Dim语句声明特定类型的对象变量。

特定的MAPGIS对象类型

常用的方法是使用Dim语句声明特定MAPGIS的对象变量,这个方法要求MAPGIS类型库包括在Visual Basic项目中。为了实际建立一个MAPGIS对象的新实例,可以在Dim语句中使用New关键字或者使用CreateObject函数。

例如,使用New关键字创建新的MAPGIS对象的语句是:

```
Dim ldot As New L_DOT
```

例如,使用CreateObject函数声明和创建一个新的MAPGIS对象:

```
Dim ldot As L_DOT
```

```
Set ldot = CreateObject("MapGisBasCom1.L_DOT")
```

有些版本的Visual Basic使用虚函数表绑定(vtable-binding),或早期绑定。直接使用MAPGIS定制接口(不通过COM IDispatch接口),产生最好的性能。

有些版本的Visual Basic使用推迟绑定的形式,叫做分派ID绑定(dispatch, binding),它使用COM IDispatch接口。IDispatch::Invoke 函数与在MAPGIS类型库中包括的dispID一起使用。这种方法使应用的性能没有使用New关键字那样快,但比使用Object类型快。

C++

在C++中只有一种方法建立MAPGIS对象,声明合适的MAPGIS“对象指针”变量,然后使用COM CoCreateInstance函数建立MAPGIS对象的新实例。把以下参数传递给CoCreateInstance:

1. MAPGIS对象的类ID
2. NULL
3. CLSCTX\_INPROC\_SERVER
4. MAPGIS对象的接口ID
5. MAPGIS“对象指针”变量的地址(使用&运算符的地址)。

例如,要建立一个新的IPntArea对象:

```
IPntArea *ptPAi;
```

```
hr=CoCreateInstance(CLSID_PntArea,
                    NULL,
                    CLSCTX_INPROC_SERVER,
                    IID_IPntArea,
                    (void**)&ptPAi);
```

## 2.6、声明和分配对象变量

为了存放从另一个对象返回的MAPGIS对象，必须声明合适的MAPGIS对象变量。

Visual Basic

首先声明特定 MAPGIS 对象类型的对象变量；

```
Dim l3dotset As L_3DOTSet
```

```
Dim l3dot2 As L_3DOT
```

```
Set l3dotset = New L_3DOTSet
```

然后把返回的MAPGIS对象(从集合、对象属性或方法返回值)分配给声明的对象变量。

例如：

```
l3dotset.Append 349.3345, 46945, -231.2355666
```

```
Set l3dot2 = l3dotset(0) '从集合中得到
```

注意可以把返回的 MAPGIS 对象赋给使用 New 关键字声明的对象变量。在把一个新对象赋予该变量之前，Visual Basic 自动释放由 New 关键字创建的对象。但是，如果声明一个只是存放返回的 MAPGIS 对象的对象变量，不要使用 New 关键字，不会产生建立和释放对象的开销。

C++

首先声明合适的MAPGIS “对象指针”变量。

```
ID_Dot *ptDot;
```

```
ID_Dot *ptDot2;
```

然后通过指针调用QueryInterface方法获得新接口指针。

例如：

```
hr = CoCreateInstance(CLSID_D_Dot,
```

```
NULL,
```

```
CLSCTX_INPROC_SERVER,
```

```
IID_ID_Dot,
```

```
(void**)&ptDot);
```

```
hr = ptDot -> QueryInterface(IID_ID_Dot, (void**)&ptDot2);
```

此时指针ptDot和ptDot2指向相同的 MAPGIS 对象。

## 2.7、使用属性

属性是存放在MAPGIS对象中的数据项。可以获得几乎全部属性值，但只能设置(改变)一定的属性值。

### 2.7.1、获得属性

Visual Basic

要得到一个属性值，使用以下语法：

[Set] variable=Object.Property

例如，要得到一个double属性：

```
Dim dbdot1 As double
Dim dbdot2 As double
dbdot1 = d_dot.x
dbdot2 = d_dot.y
```

要得到一个对象的属性：

```
Dim dots As new D_DotSet
Dim dot As D_Dot
```

.....

```
Set dot = dots.Item(0)
```

.....

C++

对于名字为 Property 的属性，调用 get\_Property()方法得到属性值。因为所有 MAPGIS 函数都返回一个 HRESULT 值，必须给 get\_Property()函数传递合适类型的变量地址（使用&运算符）。get\_Property()方法将当前属性值写到程序变量。语法：

```
pMAPGISObject -> get_Property(&variable);
```

例如：

```
double dbx ,dby ;
hr = ptDot -> get_x(&dby);
hr = ptDot -> get_y(&dbx);
```

### 2.7.2、设置属性

只能设置可读写的 MAPGIS 属性。

Visual Basic

使用下列语法设置属性值：

Object.property = Value

例如：设置一个 double 属性：

```
Dim DOT As D_DOT
```

```
Set DOT = New D_DOT
DOT.x = 1323.323
DOT.y = -0.99999
DOT.x = 2.000000
```

## C++

对于名字为 Property 的属性，使用 put\_Property() 方法把属性值设置成一个新值。把合适类型的程序变量传递给 put\_Property()方法，该方法从程序变量中读出新值，并把对象属性设置成这个值。使用以下语法：

```
pMAPGISObject -> put_Property(variable);
```

例如：

```
hr = ptDot -> put_x(dbx);
```

```
hr = ptDot -> put_y(dby);
```

## 2.8、调用方法

方法是使用 MAPGIS 对象执行一个动作的函数。

## Visual Basic

使用以下语法调用不带参数的方法：

```
[return = ] Object.Method
```

使用以下语法调用带参数但不返回值的方法，或忽略任何返回值；

```
Object.Method[Name1: = ] param1,[Name2: = ]param2,...
```

使用以下语法调用带参数柄返回一个值的方法：

```
return = Object.Method([Name1: = ]param1,[Name2: =:]param2,...)
```

Name 项是可选的参数名。如果包括参数名，可以按任何顺序指定方法参数。如果不使用参数名，必须按要求的顺序指定方法参数。

例如：

```
Dim dot3set As D_3DOTSet
```

```
Dim dot32 As D_3DOT
```

```
Set dot3set = New D_3DOTSet
```

```
dot3set.Append 3.23555, 2.3466, 9.49695
```

```
Set dot32 = dot3set(0)
```

```
dot3set.Append2 dot32
```

```
.....
```

C++

使用以下语法调用不带参数的方法：

```
pMAPGISObject -> Method();
```

使用以下语法调用带参数（但无返回参数）的方法：

```
pMAPGISObject -> Method(param1,param2,...);
```

如果方法有返回参数，必须为返回参数传递合适类型的程序变量的地址（使用&运算符）。方法将把返回值写到程序变量。

使用以下语法调用带单个返回参数的方法：

```
pMAPGISObject -> Method(&return);
```

使用以下语法调用带要求输入参数，返回参数和可选的输入参数：

```
pMAPGISObject -> Method(req_param1,&return,...,opt_param2,...);
```

例如：

```
hr = ptDotSet -> Append(dbx,dbx);
```

```
hr = ptDotSet -> Append2(ptDot);
```

```
hr = ptDotSet -> get_count(&lCount);
```

```
hr = ptDotSet -> get_item (1,&ptDot);
```

## 2.9、使用集合

一个集合含有一组对象，这些对象有相同的类型。集合的名字是它含有的对象类型的名字的复数形式。例如，D\_3DOTSet 集合中含有一组 D\_3DOT 对象。

集合类型对象说明：

**集合类型对象一般带 Set 标识（如：D\_DotSet、LinSet），具备下列部分或全部常规方法和属性：**

Append	--	添加一个新元素；
Insert	--	在指定位置插入新元素；
Update	--	更新集合中的元素；
Remove	--	删除指定位置的元素；
RemoveAll	--	删除集合中的所有元素；
count	--	集合中的元素总数（只读）。
item	--	集合中的第 index 项元素（缺省属性、只读），index 从 0 开始；

**注意：**只能对 item 进行读操作，即只能从 item 属性获得相应的值，而不能对 item 进行赋值操作。（记录集对象 RecordSet 的 Record 属性和 item 类似）

使用 item 属性示例：

```
Dim dots As D_DotSet
```

```
Dim dot As D_Dot
```

```
Set dots = New D_DotSet
dots.Append 11, 11
dots.Append 22, 22
dots.Append 33, 33
```

```
Set dot = dots.Item(0)
dot.x = 111
dot.y = 111
dots.Update2 0, dot
```

```
Set dot = Nothing
Set dots = Nothing
```

以下操作是非法的：

```
dots.Item(1).x = 222
dots.Item(1).y = 222
set dots.item(1) = dot
```

### 2.9.1 获取集合对象

集合实际上是含有集合中所有特定对象的集合对象。与所有对象一样，集合对象也有属性和方法。

Visual Basic

使用以下语法引用集合对象：

```
Dim dot3set As D_3DOTSet
Set dot3set = New D_3DOTSet
```

C++

和其他类型对象一样创建和获取，例如：

```
hr = CoCreateInstance(CLSID_D_DotSet,
    NULL,
    CLSCTX_INPROC_SERVER,
    IID_ID_DotSet,
    (void**)&ptDotSet);
```

### 2.9.2 从集合获取特定对象

在使用一个特定对象之前，必须从集合中取出它。

#### Visual Basic

因为 item 是集合对象的默认属性，可以在集合对象后的圆括号中指定集合内的序数值（从 0 开始）得到一个特定对象：

```
Object.Item( lOrdinal )
```

例如：

```
Dim dot3set As D_3DOTSet
Set dot3set = New D_3DOTSet
```

```
Dim dot32 As D_3DOT
```

```
.....
```

```
Set dot32 = dot3set( lItem )
```

#### C++

在有指向一个集合对象的指针时，可以使用 GetItem 函数，得到指向该集合中的特定对象的指针。语法如下：

```
pMAPGISObject -> get_item ( index, /* [retval][out] */ I_INTERFACE__RPC_FAR *__RPC_FAR *pVal);
```

例如：

```
hr = ptDotSet -> get_item (1,&ptDot);
```

## 2.9.3 遍历集合

对集合中的每个特定对象执行一系列的操作常常是很有用的。

#### Visual Basic

使用 Visual Basic 的 for 循环语句遍历集合中的所有特定对象。例如以下循环遍历 l3dotset 集合的对象，依次将值赋给 l3dot2 对象：

```
For lCount = 0 To l3dotset.Count -1 Step 1
```

```
    Set l3dot2 = l3dotset(iCount)
```

```
    .....
```

```
Set l3dot2=Nothing
```

```
Next
```

#### C++

可以使用集合对象的 GetCount 函数和 GetItem 函数遍历集合中的所有特定对象。对 GetCount 使用以下语法：



---

```
pMAPGISObject -> get_item( long index, /* [retval][out] */ I_INTERFACE _RPC_FAR
*_RPC_FAR *pVal);
```

例如：

```
hr = ptDotSet -> get_count(&lCount);
for(i=0;i<lCount;i++)
{
    hr = ptDotSet -> get_item (1,&ptDot);
    //use the specific ptDot object
    .....
    ptDot->Release();
}
```

### 2.9.4 给集合增加新对象

使用下列步骤把一个新的 MAPGIS 对象加到一个集合中：

- 1．建立 MAPGIS 对象的新实例；
- 2．设置新 MAPGIS 对象的属性；
- 3．把 MAPGIS 对象加到一个集合中。

#### Visual Basic

在用 Dim 语句的 New 关键词或 CreateObject 函数建立新的 MAPGIS 对象后,设置新对象的合适属性,然后使用以下语法把新对象加到合适的集合中：

```
Object.Append NewObject
```

例如：

```
Dim ldot As L_DOT
Set ldot = CreateObject("MapGisBasCom1.L_DOT")
.....
ldotset.Append2 ldot
```

#### C++

可以用以下语法把新对象加到集合中去：

```
pMAPGISObject -> Append2(pNewObject);
```

例如：

```
hr = ptDotSet -> Append2(ptDot);
```

### 2.9.5 从集合中删除已有对象

从集合中删除一个已经存在的 MAPGIS 对象就是撤消该 MAPGIS 对象。

#### Visual Basic

MAPGIS 对象有两个方法删除已有对象，使用以下语法：

Object.Remove(index As Long, nCount As Long)

从 index 位置开始删除 nCount 个实体

例如：

ldotset.Remove 1, 2

也可以使用以下语法删除集合中的所有实体：

Object.RemoveAll

#### C++

可以使用下列语法按序号从集合删除一个对象：

```
pMAPGISObject -> Remove(  
    /* [in] */ long index,  
    /* [in] */ long nCount);
```

例如，从集合中第一个对象开始，删除 2 个对象：

```
hr = ptDotSet -> Remove(0,2);
```

也可以使用以下语法从集合中删除所有对象：

```
pMAPGISObject -> RemoveAll();
```

## 2.9.6 更新集合

可用指定对象更新集合中的元素。

#### Visual Basic

使用以下语法更新集合：

Object.Update2(index As Long, newObject)

用 newObject 对象更新集合中的第 index 项数据。

例如：

ldotset.Update2(2, ldot)

#### C++

使用以下语法更新集合：

```
pMAPGISObject -> Update2(  
    /* [in] */ long index,  
    /* [in] */ *ptNewVal);
```

例如：

```
hr = ptDotSet -> Update2(1,ptDot);
```

## 2.10、释放 MAPGIS 对象

当程序完成使用它创建的 MAPGIS 对象时，必须释放该对象。在释放对 MAPGIS 对象的最后一次应用后，对象变量变成无效的了，不能再使用。要再次使用这个对象，必须得到对它的新引用。

Visual Basic

要释放使用 Dim 语句的 New 关键字或 CreateObject 函数创建的对象，把该对象变量设置成 Nothing。

例如：

```
Set oMapgis = Nothing
```

当对象变量超出作用域时，Visual Basic 自动释放该对象。

C++

使用 Release 函数释放使用 CoCreateInstance 函数创建的对象。

例如：

```
pMAPGISObject -> Release();
```

## 2.11、解除 COM 库

在程序退出之前，COM 库必须被解除初始化。

Visual Basic

MAPGIS 程序不必解除 COM 的初始化。在程序退出时，Visual Basic 自动解除 COM 的初始化。

C++

在完成 MAPGIS 或 COM 函数的全部使用后，程序必须调用 CoUninitialize 函数解除对 COM 库的初始化。

例如：

```
CoUninitialize();
```

## 2.12、Visual Basic 对象

说明：以下对象名以斜体加粗表示该对象在 Visual Basic 中不可创建。

<i>Analysis</i>	Arc_Alloc_Info	Ask_Db_Rcd
D_3Dot	D_3DotSet	D_Dot
D_DotSet	D_Rect	D_RectSet
Date_Stru	<i>ExpObj</i>	Ext_Link_Db_Info

<b>ExtDbTool</b>	Field	Field_ExtHead
Field_Head	GeoLin	GeoLin3D
GeoLin3Dset	GeoLinSet	GeoNet
GeoNetSet	GeoPath	GeoPathSet
GeoPnt	GeoPntSet	GeoReg
GeoRegSet	GisEnv	IDList
L_3Dot	L_3DotSet	L_Dot
L_DotSet	<b>Layer</b>	LayerOnOffPad
LayerPad	Lin	Lin_Info
Lin_Pad	Lin_Top	Lin3D
Lin3Dset	LinArea	<b>LinAtt</b>
<b>LinData</b>	LinSet	LONGList
Map_Para	MapGis	Net
Net_Info	Net_Pad	NetArea
<b>NetAtt</b>	NetSet	Nod_Barrier_Info
Nod_Center_Info	Nod_Stop_Info	<b>NodAtt</b>
<b>NodData</b>	Path	<b>PathData</b>
PathSet	Pnt	Pnt_Info
<b>Pnt_Info_Arc</b>	<b>Pnt_Info_Cir</b>	<b>Pnt_Info_Image</b>
<b>Pnt_Info_Note</b>	<b>Pnt_Info_Sub</b>	<b>Pnt_Info_Text</b>
Pnt_Pad	<b>Pnt_Pad_Arc</b>	<b>Pnt_Pad_Cir</b>
<b>Pnt_Pad_Image</b>	<b>Pnt_Pad_Note</b>	<b>Pnt_Pad_Sub</b>
<b>Pnt_Pad_Text</b>	PntArea	<b>PntAtt</b>
PntSet	Query	Rad
RadSet	Record	Record_Head
RecordSet	Reg	Reg_Info
Reg_Pad	RegArea	<b>RegAtt</b>
RegSet	RepFldFlag	TblArea
Tic_Dot	Tic_DotSet	Time_Stru
TimeStamp_Stru	TraceDEM	Tran_Par
<b>VectWorkArea</b>	Win_Org	<b>WorkArea</b>

## 2.13、C++对象

Analysis	Arc_Alloc_Info	Ask_Db_Rcd
CLSID_Analysis	CLSID_Arc_Alloc_Info	CLSID_Ask_Db_Rcd
IID_IAnalysis	IID_IArc_Alloc_Info	IID_IAsk_Db_Rcd

D_3Dot	D_3DotSet	D_Dot
CLSID_D_3Dot	CLSID_D_3DotSet	CLSID_D_Dot
IID_ID_3Dot	IID_ID_3DotSet	IID_ID_Dot
D_DotSet	D_Rect	D_RectSet
CLSID_D_DotSet	CLSID_D_Rect	CLSID_D_RectSet
IID_ID_DotSet	IID_ID_Rect	IID_ID_RectSet
Date_Stru	IDList	Ext_Link_Db_Info
CLSID_Date_Stru	CLSID_IDList	CLSID_Ext_Link_Db_Info
IID_IDate_Stru	IID_IDList	IID_I Ext_Link_Db_Info
Field_Head	Field	Field_ExtHead
CLSID_Field_Head	CLSID_Field	CLSID_Field_ExtHead
IID_IField_Head	IID_IField	IID_IField_ExtHead
IID_IField_ExtHead	IID_IField_Head	
GeoNet	GeoLin	GeoLin3D
CLSID_GeoNet	CLSID_GeoLin	CLSID_GeoLin3D
IID_IGeoNet	IID_IGeoLin	IID_IGeoLin3D
IID_INet	IID_ILin	IID_ILin3D
IID_IRecord	IID_IRecord	IID_IRecord
GeoLin3Dset	GeoLinSet	GeoPntSet
CLSID_GeoLin3Dset	CLSID_GeoLinSet	CLSID_GeoPntSet
IID_IGeoLin3Dset	IID_IGeoLinSet	IID_IGeoPntSet
GeoNetSet	GeoRegSet	GeoPathSet
CLSID_GeoNetSet	CLSID_GeoRegSet	CLSID_GeoPathSet
IID_IGeoNetSet	IID_IGeoRegSet	IID_IGeoPathSet
GeoPnt	GeoPath	GeoReg
CLSID_GeoPnt	CLSID_GeoPath	CLSID_GeoReg
IID_IGeoPnt	IID_IGeoPath	IID_IGeoReg
IID_IPnt	IID_IPath	IID_IReg
IID_IRecord	IID_IRecord	IID_IRecord
GisEnv	L_DotSet	L_Dot
CLSID_GisEnv	CLSID_L_DotSet	CLSID_L_Dot
IID_IGisEnv	IID_IL_DotSet	IID_IL_Dot
L_3Dot	L_3DotSet	LayerOnOffPad
CLSID_L_3Dot	CLSID_L_3DotSet	CLSID_LayerOnOffPad
IID_IL_3Dot	IID_IL_3DotSet	IID_ILayerOnOffPad

LayerPad	Lin	Lin_Info
CLSID_LayerPad	CLSID_Lin	CLSID_Lin_Info
IID_ILayerPad	IID_ILin	IID_ILin_Info
	IID_ILin_Info	
	IID_ID_DotSet	
Lin_Pad	Lin_Top	Lin3D
CLSID_Lin_Pad	CLSID_Lin_Top	CLSID_Lin3D
IID_ILin_Pad	IID_ILin_Top	IID_ILin3D
		IID_ILin_Info
		IID_ID_3DotSet
Lin3Dset	LinSet	LONGList
CLSID_Lin3Dset	CLSID_LinSet	CLSID_LONGList
IID_ILin3Dset	IID_ILinSet	IID_ILONGList
Map_Para	MapGis	Net
CLSID_Map_Para	CLSID_MapGis	CLSID_Net
IID_IMap_Para	IID_IMapGis	IID_INet
		IID_INet_Info
LinArea	RegArea	NetArea
CLSID_LinArea	CLSID_RegArea	CLSID_NetArea
IID_IlinArea	IID_IRegArea	IID_INetArea
IID_IWorkArea	IID_IWorkArea	IID_IWorkArea
IID_IVectWorkArea	IID_IVectWorkArea	IID_IVectWorkArea
IID_IExtDbTool	IID_IExtDbTool	IID_IExtDbTool
IID_ILinAtt	IID_IRegAtt	IID_INetAtt
IID_ILayer	IID_ILinAtt	IID_ILinAtt
	IID_INodAtt	IID_INodAtt
	IID_ILinData	IID_ILinData
	IID_INodData	IID_INodData
	IID_ILayer	IID_ILayer
		IID_IPathData
Net_Pad	NetSet	Nod_Barrier_Info
CLSID_Net_Pad	CLSID_NetSet	CLSID_Nod_Barrier_Info
IID_INet_Pad	IID_INetSet	IID_INod_Barrier_Info
Nod_Center_Info	Nod_Stop_Info	Path
CLSID_Nod_Center_Info	CLSID_Nod_Stop_Info	CLSID_Path
IID_I Nod_Center_Info	IID_INod_Stop_Info	IID_IPath

PathSet	Pnt	Net_Info
CLSID_PathSet	CLSID_Pnt	CLSID_Net_Info
IID_IPathSet	IID_IPnt	IID_INet_Info
	IID_IPnt_Info	
PntSet	Query	Rad
CLSID_PntSet	CLSID_Query	CLSID_Rad
IID_IPntSet	IID_IQuery	IID_IRad
Reg_Pad	Tic_Dot	TraceDEM
CLSID_Reg_Pad	CLSID_Tic_Dot	CLSID_TraceDEM
IID_IReg_Pad	IID_ITic_Dot	IID_ITraceDEM
Pnt_Pad	Pnt_Info	PntArea
CLSID_Pnt_Pad	CLSID_Pnt_Info	CLSID_PntArea
IID_IPnt_Pad	IID_IPnt_Info	IID_IPntArea
IID_IPnt_Pad_Arc	IID_IPnt_Info_Arc	IID_IWorkArea
IID_IPnt_Pad_Cir	IID_IPnt_Info_Cir	IID_IVectWorkArea
IID_IPnt_Pad_Image	IID_IPnt_Info_Image	IID_IExtDbTool
IID_IPnt_Pad_Note	IID_IPnt_Info_Note	IID_IPntAtt
IID_IPnt_Pad_Sub	IID_IPnt_Info_Sub	IID_ILayer
IID_IPnt_Pad_Text	IID_IPnt_Info_Text	
RadSet	Record	Record_Head
CLSID_RadSet	CLSID_Record	CLSID_Record_Head
IID_IRadSet	IID_IRecord	IID_IRecord_Head
	IID_IRecord_Head	IID_IField_Head
RecordSet	Reg	TblArea
CLSID_RecordSet	CLSID_Reg	CLSID_TblArea
IID_IRecordSet	IID_IReg	IID_ITblArea
IID_IRecord_Head	IID_IReg_Info	IID_IWorkArea
		IID_IExtDbTool
		IID_IRecord_Head
RegSet	RepFldFlag	Reg_Info
CLSID_RegSet	CLSID_RepFldFlag	CLSID_Reg_Info
IID_IRegSet	IID_IRepFldFlag	IID_IReg_Info
Tran_Par	Tic_DotSet	Time_Stru
CLSID_Tran_Par	CLSID_Tic_DotSet	CLSID_Time_Stru
IID_ITran_Par	IID_ITic_DotSet	IID_ITime_Stru

---

TimeStamp_Stru	Win_Org
CLSID_TimeStamp_Stru	CLSID_Win_Org
IID_ITimeStamp_Stru	IID_IWin_Org

---

## 2.14、MAPGIS 组件使用示例代码（VB）

安装 MAPGIS 平台及 MAPGIS 二次开发包后，在 MAPGIS 的 VBDemo 目录下安装了组件使用示例代码。本书的后继部分，都假设 MAPGIS 安装在 D:\mapgis61 目录下，则在 D:\mapgis61\VBDemo\ 目录下可以找到一系列的组件使用示例代码。



## 第三章 MAPGIS 对象说明

本章的组件使用示例代码包含在 D:\MAPGIS61\VBDemo\BasCom1 目录下。包含基本数据对象的使用示例 (BasObj)、属性数据使用示例 (BasAtt)、工作区对象使用示例 (BasWorkArea)、分析对象使用示例 (Analysis) 和查询对象使用示例 (Query)。

### 3.1、D\_Dot 对象

描述 D\_Dot 对象是二维双精度型坐标点。

#### 3.1.1、属性：

属性	描述
x	二维坐标点 x 坐标
Double	
可读写	
y	二维坐标点 y 坐标
Double	
可读写	

#### 3.1.2、方法：无

### 3.2、D\_DotSet 对象

描述 D\_DotSet 对象是二维双精度型坐标点集合。

#### 3.2.1、属性：

属性	描述
count	二维双精度型坐标点集当前坐标点数
Long	
只读	
item(index)	第 index 个 D_Dot 点，index 为 Long 型
D_Dot	
只读	

### 3.2.2、方法：

#### Append 方法

描述 添加一个二维坐标点。

Visual Basic D\_DotSet.Append x, y

C++ HRESULT pD\_DotSet->Append(double x, double y);

参数说明：

参数	类型	出口/入口	描述
x	Double	[in]	添加的二维坐标点 x 坐标
y	Double	[in]	添加的二维坐标点 y 坐标

#### Append2 方法

描述 添加一个 D\_Dot 点。

Visual Basic D\_DotSet.Append2 Dot

C++ HRESULT pD\_DotSet->Append2(ID\_Dot \*pIDot);

参数说明：

参数	类型	出口/入口	描述
Dot	D_Dot	[in]	添加的 D_Dot 点

#### Insert 方法

描述 在指定位置插入一个二维坐标点。

如果指定的位置>=集合中的点数，则添加一个二维坐标点。

Visual Basic D\_DotSet.Insert index, x, y

C++ HRESULT pD\_DotSet->Insert(long index, double x, double y);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
x	Double	[in]	插入的二维坐标点 x 坐标
y	Double	[in]	插入的二维坐标点 y 坐标

#### Insert2 方法

描述 在指定位置插入一个 D\_Dot 点。

如果指定的位置>=集合中的点数，则添加一个 D\_Dot 点。

Visual Basic D\_DotSet.Insert2 index, Dot

C++ HRESULT pD\_DotSet->Insert2(long index, ID\_Dot \*pIDot);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

index	Long	[in]	集合的索引号
Dot	D_Dot	[in]	插入的 D_Dot 点

**Remove 方法**

描述 删除集合中从 index 起 nCount 个 D\_Dot 点

Visual Basic D\_DotSet.Remove index, nCount

C++ HRESULT pD\_DotSet-> Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
nCount	Long	[in]	删除的个数

**RemoveAll 方法**

描述 删除所有 D\_Dot 点

Visual Basic D\_DotSet.RemoveAll

C++ HRESULT pD\_DotSet-> RemoveAll();

参数说明： 无

**Set 方法**

描述 以传入的点集合替代原集合

Visual Basic rtl = D\_DotSet.Set(ptNewVal)

C++ HRESULT pD\_DotSet->Set(ID\_DotSet \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	D_DotSet	[in]	D_DotSet 对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Update 方法**

描述 更新第 index 项数据

Visual Basic D\_DotSet.Update index, x, y

C++ HRESULT pD\_DotSet->Update(long index, double x, double y);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
x	Double	[in]	更新的二维坐标点 x 坐标
y	Double	[in]	更新的二维坐标点 y 坐标

**Update2 方法**

描述 更新第 index 项数据

Visual Basic D\_DotSet.Update2 index, ptNewVal

C++ HRESULT pD\_DotSet->Update2(long index, ID\_Dot \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
ptNewVal	D_Dot	[in]	更新的 D_Dot 点

**3.3、L\_Dot 对象**

描述 L\_Dot 对象是二维长整型坐标点。

**3.3.1、属性：**

属性	描述
x	二维坐标点 x 坐标
Long	
可读写	
y	二维坐标点 y 坐标
Long	
可读写	

**3.3.2、方法：无****3.4、L\_DotSet 对象**

描述 L\_DotSet 对象是二维长整型坐标点集合。

**3.4.1、属性：**

属性	描述
count	二维长整型坐标点集当前坐标点数
Long	
只读	

item(index)	第 index 个 L_Dot 点, index 为 Long 型
L_Dot	
只读	

### 3.4.2、方法：

#### Append 方法

描述 添加一个二维长整型坐标点。

Visual Basic L\_DotSet.Append x, y

C++ HRESULT pL\_DotSet->Append(long x, long y);

参数说明：

参数	类型	出口/入口	描述
x	Long	[in]	添加的二维坐标点 x 坐标
y	Long	[in]	添加的二维坐标点 y 坐标

#### Append2 方法

描述 添加一个 L\_Dot 点。

Visual Basic L\_DotSet.Append2 Dot

C++ HRESULT pD\_DotSet->Append2(IL\_Dot \*pIDot);

参数说明：

参数	类型	出口/入口	描述
Dot	L_Dot	[in]	添加的 L_Dot 点

#### Insert 方法

描述 在指定位置插入一个二维长整型坐标点。

如果指定的位置>=集合中的结点数，则添加一个二维长整型坐标点。

Visual Basic L\_DotSet.Insert index, x, y

C++ HRESULT pL\_DotSet->Insert(long index, long x, long y);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
x	Long	[in]	插入的二维坐标点 x 坐标
y	Long	[in]	插入的二维坐标点 y 坐标

#### Insert2 方法

描述 在指定位置插入一个 L\_Dot 点。

如果指定的位置>=集合中的结点数，则添加一个 L\_Dot 点。

Visual Basic L\_DotSet.Insert2 index, Dot

C++ HRESULT pL\_DotSet->Insert2(long index, IL\_Dot \*pIDot);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
Dot	L_Dot	[in]	插入的 L_Dot 点

### Remove 方法

描述 删除集合中从 index 起 nCount 个 L\_Dot 点。

Visual Basic L\_DotSet.Remove index, nCount

C++ HRESULT pL\_DotSet-> Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
nCount	Long	[in]	删除的个数

### RemoveAll 方法

描述 删除所有 L\_Dot 点。

Visual Basic L\_DotSet.RemoveAll

C++ HRESULT pL\_DotSet-> RemoveAll();

参数说明： 无

### Set 方法

描述 以传入的点集合替代原集合。

Visual Basic rtl = L\_DotSet.Set(ptNewVal)

C++ HRESULT pL\_DotSet->Set(IL\_DotSet \*ptNewVal,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	L_DotSet	[in]	L_DotSet 对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### Update 方法

描述 更新第 index 项数据。

Visual Basic L\_DotSet.Update index, x, y

C++ HRESULT pL\_DotSet->Update(long index, long x, long y);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号

x	Long	[in]	二维坐标点 x 坐标
y	Long	[in]	二维坐标点 y 坐标

### Update2 方法

描述 更新第 index 项数据。

Visual Basic L\_DotSet.Update2 index, ptNewVal

C++ HRESULT pL\_DotSet-> Update2(long index, IL\_Dot \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
ptNewVal	L_Dot	[in]	L_Dot 点

## 3.5、D\_3Dot 对象

描述 D\_3Dot 对象是三维双精度型坐标点。

### 3.5.1、属性：

属性	描述
x	三维坐标点 x 坐标
Double 可读写	
y	三维坐标点 y 坐标
Double 可读写	
z	三维坐标点 z 坐标
Double 可读写	

### 3.5.2、方法：无

## 3.6、D\_3DotSet 对象

描述 D\_DotSet 对象是三维双精度型坐标点集合。

### 3.6.1、属性：

属性	描述
count	三维双精度型坐标点集当前坐标点数
Long 只读	
item(index)	第 index 个 D_3Dot 点, index 为 Long 型
D_3Dot 只读	

### 3.6.2、方法：

#### Append 方法

描述 添加一个三维坐标点。

Visual Basic D\_3DotSet.Append x, y, z

C++ HRESULT pD\_3DotSet->Append(double x, double y, double z);

参数说明：

参数	类型	出口/入口	描述
x	Double	[in]	三维坐标点 x 坐标
y	Double	[in]	三维坐标点 y 坐标
z	Double	[in]	三维坐标点 z 坐标

#### Append2 方法

描述 添加一个 D\_3Dot 点。

Visual Basic D\_3DotSet.Append2 Dot

C++ HRESULT pD\_3DotSet-> Append2(ID\_3Dot \*pIDot);

参数说明：

参数	类型	出口/入口	描述
Dot	D_3Dot	[in]	添加的 D_3Dot 点

#### Insert 方法

描述 在指定位置插入一个三维坐标点。

如果指定的位置>=集合中的点数，则添加一个三维坐标点。

Visual Basic D\_3DotSet.Insert index, x, y, z

C++ HRESULT pD\_3DotSet-> Insert(long index, double x, double y double z);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
x	Double	[in]	三维坐标点 x 坐标
y	Double	[in]	三维坐标点 y 坐标



z	Double	[in]	三维坐标点 y 坐标
---	--------	------	------------

### Insert2 方法

**描述** 在指定位置插入一个 D\_3Dot 点。  
如果指定的位置>=集合中的点数，则添加一个 D\_3Dot 点。

Visual Basic D\_3DotSet.Insert2 index, Dot

C++ HRESULT pD\_3DotSet->Insert2(long index, ID\_3Dot \*pIDot);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
Dot	D_3Dot	[in]	插入的 D_3Dot 点

### Remove 方法

**描述** 删除集合中从 index 起 nCount 个 D\_3Dot 点。

Visual Basic D\_3DotSet.Remove index, nCount

C++ HRESULT pD\_3DotSet-> Remove(long index, long nCount);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
nCount	Long	[in]	删除的个数

### RemoveAll 方法

**描述** 删除所有 D\_3Dot 点。

Visual Basic D\_3DotSet.RemoveAll

C++ HRESULT pD\_3DotSet-> RemoveAll();

**参数说明：** 无

### Set 方法

**描述** 以传入的点集合替代原集合。

Visual Basic rtl = D\_3DotSet.Set(ptNewVal)

C++ HRESULT pD\_3DotSet->Set(ID\_3DotSet \*ptNewVal,VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
ptNewVal	D_3DotSet	[in]	D_3DotSet 对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Update 方法**

描述 更新第 index 项数据。

Visual Basic D\_3DotSet.Update index, x, y, z

C++ HRESULT pD\_3DotSet->Update(long index, double x, double y, double z);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
x	Double	[in]	三维坐标点 x 坐标
y	Double	[in]	三维坐标点 y 坐标
z	Double	[in]	三维坐标点 z 坐标

**Update2 方法**

描述 更新第 index 项数据。

Visual Basic D\_3DotSet.Update2 index, ptNewVal

C++ HRESULT pD\_3DotSet-> Update2(long index, ID\_3Dot \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
ptNewVal	D_3Dot	[in]	D_3Dot 点

**3.7、L\_3Dot 对象**

描述 L\_3Dot 对象是三维长整型坐标点。

**3.7.1、属性：**

属性	描述
x	三维坐标点 x 坐标
Long 可读写	
y	三维坐标点 y 坐标
Long 可读写	
z	三维坐标点 z 坐标
Long 可读写	

### 3.7.2、方法：无

## 3.8、L\_3DotSet 对象

描述 L\_3DotSet 对象是三维长整型坐标点集合。

### 3.8.1、属性：

属性	描述
count Long 只读	三维长整型坐标点集当前坐标点数
item(index) L_3Dot 只读	第 index 个 L_3Dot 点，index 为 Long 型

### 3.8.2、方法：

#### Append 方法

描述 添加一个三维长整型坐标点。

Visual Basic L\_3DotSet.Append x, y, z

C++ HRESULT pL\_3DotSet->Append(long x, long y, long z);

参数说明：

参数	类型	出口/入口	描述
x	Long	[in]	三维坐标点 x 坐标
y	Long	[in]	三维坐标点 y 坐标
z	Long	[in]	三维坐标点 z 坐标

#### Append2 方法

描述 添加一个 L\_3Dot 点。

Visual Basic L\_3DotSet.Append2 Dot

C++ HRESULT pL\_3DotSet->Append2(IL\_3Dot \*pIDot);

参数说明：

参数	类型	出口/入口	描述
Dot	L_3Dot	[in]	添加的 L_3Dot 点

**Insert 方法**

**描述** 在指定位置插入一个三维长整型坐标点。  
如果指定的位置 $\geq$ 集合中的结点数，则添加一个三维长整型坐标点。

Visual Basic L\_3DotSet.Insert index, x, y, z

C++ HRESULT pL\_3DotSet-> Insert(long index, long x, long y long z);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
x	Long	[in]	三维坐标点 x 坐标
y	Long	[in]	三维坐标点 y 坐标
z	Long	[in]	三维坐标点 z 坐标

**Insert2 方法**

**描述** 在指定位置插入一个 L\_3Dot 点。  
如果指定的位置 $\geq$ 集合中的结点数，则添加一个 L\_3Dot 点。

Visual Basic L\_3DotSet.Insert2 index, Dot

C++ HRESULT pL\_3DotSet->Insert2(long index, IL\_3Dot \*pIDot);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
Dot	L_3Dot	[in]	插入的 L_3Dot 点

**Remove 方法**

**描述** 删除集合中从 index 起 nCount 个 L\_3Dot 点。

Visual Basic L\_3DotSet.Remove index, nCount

C++ HRESULT pL\_3DotSet-> Remove(long index, long nCount);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
nCount	Long	[in]	删除的个数

**RemoveAll 方法**

**描述** 删除所有 L\_3Dot 点。

Visual Basic L\_3DotSet.RemoveAll

C++ HRESULT pL\_3DotSet-> RemoveAll();

**参数说明：** 无

**Set 方法**

描述 以传入的点集合替代原集合。

Visual Basic rtl = L\_3DotSet.Set(ptNewVal)

C++ HRESULT pL\_3DotSet->Set(IL\_3DotSet \*ptNewVal,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	L_3DotSet	[in]	L_3DotSet 对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Update 方法**

描述 更新第 index 项数据。

Visual Basic L\_3DotSet.Update index, x, y, z

C++ HRESULT pL\_3DotSet->Update(long index, long x, long y, long z);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
x	Long	[in]	三维坐标点 x 坐标
y	Long	[in]	三维坐标点 y 坐标
z	Long	[in]	三维坐标点 z 坐标

**Update2 方法**

描述 更新第 index 项数据。

Visual Basic L\_3DotSet.Update2 index, ptNewVal

C++ HRESULT pL\_3DotSet-> Update2(long index, IL\_3Dot \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
ptNewVal	L_3Dot	[in]	L_3Dot 点

**3.9、D\_Rect 对象**

描述 D\_Rect 对象是双精度型矩形框。

**3.9.1、属性：**

属性	描述
----	----

xmax	最大 x
Double	
可读写	
xmin	最小 x
Double	
可读写	
ymax	最大 y
Double	
可读写	
ymin	最小 y
Double	
可读写	

### 3.9.2、方法：无

## 3.10、D\_RectSet 对象

描述 D\_RectSet 对象是矩形框集合。

### 3.10.1、属性：

属性	描述
count	矩形集合当前所包含的矩形框个数
Long	
只读	
item(index)	第 index 个 D_Rect 矩形，index 为 Long 型
D_Rect	
只读	

### 3.10.2、方法：

#### Append 方法

描述 往矩形集合中添加一个矩形。

Visual Basic D\_RectSet.Append rc

C++ HRESULT pD\_RectSet->Append(ID\_Rect \*rc);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

rc	D_Rect	[in]	添加的矩形
----	--------	------	-------

**Insert 方法**

**描述** 往矩形集合给定位置插入一个 D\_Rect 矩形。  
如果指定的位置 $\geq$ 集合中的结点数，则添加一个 D\_Rect 矩形。

Visual Basic D\_RectSet.Insert index, rc

C++ HRESULT pD\_RectSet->Insert(long index, ID\_Rect \*rc);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
rc	D_Rect	[in]	插入的矩形

**Remove 方法**

**描述** 删除集合中从 index 起 nCount 个 D\_Rect 矩形。

Visual Basic D\_RectSet.Remove index, nCount

C++ HRESULT pD\_RectSet-> Remove(long index, long nCount);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
nCount	Long	[in]	删除的 D_Rect 矩形数

**RemoveAll 方法**

**描述** 删除集合中所有的 D\_Rect 矩形。

Visual Basic D\_RectSet.RemoveAll

C++ HRESULT pD\_RectSet-> RemoveAll();

**参数说明：** 无

**Update 方法**

**描述** 更新集合中的 D\_Rect 矩形。

Visual Basic D\_RectSet.Update index, ptNewVal

C++ HRESULT pD\_RectSet->Update(long index, ID\_Rect \*ptNewVal);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
ptNewVal	D_Rect	[in]	D_Rect 矩形对象

### 3.11、Win\_Org 对象

描述 Win\_Org 对象封装了对窗口参数的操作。

#### 3.11.1、属性：

属性	描述
frc	显示范围
D_Rect 可读写	
r	显示比例
Double 可读写	
x0	视口左下角 x 坐标
Double 可读写	
y0	视口左下角 y 坐标
Double 可读写	

#### 3.11.2、方法：无

### 3.12、Date\_Stru 对象

描述 Date\_Stru 对象是日期数据结构。

#### 3.12.1、属性：

属性	描述
day	日
Integer 可读写	
mon	月
Integer 可读写	



year	年
Integer	
可读写	

### 3.12.2、方法：无

## 3.13、Time\_Stru 对象

描述 Time\_Stru 对象是时间数据结构。

### 3.13.1、属性：

属性	描述
hour	时
Integer	
可读写	
min	分
Integer	
可读写	
sec	秒
Double	
可读写	

### 3.13.2、方法：无

## 3.14、TimeStamp\_Stru 对象

描述 Time\_Stru 对象是邮戳型数据结构。

### 3.14.1、属性：

属性	描述
year	年
Integer	
可读写	

month	月
Integer	
可读写	
day	日
Integer	
可读写	
<b>hour</b>	时
Integer	
可读写	
minute	分
Integer	
可读写	
second	秒
Integer	
可读写	
fraction	秒的小数位数值
Long	
可读写	

### 3.14.2、方法：无

## 3.15、Field\_ExtHead 对象

**描述** 字段扩展头对象 Field\_ExtHead 对应于字段的扩展结构，封装了对扩展字段的操作。如：设置字段形态、取和设最大值、最小值、缺省值等。

### 3.15.1、属性：

属性	描述
alias	字段别名
String	
可读写	
IsNull	是否允许为空
Boolean	
可读写	

shape	字段形态（编辑框/组合框/复选框/按钮）
Enum_FieldShape_Type	
可读写	

### 3.15.2、方法：

#### Init 方法

描述 初始化扩展字段。

Visual Basic rtl = Field\_ExtHead.Init(fieldtype, shapeInfoNum)

C++ HRESULT pField\_ExtHead->Init(Enum\_Field\_Type fieldtype,  
short shapeInfoNum, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fieldtype	Enum_Field_Type	[in]	字段类型
shapeInfoNum	Integer	[in]	字段形态信息数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

注： 使用扩展字段，必须首先调用这个方法进行初始化。

#### GetDefVal 方法

描述 取字段缺省值。

Visual Basic rtl = Field\_ExtHead.GetDefVal(val)

C++ HRESULT pField\_ExtHead->GetDefVal(  
VARIANT \*val, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
val	Variant	[out]	字段缺省值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

注： 数值型、日期、时间、邮戳类型字段可有最大值、最小值和缺省值。

#### SetDefVal 方法

描述 设字段缺省值。

Visual Basic rtl = Field\_ExtHead.SetDefVal(val)

C++ HRESULT pField\_ExtHead->SetDefVal(  
VARIANT val, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
val	Variant	[in]	字段缺省值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetMaxVal 方法**

描述 取字段最大值。

Visual Basic rtl = Field\_ExtHead.GetMaxVal(val)

C++ HRESULT pField\_ExtHead->GetMaxVal(  
VARIANT \*val, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
val	Variant	[out]	取得的最大值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SetMaxVal 方法**

描述 设字段最大值。

Visual Basic rtl = Field\_ExtHead.SetMaxVal(val)

C++ HRESULT pField\_ExtHead->SetMaxVal(  
VARIANT val, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
val	Variant	[in]	字段最大值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetMinVal 方法**

描述 取字段最小值。

Visual Basic rtl = Field\_ExtHead.GetMinVal(val)

C++ HRESULT pField\_ExtHead->GetMinVal(VARIANT \*val, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
val	Variant	[out]	取得的最小值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SetMinVal 方法**

描述 设字段最小值。

Visual Basic rtl = Field\_ExtHead.SetMinVal(val)

C++ HRESULT pField\_ExtHead->SetMinVal(VARIANT val, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
val	Variant	[in]	字段最小值

rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE
-----	---------	-------	------------------------

**GetShapeInfo 方法**

描述 取字段形态信息。

Visual Basic rtl = Field\_ExtHead.GetShapeInfo(seqNo, ttlStr, val)

C++ HRESULT pField\_ExtHead->GetShapeInfo(  
short seqNo, BSTR \*ttlStr, VARIANT \*val, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
seqNo	Integer	[in]	信息序列号
ttlStr	String	[out]	形态信息标题
val	Variant	[out]	形态信息值
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

**SetShapeInfo 方法**

描述 设置字段形态信息。

Visual Basic rtl = Field\_ExtHead.SetShapeInfo(seqNo, ttlStr, val)

C++ HRESULT pField\_ExtHead->SetShapeInfo(  
short seqNo, BSTR ttlStr, VARIANT val, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
seqNo	Integer	[in]	信息序列号
ttlStr	String	[in]	形态信息标题
val	Variant	[in]	形态信息值
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

**GetFieldType 方法**

描述 取扩展结构对应的字段类型。

Visual Basic fldType = Field\_ExtHead.GetFieldType()

C++ HRESULT pField\_ExtHead->GetFieldType(Enum\_Field\_Type \*fldType);

参数说明：

参数	类型	出口/入口	描述
fldType	Enum_Field_Type	[out]	返回字段类型

**GetShapeInfoNum 方法**

描述 取字段形态信息数。

Visual Basic num = Field\_ExtHead.GetShapeInfoNum()

C++ HRESULT pField\_ExtHead->GetShapeInfoNum(short \*num);

参数说明：

参数	类型	出口/入口	描述
num	Integer	[out]	返回字段形态信息数

### HasDefVal 方法

描述 判断是否有缺省值。

Visual Basic rtl = Field\_ExtHead.HasDefVal()

C++ HRESULT pField\_ExtHead->HasDefVal(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	有缺省值返回 TRUE，否则返回 FALSE

### HasMaxVal 方法

描述 判断是否有最大值。

Visual Basic rtl = Field\_ExtHead.HasMaxVal()

C++ HRESULT pField\_ExtHead->HasMaxVal(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	有最大值返回 TRUE，否则返回 FALSE

### HasMinVal 方法

描述 判断是否有最小值。

Visual Basic rtl = Field\_ExtHead.HasMinVal()

C++ HRESULT pField\_ExtHead->HasMinVal(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	有最小值返回 TRUE，否则返回 FALSE

## 3.16、Field\_Head 对象

描述 字段信息结构对象 Field\_Head 提供了对字段结构的所有操作，包括设置字段名、修改字段类型、设置可编辑状态等。

### 3.16.1、属性：

属性	描述
----	----

fieldname	字段名称
String	
可读写	
fieldtype	字段类型
Enum_Field_Type	
可读写	
edit_enable	编辑使能标志（0/1/2=不能/能/禁止）
Integer	
可读写	
msk_leng	字段字符长度
Integer	
可读写	
point_leng	小数位数
Integer	
可读写	
ptc_pos	字段序号，依次为 0,1,2...
Integer	
可读写	

### 3.16.2、方法：

#### HasField\_ExtHead 方法

描述 判断是否有扩展属性。

Visual Basic rtl = Field\_Head.HasField\_ExtHead()

C++ HRESULT pField\_Head-> HasField\_ExtHead(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	有扩展字段返回 TRUE ,否则返回 FALSE

#### GetField\_ExtHead 方法

描述 取字段扩展属性。

Visual Basic Set ExtFld = Field\_Head.GetField\_ExtHead()

C++ HRESULT pField\_Head-> GetField\_ExtHead(IField\_ExtHead \*\*pIExtFld);

参数说明：

参数	类型	出口/入口	描述
ExtFld	Field_ExtHead	[out]	返回 Field_ExtHead 对象

**SetField\_ExtHead 方法**

描述 设字段扩展属性。

Visual Basic rtl = Field\_Head.SetField\_ExtHead(ExtFld)

C++ HRESULT pField\_Head->SetField\_ExtHead(  
IField\_ExtHead \*pIExtFld, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ExtFld	Field_ExtHead	[in]	Field_ExtHead 对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DelField\_ExtHead 方法**

描述 删除字段扩展属性。

Visual Basic rtl = Field\_Head.DelField\_ExtHead()

C++ HRESULT pField\_Head->DelField\_ExtHead(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**IsEqual 方法**

描述 判断字段头是否相等。

Visual Basic rtl = Field\_Head.IsEqual(pIFld0)

C++ HRESULT pField\_Head->IsEqual(IField\_Head \*pIFld0, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pIFld0	Field_Head	[in]	比较的字段头对象
rtl	Boolean	[out]	相同返回 TRUE，否则返回 FALSE

**Load 方法**

描述 从 buf 中装入 Field\_Head。

Visual Basic rtl = Field\_Head.Load(buf, nBufSize)

C++ HRESULT pField\_Head->Load(BYTE \*buf, long nBufSize, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
buf	Byte	[in]	缓冲区，大小为 nBufSize
nBufSize	Long	[in]	缓冲区大小
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE



**Save 方法**

描述 保存 Field\_Head 到 buf 中。

Visual Basic nOutSize = Field\_Head.Save(buf, nBufSize)

C++ HRESULT pField\_Head->Save(BYTE \*buf, long nBufSize, long \*nOutSize);

参数说明：

参数	类型	出口/入口	描述
buf	Byte	[in]	缓冲区，大小为 nBufSize
nBufSize	Long	[in]	缓冲区大小
nOutSize	Long	[out]	返回实际数据长度

**Set 方法**

描述 复制字段属性，ptIFld0 为空则清除字段结构。

Visual Basic rtl = Field\_Head.Set(pIFld0)

C++ HRESULT pField\_Head->Set(IField\_Head \*pIFld0, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pIFld0	Field_Head	[in]	复制的字段属性对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Clone 方法**

描述 克隆一个 Field\_Head。

Visual Basic Set FldHd = Field\_Head.Clone()

C++ HRESULT pField\_Head->Clone(IField\_Head \*\*pIFldHd);

参数说明：

参数	类型	出口/入口	描述
FldHd	Field_Head	[out]	克隆的新字段

**CalSize 方法**

描述 计算 FIELD\_HEAD 对象所占的字节数。

Visual Basic FldHeadSize = Field\_Head.CalSize()

C++ HRESULT pField\_Head-> CalSize(long \*FldHeadSize);

参数说明：

参数	类型	出口/入口	描述
FldHeadSize	Long	[out]	对象所占的字节数

### 3.17、Record\_Head 对象

**描述** 属性结构对象 Record\_Head 是字段结构对象 Field\_Head 的集合。通过 Record\_Head 对象可以操作记录结构。Record\_Head 对象通常与记录集对象 RecordSet 或实体属性数据对象 PntAtt、LinAtt、RegAtt、NetAtt、NodAtt 等一起使用。

#### 3.17.1、属性：

属性	描述
fldEntry(i) Field_Head 可读写	字段说明入口，i 为 Integer 型
numbfield Integer 只读	字段数

#### 3.17.2、方法：

##### AllocEmptyField 方法

**描述** 分配空字段。

Visual Basic rtl = Record\_Head.AllocEmptyField(numbfield)

C++ HRESULT pRecord\_Head->AllocEmptyField(  
short numbfield, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
numbfield	Integer	[in]	字段数
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

##### AppendField 方法

**描述** 添加字段头。

Visual Basic rtl = Record\_Head.AppendField(fldEntry)

C++ HRESULT pRecord\_Head->AppendField(  
IField\_Head \*fldEntry, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
fldEntry	Field_Head	[in]	字段头
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

**CalSize 方法**

描述 计算 Record\_Head 对象占用存储空间大小。

Visual Basic rcdSize = Record\_Head.CalSize()

C++ HRESULT pRecord\_Head->CalSize(long \*rcdSize);

参数说明：

参数	类型	出口/入口	描述
rcdSize	Integer	[out]	返回占用存储空间大小

**Clone 方法**

描述 克隆一个 Record\_Head。

Visual Basic Set Stru = Record\_Head.Clone()

C++ HRESULT pRecord\_Head->Clone(IRecord\_Head \*\*pIStru);

参数说明：

参数	类型	出口/入口	描述
Stru	Record_Head	[out]	克隆出的记录头

**DelField 方法**

描述 删除字段。

Visual Basic rtl = Record\_Head.DelField(index)

C++ HRESULT pRecord\_Head->DelField(  
VARIANT index, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
index	Variant	[in]	字段名称或者字段号
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**InsertField 方法**

描述 从 insTo ( 从 0 开始 ) 位置插入字段。

Visual Basic rtl = Record\_Head.InsertField(fldEntry, insTo)

C++ HRESULT pRecord\_Head->InsertField(  
IField\_Head \*fldEntry, short insTo, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fldEntry	Field_Head	[in]	插入的字段属性
insTo	Integer	[in]	插入位置
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

## IsEqual 方法

描述 比较是否相同。

Visual Basic rtl = Record\_Head.IsEqual(pIStu0)

C++ HRESULT pRecord\_Head->IsEqual(IRecord\_Head\*pIStu0,  
VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pIStu0	Record_Head	[in]	比较的属性结构
rtl	Boolean	[out]	相同返回 TRUE 否则返回 FALSE

## Load 方法

描述 从缓冲区中装入 Record\_Head。

Visual Basic rtl = Record\_Head.Load(buf, nBufSize)

C++ HRESULT pRecord\_Head->Load(BYTE \*buf, long nBufSize, VARIANT\_BOOL  
\*rtl);

参数说明：

参数	类型	出口/入口	描述
buf	Byte	[in]	缓冲区
nBufSize	Long	[in]	缓冲区大小
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

## Save 方法

描述 保存 Record\_Head 到缓冲区中。

Visual Basic nOutSize = Record\_Head.Save(buf, nBufSize)

C++ HRESULT pRecord\_Head->Save(  
BYTE \*buf, long nBufSize, long \*nOutSize);

参数说明：

参数	类型	出口/入口	描述
buf	Byte	[in]	缓冲区
nBufSize	Long	[in]	缓冲区大小
nOutSize	Long	[out]	返回占用的缓冲区大小

## Set 方法

描述 复制，pIStu0=NULL 则清除所有字段头。

Visual Basic rtl = Record\_Head.Set(pIStu0)

C++ HRESULT pRecord\_Head->Set(

---

```
IRecord_Head *pIStru0, VARIANT_BOOL *rtl);
```

参数说明：

参数	类型	出口/入口	描述
pIStru0	Record_Head	[in]	复制的记录头
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

---

### 3.18、Field 对象

描述 字段对象 Field，包含字段结构和字段值。

#### 3.18.1、属性：

属性	描述
ActualSize Long 只读	字段值的真实字节长度
FieldHD Field_Head 可读写	字段描述头
Value Variant 可读写	字段值

---

#### 3.18.2、方法：无

### 3.19、Record 对象

描述 属性记录对象 Record 包含了属性结构、属性值以及操作属性的方法。

#### 3.19.1、属性：

属性	描述
count Integer 只读	字段数，该属性值始终和 hd 中的 numbfied 一致，设此属性仅为方便上层调用

---

hd	属性结构
Record_Head	
可读写	
item(index)	记录中的字段；index 为 Variant 类型，对应字段名或字段索引号。
Field	
只读	
Value(index)	对应字段值；index 为 Variant 类型，对应字段名或字段索引号。
Variant	
可读写	

### 3.19.2、方法：

#### Append 方法

描述 添加字段。

Visual Basic rtl = Record.Append(Fld)

C++ HRESULT pRecord -> Append(IFld \*pIFld, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
Fld	Field	[in]	添加的字段
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### Insert 方法

描述 在 index 位置插入一个字段。

Visual Basic rtl = Record.Insert(index, Fld)

C++ HRESULT pRecord -> Insert(  
VARIANT index, IFld \*pIFld, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
index	Variant	[in]	字段索引号或字段名
Fld	Field	[in]	插入的字段
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### Remove 方法

描述 删除记录中的字段。

Visual Basic rtl = Record.Remove(index)

C++ HRESULT pRecord -> Remove(VARIANT index, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
index	Variant	[in]	字段索引号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Set 方法**

描述 设置字段。

Visual Basic rtl = Record.Set(ptNewRecord)

C++ HRESULT pRecord -> Set(IRecord \*ptNewRecord);

参数说明：

参数	类型	出口/入口	描述
ptNewRecord	Record	[in]	设置一条记录
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**3.20、RecordSet 对象**

描述 记录集对象 RecordSet 是属性数据的集合。通过该对象，可以从数据集中取出数据，并对数据集中的数据记录进行实际的操作。记录集对象的概念与 VB 中的 RecordSet 的概念类似，用于操作属性数据。记录集对象是一个独立的对象，可将工作区中实体的属性提取出来，在记录集对象中独立的进行处理。

**3.20.1、属性：**

属性	描述
hd	属性结构
Record_Head	
可读写	
numbfield	字段数，该属性值始终和 hd 中的 numbfield 一致
Integer	
只读	
numbrecord	记录数
Long	
只读	
Record	记录集中的当前记录
Record	
只读	

BOF	当前记录位置是否在 RecordSet 中的第一条记录之前
Boolean	
只读	
EOF	当前记录位置是否在 RecordSet 中的最后一条记录之后
Boolean	
只读	
Bookmark	书签, 相当于 RecordSet 中记录的序号, 第 1 条记录从 1 开始
Long	
可读写	

### 3.20.2、方法：

#### Append 方法

描述 添加一条记录。

Visual Basic rtl = RecordSet.Append(pIFlds)

C++ HRESULT pRecordSet->Append(IRecord \*pIFlds);

参数说明：

参数	类型	出口/入口	描述
pIFlds	Record	[in]	添加的记录

#### Move 方法

描述 从 start 指定的位置开始移 numRecords 条记录。

Visual Basic rtl = RecordSet.Move(numRecords, start)

C++ HRESULT pRecordSet->Move(  
long numRecords, Enum\_Bookmark\_Type start, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
numRecords	Long	[in]	要移动的记录数
start	Enum_Bookmark_Type	[in]	书签类型, 表示位置
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

#### MoveFirst 方法

描述 将当前指针移到第一条记录。

Visual Basic rtl = RecordSet.MoveFirst()

C++ HRESULT pRecordSet->MoveFirst(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----



rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE
-----	---------	-------	-----------------------

### MoveLast 方法

描述 将当前指针移到最后一条记录。

Visual Basic rtl = RecordSet.MoveLast()

C++ HRESULT pRecordSet-> MoveLast(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

### MoveNext 方法

描述 将当前指针后移一条记录。

Visual Basic rtl = RecordSet.MoveNext()

C++ HRESULT pRecordSet-> MoveNext(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

### MovePrevious 方法

描述 将当前指针前移一条记录。

Visual Basic rtl = RecordSet.MovePrevious()

C++ HRESULT pRecordSet-> MovePrevious (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

### Remove 方法

描述 删除当前位置记录。

Visual Basic RecordSet.Remove

C++ HRESULT pRecordSet-> Remove ();

参数说明： 无

### Update 方法

描述 更新当前记录值。

Visual Basic RecordSet.Update ptNewVal

C++ HRESULT pRecordSet-> Update (ptNewVal);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Record	[in]	待更新的记录值

### 3.21、GisEnv 对象

描述 GisEnv 对象是 MapGis 工作环境。

#### 3.21.1、属性：

属性	描述
clib String 可读写	汉字库目录
cur String 可读写	当前工作目录
slib String 可读写	系统库目录
sys String 只读	系统目录
temp String 可读写	临时工作目录（工作区使用）

#### 3.21.2、方法：无

### 3.22、Tran\_Par 对象

描述 Tran\_Par 对象是图形变换参数结构。

#### 3.22.1、属性：

属性	描述
----	----

enlarge_rx	比例变换参数
Double	
可读写	
enlarge_ry	比例变换参数
Double	
可读写	
enlarge_rz	比例变换参数
Double	
可读写	
move_dx	平移变换参数
Double	
可读写	
move_dy	平移变换参数
Double	
可读写	
move_dz	平移变换参数
Double	
可读写	
rotate_ang	旋转变换参数，以度为单位表示
Double	
可读写	
type(index)	变换类型及变换顺序，即按 type[0]->type[1]
Enum_Transform_Type	->type[2]的顺序变换，index 为 Integer 类型
可读写	

### 3.22.2、方法：无

## 3.23、Map\_Para 对象

描述 Map\_Para 对象是地图参数结构。

### 3.23.1、属性：

属性	描述
djd	空间数据数据经度跨度，数据单位统一采用 UNIT_DM
Double	
可读写	

dwd	空间数据数据纬度跨度，数据单位统一采用 UNIT_DM
Double	
可读写	
dx	空间数据图纸坐标原点在投影坐标系中的 X 偏移 值，type 坐标系，unit 单位
Double	
可读写	
dy	空间数据图纸坐标原点在投影坐标系中的 Y 偏移 值，type 坐标系，unit 单位
Double	
可读写	
earthParam	空间数据参考椭球体参数类型
Enum_SpatialEP_Type	
可读写	
h	空间数据大地水准面与参考椭球面之间的高差
Double	
可读写	
H2	空间数据投影平面与大地水准面的高差
Double	
可读写	
infoRateX	图形显示参数单位值相对于当前数据单位值的 x 比 例系数，初始值必须赋 0 或 1
Double	
可读写	
infoRatey	图形显示参数单位值相对于当前数据单位值的 y 比 例系数，初始值必须赋 0 或 1
Double	
可读写	
infoUnit	图形显示参数单位
Enum_SpatialUnit_Type	
可读写	
jd0	空间数据数据左下角经度，数据单位统一采用 UNIT_DM
Double	
可读写	
lat	空间数据投影原点纬度，数据单位统一采用 UNIT_DMS 格式
Double	
可读写	
lat1	空间数据双纬线 1 纬度，数据单位统一采用 UNIT_DMS 格式
Double	
可读写	

lat2	空间数据双纬线 2 纬度，数据单位统一采用 UNIT_DMS 格式
Double	
可读写	
levelType	空间数据大地水准面类型
Enum_SpatialRef_Type	
可读写	
lon	空间数据中央经线经度，数据单位统一采用 UNIT_DMS 格
Double	
可读写	
lon1	空间数据双经线 1 经度，数据单位统一采用 UNIT_DMS
Double	
可读写	
lon2	空间数据双经线 2 经度，数据单位统一采用 UNIT_DMS 格式
Double	
可读写	
maph	空间数据图幅理论高；type 坐标系，unit 单位
Double	
可读写	
mapw	空间数据图幅理论宽；type 坐标系，unit 单位
Double	
可读写	
projType	空间数据投影类型
Enum_SpatialPrj_Type	
可读写	
rate	空间数据数据水平比例尺倒数，无单位
Double	
可读写	
TICtype	TIC 点理论坐标系类型
Enum_SpatialRef_Type	
可读写	
TICunit	TIC 点理论坐标单位
Enum_SpatialUnit_Type	
可读写	
type	空间数据水平坐标系类型
Enum_SpatialRef_Type	
可读写	

unit	空间数据水平坐标单位
Enum_SpatialUnit_Type	
可读写	
vRate	空间数据垂向比例尺倒数
Double	
可读写	
vUnit	空间数据垂向数据单位
Enum_SpatialUnit_Type	
可读写	
wd0	空间数据数据左下角纬度，数据单位统一采用
Double	UNIT_DMS 格式
可读写	

### 3.23.2、方法：无

## 3.24、Pnt\_Info\_Sub 对象

描述 Pnt\_Info\_Sub 对象是子图点数据类型。

### 3.24.1、属性：

属性	描述
angle	旋转角度
Double	
可读写	
fclr	辅助颜色
Long	
可读写	
height	子图高度
Double	
可读写	
penw	笔宽
Double	
可读写	
subno	子图号
Long	
可读写	

width	子图宽度
Double	
可读写	

### 3.24.2、方法：无

## 3.25、Pnt\_Info\_Note 对象

描述 Pnt\_Info\_Note 对象是注释点数据类型。

### 3.25.1、属性：

属性	描述
angle	注释角度
Double	
可读写	
chnt	汉字字体
Integer	
可读写	
height	注释高度
Double	
可读写	
hvpl	排列方式
Integer	
可读写	
ifnt	西文字体
Integer	
可读写	
ifnx	汉字字形
Integer	
可读写	
space	注释间隔
Double	
可读写	
width	注释宽度
Double	
可读写	

### 3.25.2、方法：无

## 3.26、Pnt\_Info\_Cir 对象

描述 Pnt\_Info\_Cir 对象是圆点数据类型。

### 3.26.1、属性：

属性	描述
fflag Integer 可读写	填充圆
lcol Long 可读写	填充颜色
penw Double 可读写	笔宽
radiu Double 可读写	圆半径

### 3.26.2、方法：无

## 3.27、Pnt\_Info\_Arc 对象

描述 Pnt\_Info\_Arc 对象是弧点数据类型。

### 3.27.1、属性：

属性	描述
begang Double 可读写	起始角度



endang	终止角度
Double	
可读写	
penw	笔宽
Double	
可读写	
radiu	弧半径
Double	
可读写	

### 3.27.2、方法：无

## 3.28、Pnt\_Info\_Image 对象

描述 Pnt\_Info\_Image 对象是图象点数据类型。

### 3.28.1、属性：

属性	描述
angle	图象角度
Double	
可读写	
height	图象高度
Double	
可读写	
width	图象宽度
Double	
可读写	

### 3.28.2、方法：无

## 3.29、Pnt\_Info\_Text 对象

描述 Pnt\_Info\_Text 对象是版面点数据类型。

### 3.29.1、属性：

属性	描述
angle	版面角度
Double	
可读写	
chnt	汉字字体
Integer	
可读写	
dx	版面高度
Double	
可读写	
dy	版面宽度
Double	
可读写	
height	注释高度
Double	
可读写	
hvpl	排列方式
Integer	
可读写	
ifnt	西文字体
Integer	
可读写	
ifnx	汉字字形
Integer	
可读写	
lspace	纵向间隔
Double	
可读写	
space	横向间隔
Double	
可读写	
width	注释宽度
Double	
可读写	

### 3.29.2、方法：无

### 3.30、Pnt\_Info 对象

**描述** Pnt\_Info 对象是点类型实体图形信息结构。

#### 3.30.1、属性：

属性	描述
arc	弧
Pnt_Info_Arc	
可读写	
ch	注释
Pnt_Info_Note	
可读写	
cir	圆
Pnt_Info_Cir	
可读写	
iclr	点颜色
Long	
可读写	
image	图象
Pnt_Info_Image	
可读写	
infoDx	点参数偏移 dx
Double	
可读写	
infoDy	点参数偏移 dy
Double	
可读写	
layer	图层
Integer	
可读写	
linNo	点所属的线号
Long	
可读写	
ovprnt	透明输出
Integer	
可读写	

rect	区域范围
D_Rect	
可读写	
res0	保留，必须赋 0
Long	
可读写	
res1	保留，必须赋 0
Long	
可读写	
sub	子图
Pnt_Info_Sub	
可读写	
text	版面
Pnt_Info_Text	
可读写	
type	点类型
Enum_Pnt_Type	
可读写	

### 3.30.2、方法：

#### Set 方法

描述            替换点信息。

Visual Basic    rtl = Pnt\_Info.Set(ptNewVal)

C++             HRESULT pPnt\_Info-> Set(IPnt\_Info \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Pnt_Info	[in]	复制的 Pnt_Info 对象
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

## 3.31、Pnt\_Pad\_Note 对象

描述      Pnt\_Pad\_Note 对象是注释点参数字段开关。

### 3.31.1、属性：

属性	描述
----	----

angle	注释角度
Enum_Switch_Type	
可读写	
chnt	汉字字体
Enum_Switch_Type	
可读写	
height	注释高度
Enum_Switch_Type	
可读写	
hvpl	排列方式
Enum_Switch_Type	
可读写	
ifnt	西文字体
Enum_Switch_Type	
可读写	
ifnx	汉字字形
Enum_Switch_Type	
可读写	
space	注释间隔
Enum_Switch_Type	
可读写	
width	注释宽度
Enum_Switch_Type	
可读写	

### 3.31.2、方法：无

## 3.32、Pnt\_Pad\_Sub 对象

描述 Pnt\_Pad\_Sub 对象是子图点参数字段开关。

### 3.32.1、属性：

属性	描述
angle	旋转角度
Enum_Switch_Type	
可读写	

fclr	辅助颜色
Enum_Switch_Type	
可读写	
height	子图高度
Enum_Switch_Type	
可读写	
penw	笔宽
Enum_Switch_Type	
可读写	
subno	子图号
Enum_Switch_Type	
可读写	
width	子图宽度
Enum_Switch_Type	
可读写	

### 3.32.2、方法：无

## 3.33、Pnt\_Pad\_Cir 对象

描述 Pnt\_Pad\_Cir 对象是圆点参数字段开关。

### 3.33.1、属性：

属性	描述
fflag	填充圆
Enum_Switch_Type	
可读写	
lcol	填充颜色
Enum_Switch_Type	
可读写	
penw	笔宽
Enum_Switch_Type	
可读写	
radiu	圆半径
Enum_Switch_Type	
可读写	

### 3.33.2、方法：无

## 3.34、Pnt\_Pad\_Arc 对象

描述 Pnt\_Pad\_Arc 对象是弧点参数字段开关。

### 3.34.1、属性：

属性	描述
begang	起始角度
Enum_Switch_Type	
可读写	
endang	终止角度
Enum_Switch_Type	
可读写	
penw	笔宽
Enum_Switch_Type	
可读写	
radiu	弧半径
Enum_Switch_Type	
可读写	

### 3.34.2、方法：无

## 3.35、Pnt\_Pad\_Image 对象

描述 Pnt\_Pad\_Image 对象是图象点参数字段开关。

### 3.35.1、属性：

属性	描述
angle	图象角度
Enum_Switch_Type	
可读写	

height	图象高度
Enum_Switch_Type	
可读写	
width	图象宽度
Enum_Switch_Type	
可读写	

### 3.35.2、方法：无

## 3.36、Pnt\_Pad\_Text 对象

描述 Pnt\_Pad\_Text 对象是版面点数据类型。

### 3.36.1、属性：

属性	描述
angle	版面角度
Enum_Switch_Type	
可读写	
chnt	汉字字体
Enum_Switch_Type	
可读写	
dx	版面高度
Enum_Switch_Type	
可读写	
dy	版面宽度
Enum_Switch_Type	
可读写	
height	注释高度
Enum_Switch_Type	
可读写	
hvpl	排列方式
Enum_Switch_Type	
可读写	
ifnt	西文字体
Enum_Switch_Type	
可读写	



ifnx	汉字字形
Enum_Switch_Type	
可读写	
lspace	纵向间隔
Enum_Switch_Type	
可读写	
space	横向间隔
Enum_Switch_Type	
可读写	
width	注释宽度
Enum_Switch_Type	
可读写	

### 3.36.2、方法：无

## 3.37、Pnt\_Pad 对象

描述 Pnt\_Pad 对象是点参数字段开关。

### 3.37.1、属性：

属性	描述
arc	弧
Pnt_Pad_Arc	
可读写	
ch	注释
Pnt_Pad_Note	
可读写	
cir	圆
Pnt_Pad_Cir	
可读写	
iclr	点颜色
Enum_Switch_Type	
可读写	
image	图象
Pnt_Pad_Image	
可读写	

---

infoDx	点参数偏移 dx
Enum_Switch_Type	
可读写	
infoDy	点参数偏移 dy
Enum_Switch_Type	
可读写	
layer	图层
Enum_Switch_Type	
可读写	
linNo	点所属的线号
Enum_Switch_Type	
可读写	
ovprnt	透明输出
Enum_Switch_Type	
可读写	
rect	区域范围
Enum_Switch_Type	
可读写	
res0	保留，必须赋 0
Enum_Switch_Type	
可读写	
res1	保留，必须赋 0
Enum_Switch_Type	
可读写	
sub	子图
Pnt_Pad_Sub	
可读写	
text	版面
Pnt_Pad_Text	
可读写	
type	点类型
Enum_Switch_Type	
可读写	

---

### 3.37.2、方法：

**Set 方法**

描述 替换点信息。

Visual Basic rtl = Pnt\_Pad.Set(pad)

C++ HRESULT pPnt\_Pad-> Set(IPnt\_Pad \*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	复制的 Pnt_Pad 对象
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**Turn 方法**

描述 转换值，全部打开或者全部关闭。

Visual Basic Pnt\_Pad.Turn newVal

C++ HRESULT pPnt\_Pad->Turn(Enum\_Switch\_Type newVal);

参数说明：

参数	类型	出口/入口	描述
newVal	Enum_Switch_Type	[in]	转换值

**3.38、Lin\_Info 对象**

描述 Lin\_Info 是线图形信息对象。

**3.38.1、属性：**

属性	描述
fclr	辅助颜色
Long	
可读写	
fltp	辅助线型
Integer	
可读写	
layer	图层
Integer	
可读写	
lclass	线种类
Integer	
可读写	

lclr	线颜色
Long	
可读写	
ltp	线型
Integer	
可读写	
lw	线宽
Double	
可读写	
ovprnt	透明输出
Integer	
可读写	
rect	区域范围
D_Rect	
可读写	
res0	保留，必须赋 0
Long	
可读写	
res1	保留，必须赋 0
Long	
可读写	
xscale	X 系数
Double	
可读写	
yscale	Y 系数
Double	
可读写	

### 3.38.2、方法：

#### Set 方法

描述 替换线信息。

Visual Basic rtl = Lin\_Info.Set(ptNewVal)

C++ HRESULT pLin\_Info->Set(ILin\_Info \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Lin_Info	[in]	替换的 Lin_Info 对象

rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE
-----	---------	-------	-----------------------

### 3.39、Lin\_Pad 对象

描述 Lin\_Pad 对象是线参数开关。

#### 3.39.1、属性：

属性	描述
fclr Enum_Switch_Type 可读写	辅助颜色
fltp Enum_Switch_Type 可读写	辅助线型
layer Enum_Switch_Type 可读写	图层
lclass Enum_Switch_Type 可读写	线种类
lclr Enum_Switch_Type 可读写	线颜色
ltp Enum_Switch_Type 可读写	线型
lw Enum_Switch_Type 可读写	线宽
ovprnt Enum_Switch_Type 可读写	透明输出
rect Enum_Switch_Type 可读写	区域范围

res0	保留，必须赋 0
Enum_Switch_Type	
可读写	
res1	保留，必须赋 0
Enum_Switch_Type	
可读写	
xscale	X 系数
Enum_Switch_Type	
可读写	
yscale	Y 系数
Enum_Switch_Type	
可读写	

### 3.39.2、方法：

#### Set 方法

描述 替换线参数字段开关信息。

Visual Basic rtl = Lin\_Pad.Set(ptNewVal)

C++ HRESULT pLin\_Pad->Set(ILin\_Pad \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Lin_Pad	[in]	替换的 Lin_Pad 对象
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

#### Turn 方法

描述 转换值，全部打开或者全部关闭。

Visual Basic Lin\_Pad.Turn newVal

C++ HRESULT pLin\_Pad->Turn(Enum\_Switch\_Type newVal);

参数说明：

参数	类型	出口/入口	描述
newVal	Enum_Switch_Type	[in]	转换值

## 3.40、Lin\_Top 对象

描述 Lin\_Top 是线拓扑结构对象。

### 3.40.1、属性：

属性	描述
endNod	终止结点
Long	
可读写	
lPoly	弧段左多边形
Long	
可读写	
lWeigh	左权
Double	
可读写	
rPoly	弧段右多边形
Long	
可读写	
rWeigh	右权
Double	
可读写	
stNod	起始结点
Long	
可读写	

### 3.40.2、方法：

#### Set 方法

描述 替换线拓扑数据信息。

Visual Basic rtl = Lin\_Top.Set(ptNewVal)

C++ HRESULT pLin\_Top->Set(ILin\_Top \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Lin_Top	[in]	替换的 Pnt_Top 对象
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

### 3.41、Arc\_Alloc\_Info 对象

描述 Arc\_Alloc\_Info 对象是网络弧段分配信息。

#### 3.41.1、属性：

属性	描述
allocCenter Long 可读写	分配中心
allocDirect Integer 可读写	分配方向
cumuImped Double 可读写	累积阻碍
demand Double 可读写	需求
demandMeet Double 可读写	被满足的需求
preArc Long 可读写	前一弧段

### 3.41.2、方法：

#### Set 方法

描述 替换网络弧段分配信息。

Visual Basic rtl = Arc\_Alloc\_Info.Set(ptNewVal)

C++ HRESULT pArc\_Alloc\_Info->Set(IArc\_Alloc\_Info \*ptNewVal,  
VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Arc_Alloc_Info	[in]	替换的 Arc_Alloc_Info 对象
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

## 3.42、Reg\_Info 对象

描述 Reg\_Info 是区实体图形信息对象。

### 3.42.1、属性：



属性	描述
basLN Integer 可读写	构成斜坡基线的弧段
clr Long 可读写	区域填充色
fmode Integer 可读写	覆盖模式
layer Integer 可读写	图层
patclr Long 可读写	图案颜色
pathei Double 可读写	图案高度
patno Integer 可读写	填充图案号
patwid Double 可读写	图案宽度
rect D_Rect 可读写	区域范围
res0 Long 可读写	保留，必须赋 0
res1 Long 可读写	保留，必须赋 0

### 3.42.2、方法：

**Set 方法**

描述 替换线信息。

Visual Basic rtl = Reg\_Info.Set(ptNewVal)

C++ HRESULT pReg\_Info->Set(IReg\_Info \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Reg_Info	[in]	替换的 Reg_Info 对象
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

**3.43、Reg\_Pad 对象**

描述 Reg\_Pad 对象是区参数字段开关。

**3.43.1、属性：**

属性	描述
basLN	构成斜坡基线的弧段
Enum_Switch_Type	
可读写	
clr	区域填充色
Enum_Switch_Type	
可读写	
fmode	覆盖模式
Enum_Switch_Type	
可读写	
layer	图层
Enum_Switch_Type	
可读写	
patclr	图案颜色
Enum_Switch_Type	
可读写	
pathei	图案高度
Enum_Switch_Type	
可读写	
patno	填充图案号
Enum_Switch_Type	
可读写	

patwid	图案宽度
Enum_Switch_Type	
可读写	
rect	区域范围
Enum_Switch_Type	
可读写	
res0	保留，必须赋 0
Enum_Switch_Type	
可读写	
res1	保留，必须赋 0
Enum_Switch_Type	
可读写	

### 3.43.2、方法：

#### Set 方法

描述 替换区参数字段开关信息。

Visual Basic rtl = Reg\_Pad.Set(ptNewVal)

C++ HRESULT pReg\_Pad->Set(IReg\_Pad \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Reg_Pad	[in]	替换的 Reg_Pad 对象
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

#### Turn 方法

描述 转换值，全部打开或者全部关闭。

Visual Basic Reg\_Pad.Turn newVal

C++ HRESULT pReg\_Pad->Turn(Enum\_Switch\_Type newVal);

参数说明：

参数	类型	出口/入口	描述
newVal	Enum_Switch_Type	[in]	转换值

## 3.44、Net\_Info 对象

描述 Net\_Info 对象是网络类型实体图形信息。

### 3.44.1、属性：

属性	描述
fclr Long 可读写	辅助颜色
fltp Integer 可读写	辅助线型
layer Integer 可读写	图层
lclass Integer 可读写	线种类
lclr Long 可读写	颜色
ltp Integer 可读写	线型
lw Double 可读写	线宽
ovprnt Integer 可读写	透明输出
rect D_Rect 可读写	区域范围
res0 Long 可读写	保留，必须赋 0
res1 Long 可读写	保留，必须赋 0
xscale Double 可读写	X 系数

yscale	Y 系数
Double	
可读写	

### 3.44.2、方法：

#### Set 方法

描述 替换网络类型实体图形信息。

Visual Basic rtl = Net\_Info.Set(ptNewVal)

C++ HRESULT pNet\_Info->Set(INet\_Info \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Net_Info	[in]	替换的 Net_Info 对象
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

## 3.45、Net\_Pad 对象

描述 Net\_Pad 对象是网络参数字段开关。

### 3.45.1、属性：

属性	描述
fclr	辅助颜色
Enum_Switch_Type	
可读写	
fltp	辅助线型
Enum_Switch_Type	
可读写	
layer	图层
Enum_Switch_Type	
可读写	
lclass	线种类
Enum_Switch_Type	
可读写	
lclr	颜色
Enum_Switch_Type	
可读写	

ltp	线型
Enum_Switch_Type	
可读写	
lw	线宽
Enum_Switch_Type	
可读写	
ovprnt	透明输出
Enum_Switch_Type	
可读写	
rect	区域范围
Enum_Switch_Type	
可读写	
res0	保留，必须赋 0
Enum_Switch_Type	
可读写	
res1	保留，必须赋 0
Enum_Switch_Type	
可读写	
xscale	X 系数
Enum_Switch_Type	
可读写	
yscale	Y 系数
Enum_Switch_Type	
可读写	

### 3.45.2、方法：

#### Set 方法

描述 替换网络参数字段开关信息。

Visual Basic rtl = Net\_Pad.Set(ptNewVal)

C++ HRESULT pNet\_Pad->Set(INet\_Pad \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Net_Pad	[in]	替换的 Net_Pad 对象
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

#### Turn 方法

描述 转换值，全部打开或者全部关闭。

Visual Basic Net\_Pad.Turn newVal

C++ HRESULT pNet\_Pad->Turn(Enum\_Switch\_Type newVal);

参数说明：

参数	类型	出口/入口	描述
newVal	Enum_Switch_Type	[in]	转换值

### 3.46、Nod\_Center\_Info 对象

描述 Nod\_Center\_Info 对象是网络中心结点信息。

#### 3.46.1、属性：

属性	描述
averimped Double 可读写	平均阻抗
capacity Double 可读写	容量
delay Double 可读写	延迟
limit Double 可读写	限度
linf Lin_Info 可读写	线信息
maximped Double 可读写	最大阻抗
perccap Double 可读写	容量分配百分比

### 3.46.2、方法：

#### Set 方法

描述 替换网络中心结点信息。

Visual Basic rtl = Nod\_Center\_Info.Set(ptNewVal)

C++ HRESULT pNod\_Center\_Info->Set(INod\_Center\_Info\*ptNewVal,  
VARIANT\_BOOL\*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Nod_Center_Info	[in]	替换的 Nod_Center_Info 对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### 3.47、Nod\_Stop\_Info 对象

描述 Nod\_Stop\_Info 对象是网络终止结点信息。

#### 3.47.1、属性：

属性	描述
demand	需求
Double	
可读写	

#### 3.47.2、方法：无

### 3.48、Nod\_Barrier\_Info 对象

描述 Nod\_Barrier\_Info 对象是网络障碍结点信息。

#### 3.48.1、属性：

属性	描述
res	
Long	
可读写	

#### 3.48.2、方法：无



### 3.49、Pnt 对象

**描述** 点对象 Pnt 封装了对点数据的基本操作。

#### 3.49.1、属性：

属性	描述
pos	点位置
D_3Dot 可读写	
NoteDat	注记点字符串
String 可读写	
inf	点信息
Pnt_Info 可读写	

#### 3.49.2、方法：

##### Set 方法

**描述** 设置新点值到对象中。

Visual Basic rtl = Pnt.Set(ptNewVal)

C++ HRESULT pPnt-> Set(IPnt \*ptNewVal, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
ptNewVal	Pnt	[in]	新的点
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### 3.50、PntSet 对象

**描述** 点集对象 PntSet 是 Pnt 对象的集合，它封装了对点集合的操作。

#### 3.50.1、属性：

属性	描述
count	点集中的点数
long 只读	

item(index)	第 index 项 Pnt 值
Pnt	
只读	

### 3.50.2、方法：

#### Append 方法

描述 添加一个点实体。

Visual Basic PntSet.Append pIPnt

C++ HRESULT pPntSet->Append(IPnt \*pIPnt);

参数说明：

参数	类型	出口/入口	描述
pIPnt	Pnt	[in]	添加的点

#### Insert 方法

描述 在 index 位置插入点。

Visual Basic PntSet.Insert index, pIPnt

C++ HRESULT pPntSet->Insert(long index, IPnt \*pIPnt);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	插入位置
pIPnt	Pnt	[in]	插入的点

#### Remove 方法

描述 从 index 位置开始删除 nCount 个点。

Visual Basic PntSet.Remove index, nCount

C++ HRESULT pPntSet->Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	开始删除位置
nCount	Long	[in]	删除的点个数

#### RemoveAll 方法

描述 删除所有点。

Visual Basic PntSet.RemoveAll

C++ HRESULT pPntSet->RemoveAll();

参数说明： 无

**Update 方法**

描述 更新第 index 项数据。

Visual Basic PntSet.Update index, ptNewVal

C++ HRESULT pPntSet->Set(long index, IPnt \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引号
ptNewVal	Pnt	[in]	更新的点

**Set 方法**

描述 用新点集替换。

Visual Basic rtl = PntSet.Set(ptNewVal)

C++ HRESULT pPntSet->Set(IPntSet \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	PntSet	[in]	新点集
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**3.51、Lin 对象**

描述 二维线对象 Lin 封装了对二维线的操作。

**3.51.1、属性：**

属性	描述
inf	线信息
Lin_Info	
可读写	
pos	组成线的点坐标
D_DotSet	
可读写	

**3.51.2、方法：****Set 方法**

描述 设置新二维线值到对象中。

Visual Basic rtl = Lin.Set(ptNewVal)

C++ HRESULT pLin->Set(ILin \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Lin	[in]	新的二维线
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### 3.52、LinSet 对象

描述 线集对象 LinSet 是 Lin 对象的集合，封装了对二维线集合的操作。

#### 3.52.1、属性：

属性	描述
count	线集中的线数
long 只读	
item(index)	第 index 项 Lin 值
Lin 只读	

#### 3.52.2、方法：

##### Append 方法

描述 添加一个线实体。

Visual Basic LinSet.Append pILin

C++ HRESULT pLinSet->Append(ILin \*pILin);

参数说明：

参数	类型	出口/入口	描述
pILin	Lin	[in]	添加的线

##### Insert 方法

描述 在 index 位置插入线。

Visual Basic LinSet.Insert index, pILin

C++ HRESULT pLinSet->Insert(long index, ILin \*pILin);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	插入位置
pILin	Lin	[in]	插入的线

#### Remove 方法

描述 从 index 位置开始删除 nCount 条线。

Visual Basic LinSet.Remove index, nCount

C++ HRESULT pLinSet->Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	开始删除位置
nCount	Long	[in]	删除的线个数

#### RemoveAll 方法

描述 删除所有线。

Visual Basic LinSet.RemoveAll

C++ HRESULT pLinSet->RemoveAll();

参数说明： 无

#### Update 方法

描述 更新第 index 项数据。

Visual Basic LinSet.Update index, ptNewVal

C++ HRESULT pLinSet-> Update(long index, ILin \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引号
ptNewVal	Lin	[in]	更新的线

### 3.53、Lin3D 对象

描述 三维线对象 Lin3D 封装了对三维线的操作。

#### 3.53.1、属性：

属性	描述
inf	线信息
Lin_Info	
可读写	
pos	组成线的三维点坐标
D_3DotSet	
可读写	

### 3.53.2、方法：

#### Set 方法

描述 设置新三维线值到对象中。

Visual Basic rtl = Lin3D.Set(ptNewVal)

C++ HRESULT pLin3D->Set(ILin3D \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Lin3D	[in]	新的三维线
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## 3.54、Lin3DSet 对象

描述 三维线集合对象 Lin3DSet 是 Lin3D 对象的集合，封装了对三维线集合的操作。

### 3.54.1、属性：

属性	描述
count	线集中的线数
long 只读	
item(index)	第 index 项 Lin3D 值
Lin3D 只读	

### 3.54.2、方法：

#### Append 方法

描述 添加一条三维线。

Visual Basic Lin3DSet.Append ptLin3d

C++ HRESULT pLin3DSet->Append(ILin3D \*ptLin3d);

参数说明：

参数	类型	出口/入口	描述
ptLin3d	Lin3D	[in]	添加的三维线

#### Insert 方法

描述 在 index 位置插入线。

Visual Basic Lin3DSet.Insert index, ptLin3d

C++ HRESULT pLin3DSet->Insert(long index, ILin3D \* ptLin3d);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	插入位置
ptLin3d	Lin3D	[in]	插入的线

### Remove 方法

描述 从 index 位置开始删除 nCount 条线。

Visual Basic Lin3DSet.Remove index, nCount

C++ HRESULT pLin3DSet->Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	开始删除位置
nCount	Long	[in]	删除的三维线条数

### RemoveAll 方法

描述 删除所有三维线。

Visual Basic Lin3DSet.RemoveAll

C++ HRESULT pLin3DSet->RemoveAll();

参数说明： 无

### Update 方法

描述 更新第 index 项数据。

Visual Basic Lin3DSet.Update index, ptNewVal

C++ HRESULT pLin3DSet->Update (long index, ILin3D \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引号
ptNewVal	Lin3D	[in]	更新的线

## 3.55、Reg 对象

描述 区对象 Reg 封装了对区的基本操作。

### 3.55.1、属性：

属性	描述
----	----

count	区边界弧段数据 rdata 总项数
Long	
只读	
inf	区信息
Reg_Info	
可读写	
rdata	区边界弧段数据（弧段号）
Long	
可读写	

### 3.55.2、方法：

#### Append 方法

描述 添加一条区弧段。

Visual Basic Reg.Append regNo

C++ HRESULT pReg->Append(long regNo);

参数说明：

参数	类型	出口/入口	描述
regNo	Long	[in]	添加的弧段号

#### Insert 方法

描述 插入一条弧段。

Visual Basic Reg.Insert index, regNo

C++ HRESULT pReg->Insert(long index, long regNo);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	插入位置
regNo	Long	[in]	插入的弧段号

#### Remove 方法

描述 从 index 位置开始删除 nCount 条弧段。

Visual Basic Reg.Remove index, nCount

C++ HRESULT pReg->Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	开始删除位置
nCount	Long	[in]	删除的弧段数



**RemoveAll 方法**

描述 删除所有弧段。

Visual Basic Reg.RemoveAll

C++ HRESULT pReg->RemoveAll();

参数说明： 无

**GetCircleNum 方法**

描述 取区边界圈数。

Visual Basic circleNum = Reg.GetCircleNum()

C++ HRESULT pReg->GetCircleNum(long \*circleNum);

参数说明：

参数	类型	出口/入口	描述
circleNum	Long	[out]	区边界圈数

**Set 方法**

描述 设置新区到对象中。

Visual Basic rtl = Reg.Set(ptNewVal)

C++ HRESULT pReg->Set(IReg \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Reg	[in]	新的区
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**3.56、RegSet 对象**

描述 区集合对象 RegSet 是 Reg 对象的集合，封装了对区集合的操作。

**3.56.1、属性：**

属性	描述
count	区集中的区个数
long 只读	
item(index)	第 index 项 Reg 值
Reg 只读	

### 3.56.2、方法：

#### Append 方法

描述 添加一个区实体。

Visual Basic RegSet.Append Reg

C++ HRESULT pRegSet->Append(IReg \*pIReg);

参数说明：

参数	类型	出口/入口	描述
Reg	Reg	[in]	添加的区

#### Insert 方法

描述 在 index 位置插入一个区。

Visual Basic RegSet.Insert index, Reg

C++ HRESULT pRegSet->Insert(long index, IReg \*pIReg);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	插入位置
Reg	Reg	[in]	插入的区

#### Remove 方法

描述 从 index 位置开始删除 nCount 个区。

Visual Basic RegSet.Remove index, nCount

C++ HRESULT pRegSet->Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	开始删除位置
nCount	Long	[in]	删除的区个数

#### RemoveAll 方法

描述 删除所有区。

Visual Basic RegSet.RemoveAll

C++ HRESULT pRegSet->RemoveAll();

参数说明： 无

#### Update 方法

描述 更新第 index 项数据。

Visual Basic RegSet.Update index, ptNewVal

C++ HRESULT pRegSet->Update (long index, IReg \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引号
ptNewVal	Reg	[in]	更新的区

## 3.57、Net 对象

描述 网对象 Net 封装了对网的基本操作。

### 3.57.1、属性：

属性	描述
count Long 只读	网络数据 ndata 总项数
inf Net_Info 可读写	网信息
ndata Long 可读写	网络数据（网段段号）

### 3.57.2、方法：

#### Append 方法

描述 添加一条网段。

Visual Basic Net.Append netNo

C++ HRESULT pNet->Append(long netNo);

参数说明：

参数	类型	出口/入口	描述
netNo	Long	[in]	添加的网段号

#### Insert 方法

描述 插入一条网段。

Visual Basic Net.Insert index, netNo

C++ HRESULT pNet->Insert(long index, long netNo);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	插入位置
netNo	Long	[in]	插入的网段号

**Remove 方法**

描述 从 index 位置开始删除 nCount 条网段。

Visual Basic Net.Remove index, nCount

C++ HRESULT pNet->Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	开始删除位置
nCount	Long	[in]	删除的网段数

**RemoveAll 方法**

描述 删除所有网段。

Visual Basic Net.RemoveAll

C++ HRESULT pNet->RemoveAll();

参数说明： 无

**Set 方法**

描述 设置新网到对象中。

Visual Basic rtl = Net.Set(ptNewVal)

C++ HRESULT pNet->Set(INet \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Net	[in]	新的网
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**3.58、NetSet 对象**

描述 网集合对象 NetSet 是 Net 对象的集合，封装了对网集合的操作。

**3.58.1、属性：**

属性	描述
----	----

count	网集合中的网数
long	
只读	
item(index)	第 index 项 Net 值
Net	
只读	

### 3.58.2、方法：

#### Append 方法

描述 添加一个网实体。

Visual Basic NetSet.Append pINet

C++ HRESULT pNetSet->Append(INet \*pINet);

参数说明：

参数	类型	出口/入口	描述
pINet	Net	[in]	添加的网

#### Insert 方法

描述 在 index 位置插入一个网。

Visual Basic NetSet.Insert index, Net

C++ HRESULT pNetSet->Insert(long index, INet \*pINet);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	插入位置
Net	Net	[in]	插入的网

#### Remove 方法

描述 从 index 位置开始删除 nCount 个网。

Visual Basic NetSet.Remove index, nCount

C++ HRESULT pNetSet->Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	开始删除位置
nCount	Long	[in]	删除的网个数

#### RemoveAll 方法

描述 删除所有网。

Visual Basic NetSet.RemoveAll

C++ HRESULT pNetSet->RemoveAll();

参数说明： 无

### Update 方法

描述 更新第 index 项数据。

Visual Basic NetSet.Update index, ptNewVal

C++ HRESULT pNetSet-> Update (long index, INet \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引号
ptNewVal	Net	[in]	更新的网

## 3.59、Path 对象

描述 网络路径对象 Path 封装了对网络路径的基本操作。

### 3.59.1、属性：

属性	描述
count_nod Long 只读	路径中包含的结点数
count_arc Long 只读	路径中包含的弧段数
item_nod(index) Long 可读写	第 index 项结点数据
item_arc(index) Long 可读写	第 index 项弧段数据
inf Lin_Info 可读写	路径图形信息

### 3.59.2、方法：

**AppendNod 方法**

描述 添加一个结点。

Visual Basic Path.AppendNod nodNo

C++ HRESULT pPath-> AppendNod(long nodNo);

参数说明：

参数	类型	出口/入口	描述
nodNo	Long	[in]	添加的结点号

**AppendArc 方法**

描述 添加一条弧段。

Visual Basic Path.AppendArc arcNo

C++ HRESULT pPath-> AppendArc(long arcNo);

参数说明：

参数	类型	出口/入口	描述
arcNo	Long	[in]	添加的弧段号

**InsertNod 方法**

描述 插入一个结点。

Visual Basic Path.InsertNod index, nodNo

C++ HRESULT pPath->InsertNod(long index, long nodNo);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	插入位置
nodNo	Long	[in]	插入的结点号

**InsertArc 方法**

描述 插入一条弧段。

Visual Basic Path.InsertArc index, arcNo

C++ HRESULT pPath->InsertArc(long index, long arcNo);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	插入位置
arcNo	Long	[in]	插入的弧段号

**RemoveNod 方法**

描述 从 index 位置开始删除 nCount 个结点。

Visual Basic Path.RemoveNod index, nCount

C++ HRESULT pPath->RemoveNod(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	开始删除位置
nCount	Long	[in]	删除的结点数

### RemoveArc 方法

描述 从 index 位置开始删除 nCount 条弧段。

Visual Basic Path.RemoveArc index, nCount

C++ HRESULT pPath->RemoveArc(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	开始删除位置
nCount	Long	[in]	删除的弧段数

### RemoveAllNod 方法

描述 删除所有结点。

Visual Basic Path.RemoveAllNod

C++ HRESULT pPath->RemoveAllNod();

参数说明： 无

### RemoveAllArc 方法

描述 删除所有弧段。

Visual Basic Path.RemoveAllArc

C++ HRESULT pPath->RemoveAllArc();

参数说明： 无

### Set 方法

描述 设置新路径到对象中。

Visual Basic rtl = Path.Set(ptNewVal)

C++ HRESULT pPath->Set(IPath \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Path	[in]	新的路径
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE



### 3.60、PathSet 对象

**描述** 路径集合对象 PathSet 是 Path 对象的集合，封装了对网络路径集合的操作。

#### 3.60.1、属性：

属性	描述
count	路径数
Long 只读	
item(index)	第 index 条路径
Path 只读	

#### 3.60.2、方法：

##### Append 方法

**描述** 添加一条路径

Visual Basic PathSet.Append Path

C++ HRESULT pPathSet->Append(IPath \*pIPath);

**参数说明：**

参数	类型	出口/入口	描述
Path	Path	[in]	添加的路径

##### Insert 方法

**描述** 在 index 位置插入一条路径

Visual Basic PathSet.Insert index, Path

C++ HRESULT pPathSet->Insert(long index, IPath \*pIPath);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	插入位置
Path	Path	[in]	插入的路径

##### Remove 方法

**描述** 从 index 位置开始删除 nCount 条路径

Visual Basic PathSet.Remove index, nCount

C++ HRESULT pPathSet->Remove(long index, long nCount);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	开始删除位置
nCount	Long	[in]	删除的路径数

**RemoveAll 方法**

描述 删除所有路径

Visual Basic PathSet.RemoveAll

C++ HRESULT pPathSet->RemoveAll();

参数说明： 无

**Update 方法**

描述 更新第 index 项数据

Visual Basic PathSet.Update index, ptNewVal

C++ HRESULT pPathSet->Update (long index, IPath \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引号
ptNewVal	Path	[in]	更新的路径

**3.61、GeoPnt 对象**

描述 GeoPnt 对象是地理点实体对象。GeoPnt 对象包含了图形实体 Pnt 对象和实体属性数据。

**3.61.1、属性：**

属性	描述
att	GeoPnt 属性值
Record	
可读写	
pos	点坐标对象
D_3Dot	
可读写	
NoteDat	注记点字符串
String	
可读写	

inf	点信息
Pnt_Info	
可读写	

### 3.61.2、方法：

#### Set 方法

描述 替换 GeoPnt 点实体。

Visual Basic rtl = GeoPnt.Set(ptNewVal)

C++ HRESULT pGeoPnt -> Set(IGeoPnt \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	GeoPnt	[in]	GeoPnt 点对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## 3.62、GeoPntSet 对象

描述 地理点实体集合对象 GeoPntSet 是 GeoPnt 对象的集合，它封装了对地理点实体对象集合的操作。

### 3.62.1、属性：

属性	描述
count	GeoPnt 实体数
Long	
只读	
item (index)	第 index 个 GeoPnt 值
GeoPnt	
只读	

### 3.62.2、方法：

#### Append 方法

描述 添加一个 GeoPnt 实体。

Visual Basic GeoPntSet.Append pIGeoPnt

C++ HRESULT pGeoPntSet -> Append(IGeoPnt \*pIGeoPnt);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

pIGeoPnt	GeoPnt	[in]	添加的 GeoPnt 实体
----------	--------	------	---------------

**Insert 方法**

描述 在 index 处插入一个 GeoPnt 实体。

Visual Basic GeoPntSet.Insert index, pIGeoPnt

C++ HRESULT pGeoPntSet -> Insert(long index, IGeoPnt \* pIGeoPnt);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
pIGeoPnt	GeoPnt	[in]	插入的 GeoPnt 实体

**Remove 方法**

描述 从 index 位置开始删除 nCount 个 GeoPnt 实体。

Visual Basic GeoPntSet.Remove index, nCount

C++ HRESULT pGeoPntSet -> Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
nCount	Long	[in]	删除的 GeoPnt 实体个数

**RemoveAll 方法**

描述 删除所有的 GeoPnt 实体。

Visual Basic GeoPntSet.RemoveAll

C++ HRESULT pGeoPntSet -> RemoveAll ();

参数说明： 无

**Update 方法**

描述 更新第 index 项数据。

Visual Basic GeoPntSet.Update index, ptNewVal

C++ HRESULT pGeoPntSet-> Update (long index, IGeoPnt \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引号
ptNewVal	GeoPnt	[in]	更新的 GeoPnt 实体

### 3.63、GeoLin 对象

**描述** GeoLin 对象是地理线实体对象。GeoLin 对象包含了图形实体 Lin 对象和实体属性数据。

#### 3.63.1、属性：

属性	描述
att	GeoLin 属性值
Record	
可读写	
inf	地理线信息
Lin_Info	
可读写	
pos	线坐标对象
D_DotSet	
可读写	

#### 3.63.2、方法：

##### Set 方法

**描述** 替换 GeoLin 线实体。

Visual Basic rtl = GeoLin.Set(ptNewVal)

C++ HRESULT pGeoLin-> Set(IGeoLin \*ptNewVal, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
ptNewVal	GeoLin	[in]	GeoLin 对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### 3.64、GeoLinSet 对象

**描述** 地理线实体集合对象 GeoLinSet 是 GeoLin 对象的集合，它封装了对地理线实体对象集合的操作。

#### 3.64.1、属性：

属性	描述
----	----

count	GeoLin 实体数
Long	
只读	
item (index)	第 index 个 GeoLin 值
GeoLin	
只读	

### 3.64.2、方法：

#### Append 方法

描述 添加一个 GeoLin 实体。

Visual Basic GeoLinSet.Append pIGeoLin

C++ HRESULT pGeoLinSet -> Append(IGeoLin \* pIGeoLin);

参数说明：

参数	类型	出口/入口	描述
pIGeoLin	GeoLin	[in]	添加的 GeoLin 实体

#### Insert 方法

描述 在 index 处插入一个 GeoLin 实体。

Visual Basic GeoLinSet.Insert index, pIGeoLin

C++ HRESULT pGeoLinSet -> Insert(long index, IGeoLin \* pIGeoLin);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
pIGeoLin	GeoLin	[in]	插入的 GeoLin 实体

#### Remove 方法

描述 从 index 位置开始删除 nCount 个 GeoLin 实体。

Visual Basic GeoLinSet.Remove index, nCount

C++ HRESULT pGeoLinSet -> Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
nCount	Long	[in]	删除的 GeoLin 实体个数

#### RemoveAll 方法

描述 删除所有的 GeoLin 实体。

Visual Basic GeoLinSet.RemoveAll

C++ HRESULT pGeoLinSet -> RemoveAll ();

参数说明： 无

### Update 方法

描述 更新第 index 项数据。

Visual Basic GeoLinSet.Update index, ptNewVal

C++ HRESULT pGeoLinSet-> Update (long index, IGeoLin \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引号
ptNewVal	GeoLin	[in]	更新的 GeoLin 实体

## 3.65、GeoLin3D 对象

描述 GeoLin3D 对象是地理三维线实体对象。GeoLin3D 对象包含了图形实体 Lin3D 对象和实体属性数据。

### 3.65.1、属性：

属性	描述
att	GeoLin3D 属性值
Record	
可读写	
inf	地理三维线信息
Lin_Info	
可读写	
pos	线坐标对象
D_3DotSet	
可读写	

### 3.65.2、方法：

#### Set 方法

描述 替换 GeoLin3D 线实体。

Visual Basic rtl = GeoLin3D.Set(ptNewVal)

C++ HRESULT pGeoLin3D->Set(  
IGeoLin3D \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	GeoLin3D	[in]	GeoLin3D 对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### 3.66、GeoLin3DSet 对象

**描述** 地理三维线实体集合对象 GeoLin3DSet 是 Geo3DLin 对象的集合，它封装了对地理三维线实体对象集合的操作。

#### 3.66.1、属性：

属性	描述
count	GeoLin3D 实体数
Long 只读	
item (index)	第 index 个 GeoLin3D 值
GeoLin3D 只读	

#### 3.66.2、方法：

##### Append 方法

**描述** 添加一个 GeoLin3D 实体。

Visual Basic GeoLin3DSet.Append ptNewVal

C++ HRESULT pGeoLin3DSet -> Append(IGeoLin3D \*ptNewVal);

**参数说明：**

参数	类型	出口/入口	描述
ptNewVal	GeoLin3D	[in]	添加的 GeoLin3D 实体

##### Insert 方法

**描述** 插入一个 GeoLin3D 实体。

Visual Basic GeoLin3DSet.Insert index, ptNewVal

C++ HRESULT pGeoLin3DSet -> Insert(long index, IGeoLin3D \* ptNewVal);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	索引值
ptNewVal	GeoLin3D	[in]	插入的 GeoLin3D 实体



**Remove 方法**

描述 从 index 位置开始删除 nCount 个 GeoLin3D 实体。

Visual Basic GeoLin3DSet.Remove index, nCount

C++ HRESULT pGeoLin3DSet -> Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
nCount	Long	[in]	删除的 GeoLin3D 个数

**RemoveAll 方法**

描述 删除所有的 GeoLin3D 实体。

Visual Basic GeoLin3DSet.RemoveAll

C++ HRESULT pGeoLin3DSet -> RemoveAll ();

参数说明： 无

**Update 方法**

描述 更新第 index 项数据。

Visual Basic GeoLin3DSet.Update index, ptNewVal

C++ HRESULT pGeoLin3DSet -> Update (long index, IGeoLin3D \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引号
ptNewVal	GeoLin3D	[in]	更新的 GeoLin3D 实体

**3.67、GeoReg 对象**

描述 GeoReg 对象是地理区实体对象。GeoReg 对象包含了图形实体 Reg 对象和实体属性数据。

**3.67.1、属性：**

属性	描述
att	GeoReg 属性值
Record	
可读写	
count	区边界弧段数据总项数
Long	
只读	

rdata (index)	第 index 个区边界弧段数据 (弧段号)
Long	
可读写	
inf	区信息
Reg_Info	
可读写	

### 3.67.2、方法：

#### Append 方法

描述 添加弧段。

Visual Basic GeoReg.Append arcNo

C++ HRESULT pGeoReg -> Append(long arcNo);

参数说明：

参数	类型	出口/入口	描述
arcNo	Long	[in]	添加的弧段号

#### Insert 方法

描述 在 index 处插入弧段。

Visual Basic GeoReg.Insert index, arcNo

C++ HRESULT pGeoReg-> Insert(long index, long arcNo);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
arcNo	Long	[in]	插入的弧段号

#### Remove 方法

描述 从 index 位置开始删除 nCount 个弧段。

Visual Basic GeoReg.Remove index, nCount

C++ HRESULT pGeoReg -> Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
nCount	Long	[in]	删除的弧段个数

#### RemoveAll 方法

描述 删除所有的弧段。

Visual Basic GeoReg.RemoveAll

C++ HRESULT pGeoReg -> RemoveAll ();

参数说明： 无

### Set 方法

描述 替换 GeoReg 区实体。

Visual Basic rtl = GeoReg.Set(ptNewVal)

C++ HRESULT pGeoReg -> Set(IGeoReg \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	GeoReg	[in]	GeoReg 对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetCircleNum 方法

描述 获取区边界圈数。

Visual Basic CircleNum = GeoReg.GetCircleNum()

C++ HRESULT pGeoReg -> GetCircleNum(long \*CircleNum);

参数说明：

参数	类型	出口/入口	描述
CircleNum	Long	[out]	区边界圈数

## 3.68、GeoRegSet 对象

描述 地理区实体集合对象 GeoRegSet 是 GeoReg 对象的集合，它封装了对地理区实体对象集合的操作。

### 3.68.1、属性：

属性	描述
count	GeoReg 实体数
Long 只读	
item (index)	第 index 个 GeoReg 值
GeoReg 只读	

### 3.68.2、方法：

**Append 方法**

描述 添加一个 GeoReg 实体。

Visual Basic GeoRegSet.Append geoReg

C++ HRESULT pGeoRegSet -> Append(IGeoReg \*geoReg);

参数说明：

参数	类型	出口/入口	描述
geoReg	GeoReg	[in]	添加的 GeoReg 实体

**Insert 方法**

描述 插入一个 GeoReg 实体。

Visual Basic GeoRegSet.Insert index, geoReg

C++ HRESULT pGeoRegSet -> Insert(long index, IGeoReg \* geoReg);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
geoReg	GeoReg	[in]	插入的 GeoReg 实体

**Remove 方法**

描述 从 index 位置开始删除 nCount 个 GeoReg 实体。

Visual Basic GeoRegSet.Remove index, nCount

C++ HRESULT pGeoRegSet -> Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
nCount	Long	[in]	删除的 GeoReg 个数

**RemoveAll 方法**

描述 删除所有的 GeoReg 实体。

Visual Basic GeoRegSet.RemoveAll

C++ HRESULT pGeoRegSet -> RemoveAll ();

参数说明： 无

**Update 方法**

描述 更新第 index 项数据。

Visual Basic GeoRegSet.Update index, ptNewVal

C++ HRESULT pGeoRegSet-> Update (long index, IGeoReg \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引号
ptNewVal	GeoReg	[in]	更新的地理实体

### 3.69、GeoNet 对象

**描述** GeoNet 对象是地理网实体对象。GeoNet 对象包含了图形实体 Net 对象和实体属性数据。

#### 3.69.1、属性：

属性	描述
att Record 可读写	网络数据属性
count Long 只读	网络数据总项数
ndata (index) GeoNet 可读写	第 index 项网络数据（网段段号）
inf Net_Inf 可读写	网信息

#### 3.69.2、方法：

##### Append 方法

**描述** 添加网段。

Visual Basic GeoNet.Append netNo

C++ HRESULT pGeoNet -> Append(long netNo);

**参数说明：**

参数	类型	出口/入口	描述
netNo	Long	[in]	添加的网段号

##### Insert 方法

**描述** 在 index 处插入网段。

Visual Basic GeoNet.Insert index, netNo

C++ HRESULT pGeoNet-> Insert(long index, long netNo);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
netNo	Long	[in]	插入的网段号

### Remove 方法

描述 从 index 位置开始删除 nCount 个网段。

Visual Basic GeoNet.Remove index, nCount

C++ HRESULT pGeoNet-> Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
nCount	Long	[in]	删除的网段个数

### RemoveAll 方法

描述 删除所有的网段。

Visual Basic GeoNet.RemoveAll

C++ HRESULT pGeoNet-> RemoveAll ();

参数说明： 无

### Set 方法

描述 替换 GeoNet 网实体。

Visual Basic rtl = GeoNet.Set(ptNewVal)

C++ HRESULT pGeoNet -> Set(IGeoNet\*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	GeoNet	[in]	GeoNet 对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## 3.70、GeoNetSet 对象

描述 地理网实体集合对象 GeoNetSet 是 GeoReg 对象的集合，它封装了对地理区实体对象集合的操作。

### 3.70.1、属性：

属性	描述
----	----

count	GeoNet 实体数
Long	
只读	
item(index)	第 index 项 GeoNet 实体值
GeoNet	
只读	

### 3.70.2、方法：

#### Append 方法

描述 添加一个 GeoNet 实体。

Visual Basic GeoNetSet.Append geoNet

C++ HRESULT pGeoNetSet -> Append(IGeoNet \*geoNet);

参数说明：

参数	类型	出口/入口	描述
geoNet	GeoNet	[in]	添加的 GeoNet 实体

#### Insert 方法

描述 插入一个 GeoNet 实体。

Visual Basic GeoNetSet.Insert index, geoNet

C++ HRESULT pGeoNetSet -> Insert(long index, IGeoPath \* geoNet);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
geoNet	GeoNet	[in]	插入的 GeoNet 实体

#### Remove 方法

描述 从 index 位置开始删除 nCount 个 GeoNet 实体。

Visual Basic GeoNetSet.Remove index, nCount

C++ HRESULT pGeoNetSet -> Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
nCount	Long	[in]	删除的 GeoNet 个数

#### RemoveAll 方法

描述 删除所有的 GeoNet 实体。

Visual Basic GeoNetSet.RemoveAll

C++ HRESULT pGeoNetSet -> RemoveAll ();

参数说明： 无

### Update 方法

描述 更新第 index 项数据。

Visual Basic GeoNetSet.Update index, ptNewVal

C++ HRESULT pGeoNetSet-> Update (long index, IGeoNet \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引号
ptNewVal	GeoNet	[in]	更新的 GeoNet 实体

## 3.71、GeoPath 对象

描述 GeoPath 对象是地理路径对象。GeoPath 对象包含了图形实体 Path 对象和实体属性数据。

### 3.71.1、属性：

属性	描述
inf	线信息
Lin_Info	
可读写	
att	属性数据
Record	
可读写	
count_nod	结点数
Long	
只读	
count_arc	弧段数
Long	
只读	
item_nod(index)	第 index 项结点
Long	
可读写	
item_arc(index)	第 index 项弧段
Long	
可读写	



## 3.71.2、方法：

**Set 方法**

描述 替换 GeoPath 路径实体。

Visual Basic rtl = GeoPath.Set(ptNewVal)

C++ HRESULT pGeoPath -> Set(IGeoPath \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	GeoPath	[in]	GeoPath 对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**AppendNod 方法**

描述 添加一个结点。

Visual Basic GeoPath.AppendNod nodNo

C++ HRESULT pGeoPath -> AppendNod (long nodNo);

参数说明：

参数	类型	出口/入口	描述
nodNo	Long	[in]	添加的结点

**AppendArc 方法**

描述 添加一条弧段。

Visual Basic GeoPath.AppendArc arcNo

C++ HRESULT pGeoPath -> AppendArc (long arcNo);

参数说明：

参数	类型	出口/入口	描述
arcNo	Long	[in]	添加的弧段

**InsertNod 方法**

描述 插入一个结点。

Visual Basic GeoPath.InsertNod index, nodNo

C++ HRESULT pGeoPath-> InsertNod (long index, long nodNo);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
nodNo	Long	[in]	插入的结点

**InsertArc 方法**

描述 插入弧段。

Visual Basic GeoPath.InsertArc index, arcNo

C++ HRESULT pGeoPath-> InsertArc (long index, long arcNo);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
arcNo	Long	[in]	插入的弧段

**RemoveNod 方法**

描述 从 index 位置开始删除 nCount 个结点。

Visual Basic GeoPath.RemoveNod index, nCount

C++ HRESULT pGeoPath -> RemoveNod (long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
nCount	Long	[in]	删除的结点数

**RemoveArc 方法**

描述 从 index 位置开始删除 nCount 个弧段。

Visual Basic GeoPath.RemoveArc index, nCount

C++ HRESULT pGeoPath -> RemoveArc (long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
nCount	Long	[in]	删除的弧段数

**RemoveAllNod 方法**

描述 删除所有的结点。

Visual Basic GeoPath.RemoveAllNod

C++ HRESULT pGeoPath-> RemoveAllNod ();

参数说明： 无

**RemoveAllArc 方法**

描述 删除所有的弧段。

Visual Basic GeoPath.RemoveAllArc

C++ HRESULT pGeoPath-> RemoveAllArc ();

参数说明： 无

### 3.72、GeoPathSet 对象

**描述** 地理路径集合对象 GeoPathSet 是 GeoPath 对象的集合，它封装了对地理路径实体对象集合的操作。

#### 3.72.1、属性：

属性	描述
count Long 只读	路径数
item(index) GeoPath 只读	第 index 条路径

#### 3.72.2、方法：

##### Append 方法

**描述** 添加一个 GeoPath 实体。

Visual Basic GeoPathSet.Append geoPath

C++ HRESULT pGeoPathSet -> Append(IGeoPath \*geoPath);

参数说明：

参数	类型	出口/入口	描述
geoPath	GeoPath	[in]	添加的 GeoPath 实体

##### Insert 方法

**描述** 插入一个 GeoPath 实体。

Visual Basic GeoPathSet.Insert index, geoPath

C++ HRESULT p GeoPathSet -> Insert(long index, IGeoPath \*geoPath);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
geoPath	GeoPath	[in]	插入的 GeoPath 实体

##### Remove 方法

**描述** 从 index 位置开始删除 nCount 个 GeoPath 实体。

Visual Basic GeoPathSet.Remove index, nCount  
 C++ HRESULT pGeoPathSet -> Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
nCount	Long	[in]	删除的 GeoPath 个数

### RemoveAll 方法

描述 删除所有的 GeoPath 实体。

Visual Basic GeoPathSet.RemoveAll  
 C++ HRESULT pGeoPathSet -> RemoveAll ();

### Update 方法

描述 更新第 index 项数据。

Visual Basic GeoPathSet.Update index, ptNewVal  
 C++ HRESULT pGeoPathSet-> Update (long index, IGeoPath \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引号
ptNewVal	GeoPath	[in]	更新的 GeoPath 实体

## 3.73、LayerOnOffPad 对象

描述 LayerOnOffPad 对象是图层开关板对象。使用该对象可以控制工作区中的某一图层的打开、关闭状态。

### 3.73.1、属性：

属性	描述
item(index) Enum_Switch_Type 可读写	第 index 项开关状态
count Long 只读	图层开关板项数

### 3.73.2、方法：

**SetAllOn 方法**

描述 所有开关置为 ON。  
 Visual Basic LayerOnOffPad.SetAllOn  
 C++ HRESULT pLayerOnOffPad->SetAllOn();  
 参数说明： 无

**SetAllOff 方法**

描述 所有开关置为 OFF。  
 Visual Basic LayerOnOffPad.SetAllOff  
 C++ HRESULT pLayerOnOffPad->SetAllOff ();  
 参数说明： 无

**3.74、LayerPad 对象**

描述 LayerPad 对象是图层开关板对象，与 LayerOnOffPad 对象类似。

**3.74.1、属性：**

属性	描述
item(index) Enum_Switch_Type 可读写	第 index 项开关状态
count Long 只读	图层开关板项数

**3.74.2、方法：****SetAllOn 方法**

描述 所有开关置为 ON。  
 Visual Basic LayerPad.SetAllOn  
 C++ HRESULT pLayerPad->SetAllOn();  
 参数说明： 无

**SetAllOff 方法**

描述 所有开关置为 OFF。  
 Visual Basic LayerPad.SetAllOff  
 C++ HRESULT pLayerPad->SetAllOff ();

参数说明： 无

### 3.75、IDList 对象

描述 IDList 对象是实体号列表对象。

#### 3.75.1、属性：

属性	描述
item(index) Long 可读写	第 index 项实体号
count Long 只读	实体号列表项数

#### 3.75.2、方法：

##### Append 方法

描述 添加一个 ID。

Visual Basic IDList.Append id

C++ HRESULT pIDList-> Append(long id);

参数说明：

参数	类型	出口/入口	描述
id	Long	[in]	添加的 ID

##### Append2 方法

描述 添加一个 IDList 实体。

Visual Basic IDList.Append2 sourceLst

C++ HRESULT pIDList-> Append2 (IIDList \*sourceLst);

参数说明：

参数	类型	出口/入口	描述
sourceLst	IDList	[in]	添加的 IDList 实体

##### Insert 方法

描述 在 index 位置插入一个 ID。

Visual Basic IDList.Insert index, id

C++ HRESULT pIDList-> Insert(long index, long id);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
id	Long	[in]	插入的 ID

### Insert2 方法

描述 在 index 位置插入一个 IDList 实体。

Visual Basic IDList.Insert2 index, sourceLst

C++ HRESULT pIDList->Insert2(long index, IIDList \*sourceLst);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
sourceLst	IDList	[in]	插入的 IDList 实体

### Remove 方法

描述 从第 index 开始删除 nCount 个 ID。

Visual Basic IDList.Remove index, nCount

C++ HRESULT pIDList-> Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
nCount	Long	[in]	删除的 ID 个数

### RemoveAll 方法

描述 删除所有的 ID。

Visual Basic IDList.RemoveAll

C++ HRESULT pIDList-> RemoveAll ();

参数说明： 无

### Set 方法

描述 替换 IDList 值。

Visual Basic rtl = IDList.Set(ptNewVal)

C++ HRESULT pIDList -> Set (IIDList \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	IDList	[in]	IDList 实体对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## 3.76、LONGList 对象

**描述** LONGList 对象是长整型列表对象。

### 3.76.1、属性：

属性	描述
count	长整型数列表项数
Long 只读	
item(index)	第 index 项长整型数据
Long 可读写	

### 3.76.2、方法：

#### Append 方法

**描述** 添加一个长整型数据。

Visual Basic LONGList.Append lng

C++ HRESULT pLONGList-> Append(long lng);

**参数说明：**

参数	类型	出口/入口	描述
lng	Long	[in]	添加的长整型数据

#### Append2 方法

**描述** 添加一个 LONGList 实体。

Visual Basic LONGList.Append2 sourceLst

C++ HRESULT pLONGList-> Append2 (ILONGList \*sourceLst);

**参数说明：**

参数	类型	出口/入口	描述
sourceLst	LONGList	[in]	添加的 LONGList 实体

#### Insert 方法

**描述** 在 index 位置插入一个长整型数据。

Visual Basic LONGList.Insert index, lng

C++ HRESULT pLONGList-> Insert(short index, long lng);

**参数说明：**



参数	类型	出口/入口	描述
index	Integer	[in]	索引值
lng	Long	[in]	插入的长整型数据

**Insert2 方法**

描述 在 index 位置插入一个 LONGList 实体。

Visual Basic LONGList.Insert2 index, sourceLst

C++ HRESULT pLONGList->Insert2(long index, ILONGList \*sourceLst);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
sourceLst	LONGList	[in]	插入的 LONGList 实体

**Remove 方法**

描述 从第 index 开始删除 nCount 个长整型数据。

Visual Basic LONGList.Remove index, nCount

C++ HRESULT pLONGList-> Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
nCount	Long	[in]	删除的长整型数据个数

**RemoveAll 方法**

描述 删除所有长整型数据。

Visual Basic LONGList.RemoveAll

C++ HRESULT pLONGList-> RemoveAll ();

参数说明： 无

**Set 方法**

描述 替换 LONGList 值。

Visual Basic rtl = LONGList.Set(ptNewVal)

C++ HRESULT pLONGList -> Set (  
ILONGList \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	LONGList	[in]	LONGList 实体对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### 3.77、RepFldFlag 对象

**描述** RepFldFlag 对象是替换字段标志对象。

#### 3.77.1、属性：

属性	描述
count	字段开关项数
Integer 只读	
item(index)	第 index 项字段开关值
Integer 可读写	

#### 3.77.2、方法：

##### Append 方法

**描述** 添加一个字段开关。

Visual Basic RepFldFlag.Append flg

C++ HRESULT pRepFldFlag-> Append(short flg);

**参数说明：**

参数	类型	出口/入口	描述
flg	Integer	[in]	添加的字段开关

##### Insert 方法

**描述** 插入一个字段开关。

Visual Basic RepFldFlag.Insert index, flg

C++ HRESULT pRepFldFlag-> Insert(short index, short flg);

**参数说明：**

参数	类型	出口/入口	描述
index	Integer	[in]	索引值
flg	Integer	[in]	插入的字段开关

##### Remove 方法

**描述** 从 index 位置开始删除 nCount 项字段开关。

Visual Basic RepFldFlag.Remove index, nCount

C++ HRESULT pRepFldFlag-> Remove(short index, short nCount);

**参数说明：**

参数	类型	出口/入口	描述
index	Integer	[in]	索引值
nCount	Integer	[in]	删除的字段开关个数

#### RemoveAll 方法

描述 删除所有的字段开关。

Visual Basic RepFldFlag.RemoveAll

C++ HRESULT pRepFldFlag-> RemoveAll ();

参数说明： 无

### 3.78、Tic\_Dot 对象

描述 Tic\_Dot 对象是 Tic 点对象。

#### 3.78.1、属性：

属性	描述
xy0	图形坐标值
D_Dot 可读写	
xy1	理论坐标值
D_Dot 可读写	

#### 3.78.2、方法：

##### Set 方法

描述 替换 Tic\_Dot 点值。

Visual Basic rtl = TicDot.Set(ptNewVal)

C++ HRESULT pTicDot-> Set(ITic\_Dot \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Tic_Dot	[in]	Tic 点对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### 3.79、Tic\_DotSet 对象

**描述** Tic\_DotSet 对象是 Tic 点集对象。

#### 3.79.1、属性：

属性	描述
count	集合中的 Tic_Dot 数
Long 只读	
item(index)	第 index 项 Tic_Dot 实体
Tic_Dot 只读	

#### 3.79.2、方法：

##### Append 方法

**描述** 添加一个 Tic\_Dot。

Visual Basic TicDotSet.Append dot

C++ HRESULT pTicDotSet-> Append(ITic\_Dot \*dot);

**参数说明：**

参数	类型	出口/入口	描述
dot	Tic_Dot	[in]	Tic_Dot 点对象

##### Insert 方法

**描述** 在 index 位置插入一个 Tic\_Dot。

Visual Basic TicDotSet.Insert index, dot

C++ HRESULT pTicDotSet-> Insert(long index, ITic\_Dot \*dot);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	索引值
dot	Tic_Dot	[in]	Tic_Dot 点对象

##### Remove 方法

**描述** 从 index 位置开始删除 nCount 个 Tic\_Dot。

Visual Basic TicDotSet.Remove index, nCount

C++ HRESULT pTicDotSet-> Remove(long index, long nCount);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	索引值
nCount	Long	[in]	删除的 Tic_Dot 个数

**RemoveAll 方法**

描述 删除所有 Tic\_Dot。

Visual Basic TicDotSet.RemoveAll

C++ HRESULT pTicDotSet-> RemoveAll();

参数说明： 无

**Update 方法**

描述 更新第 index 项数据。

Visual Basic TicDotSet.Update index, ptNewVal

C++ HRESULT pTicDotSet-> Update(long index, ITic\_Dot \*ptNewVal);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
ptNewVal	Tic_Dot	[in]	更新的 Tic_Dot 点对象

**Set 方法**

描述 替换 Tic\_DotSet 点集实体。

Visual Basic rtl = TicDotSet.Set(ptNewVal)

C++ HRESULT pTicDotSet->Set(ITic\_DotSet \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	Tic_DotSet	[in]	Tic_Dot 点集对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**3.80、Ask\_Db\_Rcd 对象**

描述 Ask\_Db\_Rcd 对象是查询数据库记录对象。主要在 ExtDbTool 对象的 GetExtDbRcd 方法中使用。

**3.80.1、属性：**

属性	描述
----	----

dbfldName	数据库字段名
String	
可读写	
Val	字段值
Variant	
可读写	

### 3.80.2、方法：无

## 3.81、Ext\_Link\_Db\_Info 对象

描述 Ext\_Link\_Db\_Info 对象是外挂数据库连接信息对象。

### 3.81.1、属性：

属性	描述
keyFldType	外挂数据库关键字段类型
Enum_Field_Type	
可读写	
KeyFldName	外挂数据库关键字段名称
String	
可读写	
dbsfName	外挂数据库关键字段所在数据库
String	
可读写	

### 3.81.2、方法：无

## 3.82、Rad 对象

描述 Rad 对象是可变半径对象。

### 3.82.1、属性：

属性	描述
----	----

i	实体号
Long	
可读写	
r	半径
Double	
可读写	

### 3.82.2、方法：无

## 3.83、RadSet 对象

描述 RadSet 对象是可变半径集合对象。

### 3.83.1、属性：

属性	描述
item(index)	取第 index 项 Rad 可变半径
Tic_Dot	
只读	
count	集合中的 Rad 可变半径数
Long	
只读	

### 3.83.2、方法：

#### Append 方法

描述 添加一个 Rad 可变半径。

Visual Basic RadSet.Append rad

C++ HRESULT pRadSet-> Append(IRad \* rad);

参数说明：

参数	类型	出口/入口	描述
rad	Rad	[in]	添加的 Rad 对象

#### Insert 方法

描述 在 index 位置插入一个 Rad 可变半径。

Visual Basic RadSet.Insert index, rad

C++ HRESULT pRadSet-> Insert(long index, IRad \* rad);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
rad	Rad	[in]	插入的 Rad 对象

**Remove 方法**

描述 从 index 位置开始删除 nCount 个 Rad 可变半径。

Visual Basic RadSet.Remove index, nCount

C++ HRESULT pRadSet-> Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
nCount	Long	[in]	删除的 Rad 实体个数

**RemoveAll 方法**

描述 删除所有 Rad 可变半径。

Visual Basic RadSet.RemoveAll

C++ HRESULT pRadSet-> RemoveAll();

参数说明： 无

**Update 方法**

描述 更新第 index 项数据。

Visual Basic RadSet.Update index, rad

C++ HRESULT pRadSet-> Update(long index, IRad \*rad);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	索引值
rad	Rad	[in]	更新的 Rad 对象

**3.84、ExtDbTool 对象**

描述 ExtDbTool 对象封装对外挂数据库的操作。ExtDbTool 对象不可创建，但能被内部实例化。可以通过工作区对象的 extDB 属性或查询工作区对象的 IExtDbTool 接口来引用 ExtDbTool 对象。

**3.84.1、属性：无****3.84.2、方法：**



**SetExtDbFld 方法**

描述 设置外挂数据库字段。

Visual Basic rtl = ExtDbTool.SetExtDbFld(aType, fldName, keyFlg)

C++ HRESULT pExtDbTool->SetExtDbFld(Enum\_Entity\_Type aType,  
BSTR fldName, short keyFlg, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
aType	Enum_Entity_Type	[in]	属性类型
fldName	String	[in]	外挂数据库字段名
keyFlg	Integer	[in]	外挂标志；1=设置，0=取消
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ModLstRcdFldExtDb 方法**

描述 统改所有字段的外挂数据库号（仅作用于关键字段）

Visual Basic rtl = ExtDbTool.ModLstRcdFldExtDb(aType, lst, fldI, dbNo)

C++ HRESULT pExtDbTool->ModLstRcdFldExtDb(Enum\_Entity\_Type aType,  
IIDList \*lst, short fldI, short dbNo, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
aType	Enum_Entity_Type	[in]	属性类型
lst	IDList	[in]	记录号列表；lst 为空则统改所有记录；lst[0]为记录数，lst[1]开始为记录号
fldI	Integer	[in]	字段号
dbNo	Integer	[in]	外挂数据库号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Get\_AreaExtDb\_LinkFlag 方法**

描述 取工作区与外挂数据之间的连接标志（0/1=松散/整体）。

整体连接关系：即外挂数据作为父工作区的一部分看待，外挂数据的任何改变被看作父工作区数据的改变。

Visual Basic flg = ExtDbTool.Get\_AreaExtDb\_LinkFlag()

C++ HRESULT pExtDbTool->Get\_AreaExtDb\_LinkFlag(short \*flg);

参数说明：

参数	类型	出口/入口	描述
flg	Integer	[out]	返工作区与回外挂数据之间的连接标志

**Set\_AreaExtDb\_LinkFlag 方法**

描述 设工作区与外挂数据之间的连接标志（0/1=松散/整体）。

Visual Basic preFlg = ExtDbTool.Set\_AreaExtDb\_LinkFlag(flgl)

C++ HRESULT pExtDbTool->Set\_AreaExtDb\_LinkFlag(short flgl, short \* preFlgl);

参数说明：

参数	类型	出口/入口	描述
flgl	Integer	[in]	连接标志
preFlgl	Integer	[out]	返回以前的状态

**SetExtDbLinkInfo 方法**

描述 设置外挂数据库连接信息。

Visual Basic dbNo = ExtDbTool.SetExtDbLinkInfo(dbInfl)

C++ HRESULT pExtDbTool->SetExtDbLinkInfo(  
IExt\_Link\_Db\_Info \*dbInfl, short \*dbNo);

参数说明：

参数	类型	出口/入口	描述
dbInfl	Ext_Link_Db_Info	[in]	外挂数据库信息
dbNo	Integer	[out]	失败返回 0；成功返回外挂数据库编号（dbNo>0），此项编号由系统内部分配，保证同一个数据（或工作区）中所有外挂数据库项有不同的编号

**GetExtDbNum 方法**

描述 外挂数据库信息项数。

Visual Basic num = ExtDbTool.GetExtDbNum()

C++ HRESULT pExtDbTool->GetExtDbNum(short \*num);

参数说明：

参数	类型	出口/入口	描述
num	Integer	[out]	返回外挂数据库信息项数

**GetExtDbLinkInfo0 方法**

描述 取外挂数据库连接信息。

Visual Basic rtl = ExtDbTool.GetExtDbLinkInfo0(seqNo, dbInfl)

C++ HRESULT pExtDbTool->GetExtDbLinkInfo0(  
short seqNo, IExt\_Link\_Db\_Info \*\*dbInfl, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

seqNo	Integer	[in]	外挂数据库序号
dbInf	Ext_Link_Db_Info	[out]	外挂数据库信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetExtDbLinkInfo 方法**

描述 取外挂数据库连接信息。

Visual Basic rtl = ExtDbTool.GetExtDbLinkInfo(dbNo, dbInf)

C++ HRESULT pExtDbTool->GetExtDbLinkInfo (  
short dbNo, IExt\_Link\_Db\_Info \*\*dbInf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dbNo	Integer	[in]	数据库编号
dbInf	Ext_Link_Db_Info	[out]	外挂数据库信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UpdateExtDbLinkInfo0 方法**

描述 根据外挂库序号更新外挂数据库连接信息。

Visual Basic rtl = ExtDbTool.UpdateExtDbLinkInfo0(seqNo, dbInf)

C++ HRESULT pExtDbTool->UpdateExtDbLinkInfo0(  
short seqNo, IExt\_Link\_Db\_Info \*dbInf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
seqNo	Integer	[in]	外挂数据库序号
dbInf	Ext_Link_Db_Info	[in]	外挂数据库信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UpdateExtDbLinkInfo 方法**

描述 根据外挂数据库编号更新外挂数据库连接信息。

Visual Basic rtl = ExtDbTool.UpdateExtDbLinkInfo(dbNo, dbInf)

C++ HRESULT pExtDbTool->UpdateExtDbLinkInfo (  
short dbNo, IExt\_Link\_Db\_Info \*dbInf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dbNo	Integer	[in]	数据库编号
dbInf	Ext_Link_Db_Info	[in]	外挂数据库信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SelExtDataBase 方法**

描述 选择外挂数据库。

Visual Basic rtl = ExtDbTool.SelExtDataBase(dbNo, dbInf, ttlName)

C++ HRESULT pExtDbTool->SelExtDataBase(short \*dbNo,  
IExt\_Link\_Db\_Info \*\*dbInf, BSTR ttlName, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dbNo	Integer	[out]	数据库编号
dbInf	Ext_Link_Db_Info	[out]	外挂数据库信息
ttlName	String	[in]	对话框标题字符串
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CopyExtDbInfo 方法**

描述 拷贝外挂数据库信息表。

Visual Basic rtl = ExtDbTool.CopyExtDbInfo(ai1)

C++ HRESULT pExtDbTool->CopyExtDbInfo(  
IWorkArea \*ai1, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai1	WorkArea	[in]	目的工作区号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**OpenExtDataBase 方法**

描述 打开外部数据库。

Visual Basic rtl = ExtDbTool.OpenExtDataBase(dbNo, ai)

C++ HRESULT pExtDbTool->OpenExtDataBase(short dbNo,  
ITblArea \*\*ai, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dbNo	Integer	[in]	数据库编号
ai	TblArea	[out]	外挂数据库工作区对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SaveAllExtDataBase 方法**

描述 保存所有的外挂数据。

Visual Basic rtl = ExtDbTool.SaveAllExtDataBase()

C++ HRESULT pExtDbTool->SaveAllExtDataBase(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### SetExtDbUserClass0 方法

描述 设置外部数据库信息表某项的用户分类编码。

Visual Basic rtl = ExtDbTool.SetExtDbUserClass0(seqNo, useCls)

C++ HRESULT pExtDbTool->SetExtDbUserClass0(  
short seqNo, short useCls, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
seqNo	Integer	[in]	外挂库顺序号，>=0 且<GetExtDbNum 的返回值
useCls	Integer	[in]	用户分类编码
rtl	Integer	[out]	成功返回 1；失败返回 0；返回值<0 表示与 dbNo=返回值的绝对值的外挂库有相同的用户分类编码

#### SetExtDbUserClass 方法

描述 设置外部数据库信息表某项的用户分类编码。

Visual Basic rtl = ExtDbTool.SetExtDbUserClass(dbNo, useCls)

C++ HRESULT pExtDbTool->SetExtDbUserClass(  
short dbNo, short useCls, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
dbNo	Integer	[in]	外挂数据库编号，SetExtDbLinkInfo 函数的返回值
useCls	Integer	[in]	用户分类编码
rtl	Integer	[out]	成功返回 1；失败返回 0；返回值<0 表示与 dbNo=返回值的绝对值的外挂库有相同的用户分类编码

#### GetExtDbUserClass0 方法

描述 取外部数据库信息表某项的用户分类编码。

Visual Basic rtl = ExtDbTool.GetExtDbUserClass0(seqNo, useCls)

C++ HRESULT pExtDbTool->GetExtDbUserClass0(  
short seqNo, short \*useCls, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
seqNo	Integer	[in]	外挂库顺序号，>=0 且<GetExtDbNum 的返回值
useCls	Integer	[out]	用户分类编码
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetExtDbUserClass 方法

描述 取外部数据库信息表某项的用户分类编码。

Visual Basic rtl = ExtDbTool.GetExtDbUserClass(dbNo, useCls)

C++ HRESULT pExtDbTool->GetExtDbUserClass(  
short dbNo, short \*useCls, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dbNo	Integer	[in]	外挂数据库编号，SetExtDbLinkInfo 函数的返回值
useCls	Integer	[out]	用户分类编码
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetExtDbNo0 方法

描述 取外部数据库信息表某项的数据库编码。

Visual Basic rtl = ExtDbTool.GetExtDbNo0(seqNo, dbNo)

C++ HRESULT pExtDbTool->GetExtDbNo0(  
short seqNo, short \*dbNo, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
seqNo	Integer	[in]	外挂库顺序号
dbNo	Integer	[out]	数据库编号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetExtDbNo1 方法

描述 取外部数据库信息表某项的数据库编码。

Visual Basic rtl = ExtDbTool.GetExtDbNo1(useCls, dbNo)

C++ HRESULT pExtDbTool->GetExtDbNo1(  
short useCls, short \*dbNo, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

useCls	Integer	[in]	用户分类编码
dbNo	Integer	[out]	数据库编号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetExtDbAi0 方法**

描述 取外部数据库对应的工作区号。

Visual Basic rtl = ExtDbTool.GetExtDbAi0(seqNo, dbAi)

C++ HRESULT pExtDbTool->GetExtDbAi0(  
short seqNo, ITblArea \*\* dbAi, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
seqNo	Integer	[in]	外挂库顺序号
dbAi	TblArea	[out]	外挂数据库对应的工作区号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetExtDbAi1 方法**

描述 取外部数据库对应的工作区号。

Visual Basic rtl = ExtDbTool.GetExtDbAi1(useCls, dbAi)

C++ HRESULT pExtDbTool->GetExtDbAi1(  
short useCls, ITblArea \*\* dbAi, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
useCls	Integer	[in]	用户分类编码
dbAi	TblArea	[out]	外挂数据库对应的工作区号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetExtDbAi 方法**

描述 取外部数据库对应的工作区号。

Visual Basic rtl = ExtDbTool.GetExtDbAi(dbNo, dbAi)

C++ HRESULT pExtDbTool->GetExtDbAi(  
short dbNo, ITblArea \*\* dbAi, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dbNo	Integer	[in]	外挂数据库编号
dbAi	TblArea	[out]	外挂数据库对应的工作区号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetExtDbRcd 方法**

**描述** 根据 ASK\_DB\_RCD 结构信息取外挂数据库记录。

**Visual Basic** seqNo = ExtDbTool.GetExtDbRcd(askDbRcd, att)

**C++** HRESULT pExtDbTool->GetExtDbRcd(  
IAsk\_Db\_Rcd \*askDbRcd, IRecord \*\*att, long \*seqNo);

**参数说明：**

参数	类型	出口/入口	描述
askDbRcd	Ask_Db_Rcd	[in]	查找数据库记录结构对象
att	Record	[out]	属性记录
seqNo	Long	[out]	成功返回记录物理序号 (>0), 失败返回 0

**GetExtDbRcd1 方法**

**描述** 取工作区中 aType 类型属性的 rcdi 号记录的 fldName 字段对应的外挂数据库记录。

**Visual Basic** seqNo = ExtDbTool.GetExtDbRcd1(aType, rcdi, fldName, att)

**C++** HRESULT pExtDbTool->GetExtDbRcd1(Enum\_Entity\_Type aType,  
long rcdi, BSTR fldName, IRecord \*\*att, long \*seqNo);

**参数说明：**

参数	类型	出口/入口	描述
aType	Enum_Entity_Type	[in]	属性类型
rcdi	Long	[in]	记录号
fldName	String	[in]	外挂数据库的字段名
att	Record	[out]	属性记录
seqNo	Long	[out]	成功返回记录物理序号 (>0), 失败返回 0

**GetFldExtDbNo 方法**

**描述** 根据字段序号取外挂数据库号。

**Visual Basic** rtl = ExtDbTool.GetFldExtDbNo(att, fldI, type, dbNo)

**C++** HRESULT pExtDbTool->GetFldExtDbNo(IRecord \*att,  
short fldI, Enum\_Field\_Type \*type, short \*dbNo, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
att	Record	[in]	属性记录数据
fldI	Integer	[in]	字段序号, 第一个字段序号为 0
type	Enum_Field_Type	[out]	fldI 号字段类型
dbNo	Integer	[out]	外挂数据库号
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE



**SetFldExtDbNo 方法**

描述 根据字段序号设置外挂数据库号。

Visual Basic rtl = ExtDbTool.SetFldExtDbNo(att, fldI, dbNo)

C++ HRESULT pExtDbTool->SetFldExtDbNo (  
IRecord \*att, short fldI, short dbNo, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
att	Record	[in]	属性记录数据
fldI	Integer	[in]	字段序号，从 0 开始第一个字段序号为 0
dbNo	Integer	[out]	外挂数据库号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**3.85、Layer 对象**

描述 图层对象 Layer 封装了操作图层的方法。Layer 对象不可创建，但能被内部实例化。可以通过工作区对象的 layer 属性或查询工作区对象的 ILayer 接口来引用 Layer 对象。

**3.85.1、属性：无****3.85.2、方法：****GetLayerPad 方法**

描述 取工作区图层开关板。

Visual Basic Set layPad = Layer.GetLayerPad()

C++ HRESULT pLayer->GetLayerPad(ILayerPad \*\*layPad);

参数说明：

参数	类型	出口/入口	描述
layPad	LayerPad	[out]	返回图层板对象

**GetLayerOnOffPad 方法**

描述 取工作区图层开关板。

Visual Basic Set layOnOffPad = Layer.GetLayerOnOffPad()

C++ HRESULT pLayer->GetLayerOnOffPad(ILayerOnOffPad \*\*layOnOffPad);

参数说明：

参数	类型	出口/入口	描述
layOnOffPad	LayerOnOffPad	[out]	返回图层开关板对象

**ReSetRangeAndLayer 方法**

描述 重新设置图形范围和图层号。

Visual Basic rtl = Layer.ReSetRangeAndLayer()

C++ HRESULT pLayer->ReSetRangeAndLayer(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ModLayer 方法**

描述 统改图层号。

Visual Basic rtl = Layer.ModLayer(layer0, layer1)

C++ HRESULT pLayer->ModLayer(  
short layer0, short layer1, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
layer0	Integer	[in]	原始图层号
layer1	Integer	[in]	新的图层号
Rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ModLayer1 方法**

描述 统改图层号（弹出对话框用以选择图层）。

Visual Basic rtl = Layer.ModLayer1()

C++ HRESULT player->ModLayer1(VARIANT\_BOOL \*rtl)

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UpdateLayer 方法**

描述 更新图层号。

Visual Basic rtl = Layer.UpdateLayer(i, layer)

C++ HRESULT pLayer->UpdateLayer(long i, short layer, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
i	Long	[in]	图元号
layer	Integer	[in]	新的层号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UpdateLayer1 方法**

描述 更新图层号（弹出对话框用以选择图层）。

Visual Basic rtl = Layer.UpdateLayer1(i)

C++ HRESULT pLayer->UpdateLayer1(long i, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
i	Long	[in]	图元号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SetCurLayer 方法**

描述 设置当前层。

Visual Basic rtl = Layer.SetCurLayer(layer)

C++ HRESULT pLayer->SetCurLayer(short layer, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
layer	Integer	[in]	层号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SelCurLayer 方法**

描述 选择当前层（弹出对话框以选择当前层）。

Visual Basic rtl = Layer.SelCurLayer()

C++ HRESULT pLayer->SelCurLayer(short \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Integer	[out]	成功返回 1，失败返回 0

**SelLayer 方法**

描述 选择层（弹出对话框以选择层）。

Visual Basic rtl = Layer.SelLayer(titleStr, layer)

C++ HRESULT pLayer->SelLayer(  
BSTR ttlStr, short \*layer, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
titleStr	String	[in]	窗口标题，可以为空
layer	Integer	[out]	图层号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetCurLayer 方法**

描述 取当前层号。

Visual Basic rtl = Layer.GetCurLayer(curLay)

C++ HRESULT pLayer->GetCurLayer(short \*curLay, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
curLay	Integer	[out]	当前层号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetCurLayerName 方法**

描述 取当前图层名称。

Visual Basic rtl = Layer.GetCurLayerName(layName)

C++ HRESULT pLayer->GetCurLayerName(  
BSTR \*layName, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
layName	String	[out]	当前层名
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SaveCurLayer 方法**

描述 保存当前层图元。

Visual Basic rtl = Layer.SaveCurLayer()

C++ HRESULT pLayer->SaveCurLayer(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SaveALayer 方法**

描述 存某层图元。

Visual Basic rtl = Layer.SaveALayer(layer)

C++ HRESULT pLayer->SaveALayer(short layer, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
layer	Integer	[in]	层号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DelCurLayer 方法**

描述 删除当前层图元。

Visual Basic rtl = Layer.DelCurLayer()

C++ HRESULT player->DelCurLayer(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DelALayer 方法**

描述 删除某层图元。

Visual Basic rtl = Layer.DelALayer(layer)

C++ HRESULT pLayer->DelALayer(short layer, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
layer	Integer	[in]	图层号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**TurnLayerSwitch 方法**

描述 改层开关。

Visual Basic rtl = Layer.TurnLayerSwitch()

C++ HRESULT player->TurnLayerSwitch(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**TurnOnOffAllLayer 方法**

描述 开或关所有层。

Visual Basic rtl = Layer.TurnOnOffAllLayer(onoff)

C++ HRESULT pLayer->TurnOnOffAllLayer(short onoff, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
onoff	Integer	[in]	开关状态，1/0：打开/关闭
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**TurnOnOffLayer 方法**

描述 开或关某一层。

Visual Basic rtl = Layer.TurnOnOffLayer(layer, onoff)

C++ HRESULT pLayer->TurnOnOffLayer(  
short layer, short onoff, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
layer	Integer	[in]	层号
onoff	Integer	[in]	开关状态，1/0：打开/关闭
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetLayerOnOff 方法

描述 取某一层开关状态。

Visual Basic onoff = Layer.GetLayerOnOff(layer)

C++ HRESULT pLayer->GetLayerOnOff (short layer, short \*onoff);

参数说明：

参数	类型	出口/入口	描述
layer	Integer	[in]	层号
onoff	Integer	[out]	返回开关状态，1/0：打开/关闭

### RenameLayer 方法

描述 修改层名（弹出模板）。

Visual Basic rtl = Layer.RenameLayer()

C++ HRESULT pLayer->RenameLayer(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### SetLayerNameLst 方法

描述 设置层名表。

Visual Basic rtl = Layer.SetLayerNameLst()

C++ HRESULT pLayer->SetLayerNameLst(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### IsALayHasElement 方法

描述 判断某一层是否有图元。

Visual Basic HasElement = Layer.IsALayHasElement(layer)

C++ HRESULT pLayer->IsALayHasElement(short layer, short \*HasElement);

参数说明：

参数	类型	出口/入口	描述
layer	Integer	[in]	图层号
HasElement	Integer	[out]	有图元返回值 1,没有图元返回值 0

**ShowLaysOfElement 方法**

描述 显示有图元的层。

Visual Basic rtl = Layer.ShowLaysOfElement()

C++ HRESULT pLayer-&gt;ShowLaysOfElement(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**3.86、PntAtt 对象**

描述 点属性对象 PntAtt 封装了操作点属性的方法。PntAtt 对象不可创建，但能被内部实例化。可以通过点工作区对象的 att 属性或查询点工作区对象的 IPntAtt 接口来引用 PntAtt 对象。

**3.86.1、属性：**

属性	描述
stru	点属性结构；通过此属性可以操作点属性结构
Record_Head	
可读写	

**3.86.2、方法：****Get 方法**

描述 取点 pi 的属性。

Visual Basic rtl = PntAtt.Get(pi, att)

C++ HRESULT pPntAtt-&gt;Get(long pi, IRecord \*\*att, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	点实体号
att	Record	[out]	属性记录值
rtl	Integer	[out]	返回值：1=成功、2=空属性记录、0=失败、-1=该属性对应的图元已经被删除

**GetDef 方法**

描述 取缺省属性记录。

Visual Basic rtl = PntAtt.GetDef(att)

C++ HRESULT pPntAtt->GetDef(IRecord \*\*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
att	Record	[out]	缺省属性记录值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Write 方法**

描述 写属性。

Visual Basic rtl = PntAtt.Write(pnti, att)

C++ HRESULT pPntAtt->Write(  
long pnti, IRecord \*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pnti	Long	[in]	点实体号
att	Record	[in]	属性记录值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**WriteDef 方法**

描述 设置缺省属性记录值。

Visual Basic rtl = PntAtt.WriteDef(att)

C++ HRESULT pPntAtt->WriteDef(IRecord \*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
att	Record	[in]	缺省属性记录值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetID 方法**

描述 取属性 ID。

Visual Basic pid = PntAtt.GetID(pi)

C++ HRESULT pPntAtt->GetID(long pi, long \*pid);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	点实体号
pid	Long	[out]	返回属性 ID



**GetMaxID 方法**

描述 取最大属性 ID 值。

Visual Basic max\_id\_val = PntAtt.GetMaxID()

C++ HRESULT pPntAtt->GetMaxID(long \*max\_id\_val);

参数说明：

参数	类型	出口/入口	描述
max_id_val	Long	[out]	返回最大属性 ID

**CopyAll 方法**

描述 拷贝所有属性记录到表工作区 ai1 中。

Visual Basic attnum = PntAtt.CopyAll(ai1)

C++ HRESULT pPntAtt->CopyAll(ITblArea \*ai1, long \*attnum);

参数说明：

参数	类型	出口/入口	描述
ai1	TblArea	[in]	目的工作区
attnum	Long	[out]	拷贝的记录数

**CopyList 方法**

描述 拷贝 lst 指定的记录到表工作区 ai1 中。

Visual Basic attnum = PntAtt.CopyList(lst, ai1, [changeStru], [rtnNo])

C++ HRESULT pPntAtt->CopyList(IDList \*lst, ITblArea \*ai1,  
short changeStru=1, short rtnNo=0, long \*attnum);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	属性记录列表
ai1	TblArea	[in]	目的工作区
changeStru	Integer	[in]	是否改变 ai1 的属性结构，使其和 PntAtt 的结构相同。 可选；缺省值：1
rtnNo	Integer	[in]	是否通过 lst 返回新的记录号。 可选；缺省值：0
attnum	Long	[out]	拷贝的记录数

**RepLstRcdFld0 方法**

描述 统改属性记录字段值。

Visual Basic nCount = PntAtt.RepLstRcdFld0(lst, repFldFlg, attExp, grpPad, grpInf, repAtt)

C++ HRESULT pPntAtt->RepLstRcdFld0(  
IIDList \*lst, IRepFldFlag \*pRepFldFlg, BSTR attExp,  
IPnt\_Pad \*grpPad, IPnt\_Info \*inf, IRecord \*repAtt, long \*nCount);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	将进行替换的记录号列表对象，若 lst 为空则统改所有记录
repFldFlg	RepFldFlag	[in]	替换字段开关；repFldFlag[i]对应 i（从 0 开始）号字段，1=替换，0=不替换
attExp	String	[in]	属性条件表达式字符串；可为空，即无属性条件
grpPad	Pnt_Pad	[in]	图形参数开关，可为空
grpInf	Pnt_Info	[in]	图形参数，可为空
repAtt	Record	[in]	替换值记录数据
nCount	Long	[out]	成功返回统改的记录数，失败返回 0

### RepLstRcdFld 方法

描述 字段统改（弹出统改属性值条件窗口）。

Visual Basic rtl = PntAtt.RepLstRcdFld(lst, ttlName)

C++ HRESULT pPntAtt->RepLstRcdFld(  
IIDList \*lst, BSTR ttlName, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	统改记录号表，若为空则统改所有记录
ttlName	String	[in]	统改属性值条件窗口标题
rtl	Integer	[out]	成功返回 1，失败返回 0

### ModLstRcdFldVal 方法

描述 统改字段域值。

Visual Basic rtl = PntAtt.ModLstRcdFldVal(lst, fldI, fldVal)

C++ HRESULT pPntAtt->ModLstRcdFldVal(  
IIDList \*lst, short fldI, VARIANT fldVal, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	统改记录号表，若为空则统改所有记录
fldI	Integer	[in]	统改字段号
fldVal	Variant	[in]	字段域值

rtl	Integer	[out]	成功返回 1，失败返回 0
-----	---------	-------	---------------

**CalculateFldVal 方法**

描述 计算字段值。

Visual Basic rtl = PntAtt.CalculateFldVal(fldName, calExp, domainErrVal)

C++ HRESULT pPntAtt->CalculateFldVal(  
BSTR fldName,BSTR calExp,  
VARIANT domainErrVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fldName	String	[in]	计算字段名称
calExp	String	[in]	计算表达式
domainErrVal	Variant	[in]	表达式计算过程中定义域错时（如除 0 错），计算结果替换值，若 domainErrVal 不为空,则用 domainErrVal 作为 fldName 的字段值；若为空，则不修改当前记录的 fldName 字段值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**3.87、LinAtt 对象**

描述 线属性对象 LinAtt 封装了操作线属性的方法。LinAtt 对象不可创建，但能被内部实例化。可以通过线工作区对象的 att 属性，也可以通过查询线、区、网工作区对象的 ILinAtt 接口来引用 LinAtt 对象。

**3.87.1、属性：**

属性	描述
stru	线属性结构；通过此属性可以操作线属性结构
Record_Head	
可读写	

**3.87.2、方法：****Get 方法**

描述 取线 li 的属性。

Visual Basic rtl = LinAtt.Get(li, att)

C++ HRESULT pLinAtt->Get(

```
long li, IRecord **att, short *rtl);
```

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
att	Record	[out]	属性记录值
rtl	Integer	[out]	返回值：1=成功、2=空属性记录、0=失败、-1=该属性对应的图元已经被删除

### GetDef 方法

描述 取缺省属性记录。

Visual Basic rtl = LinAtt.GetDef(att)

C++ HRESULT pLinAtt->GetDef(IRecord \*\*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
att	Record	[out]	缺省属性记录值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### Write 方法

描述 写属性。

Visual Basic rtl = LinAtt.Write(li, att)

C++ HRESULT pLinAtt->Write(  
long li, IRecord \*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
att	Record	[in]	属性记录值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### WriteDef 方法

描述 设置缺省属性记录值。

Visual Basic rtl = LinAtt.WriteDef(att)

C++ HRESULT pLinAtt->WriteDef(IRecord \*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
att	Record	[in]	缺省属性记录值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetID 方法**

描述 取属性 ID。

Visual Basic lid = LinAtt.GetID(li)

C++ HRESULT pLinAtt->GetID(long li, long \*lid);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
lid	Long	[out]	返回属性 ID

**GetMaxID 方法**

描述 取最大属性 ID 值。

Visual Basic max\_id\_val = LinAtt.GetMaxID()

C++ HRESULT pLinAtt->GetMaxID(long \*max\_id\_val);

参数说明：

参数	类型	出口/入口	描述
max_id_val	Long	[out]	返回最大属性 ID

**CopyAll 方法**

描述 拷贝所有属性记录到表工作区 ai1 中。

Visual Basic attnum = LinAtt.CopyAll(ai1)

C++ HRESULT pLinAtt->CopyAll(ITblArea \*ai1, long \*attnum);

参数说明：

参数	类型	出口/入口	描述
ai1	TblArea	[in]	目的工作区
attnum	Long	[out]	拷贝的记录数

**CopyList 方法**

描述 拷贝 lst 指定的记录到表工作区 ai1 中。

Visual Basic attnum = LinAtt.CopyList(lst, ai1, [changeStru], [rtnNo])

C++ HRESULT pLinAtt->CopyList(IIDList \*lst, ITblArea \*ai1,  
short changeStru=1, short rtnNo=0, long \*attnum);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	属性记录列表
ai1	TblArea	[in]	目的工作区

changeStru	Integer	[in]	是否改变 ai1 的属性结构，使其和 LinAtt 的结构相同。 可选；缺省值：1
rtnNo	Integer	[in]	是否通过 lst 返回新的记录号。 可选；缺省值：0
attnum	Long	[out]	拷贝的记录数

**RepLstRcdFld0 方法**

描述 统改属性记录字段值。

Visual Basic nCount = LinAtt.RepLstRcdFld0(lst, repFldFlg, attExp, grpPad, grpInf, repAtt)

C++ HRESULT pLinAtt->RepLstRcdFld0(  
IIDList \*lst, IRepFldFlag \*pRepFldFlg, BSTR attExp,  
ILin\_Pad \*grpPad, ILin\_Info \*grpInf, IRecord \*repAtt, long \*nCount);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	将进行替换的记录号列表对象，若 lst 为空则统改所有记录
repFldFlg	RepFldFlag	[in]	替换字段开关；repFldFlag[i]对应 i（从 0 开始）号字段，1=替换，0=不替换
attExp	String	[in]	属性条件表达式字符串；可为空，即无属性条件
grpPad	Lin_Pad	[in]	图形参数开关，可为空
grpInf	Lin_Info	[in]	图形参数，可为空
repAtt	Record	[in]	替换值记录数据
nCount	Long	[out]	成功返回统改的记录数，失败返回 0

**RepLstRcdFld 方法**

描述 字段统改（弹出统改属性值条件窗口）。

Visual Basic rtl = LinAtt.RepLstRcdFld(lst, ttlName)

C++ HRESULT pLinAtt->RepLstRcdFld(  
IIDList \*lst, BSTR ttlName, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	统改记录号表，若为空则统改所有记录
ttlName	String	[in]	统改属性值条件窗口标题
rtl	Integer	[out]	成功返回 1，失败返回 0

**ModLstRcdFldVal 方法**

描述 统改字段域值。

Visual Basic rtl = LinAtt.ModLstRcdFldVal(lst, fldI, fldVal)

C++ HRESULT pLinAtt->ModLstRcdFldVal(  
IIDList \*lst, short fldI, VARIANT fldVal, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	统改记录号表，若为空则统改所有记录
fldI	Integer	[in]	统改字段号
fldVal	Variant	[in]	字段域值
rtl	Integer	[out]	成功返回 1，失败返回 0

**CalculateFldVal 方法**

描述 计算字段值。

Visual Basic rtl = LinAtt.CalculateFldVal(fldName, calExp, domainErrVal)

C++ HRESULT pLinAtt->CalculateFldVal(  
BSTR fldName,BSTR calExp,  
VARIANT domainErrVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fldName	String	[in]	计算字段名称
calExp	String	[in]	计算表达式
domainErrVal	Variant	[in]	表达式计算过程中定义域错时（如除 0 错），计算结果替换值，若 domainErrVal 不为空，则用 domainErrVal 作为 fldName 的字段值；若为空，则不修改当前记录的 fldName 字段值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**3.88、RegAtt 对象**

描述 区属性对象 RegAtt 封装了操作区属性的方法。RegAtt 对象不可创建，但能被内部实例化。可以通过区工作区对象的 att 属性或通过查询区工作区对象的 IRegAtt 接口来引用 RegAtt 对象。

**3.88.1、属性：**

属性	描述
----	----

stru	区属性结构；通过此属性可以操作区属性结构
Record_Head	
可读写	

### 3.88.2、方法：

#### Get 方法

描述 取区 ri 的属性。

Visual Basic rtl = RegAtt.Get(ri, att)

C++ HRESULT pRegAtt->Get(  
long ri, IRecord \*\*att, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	区实体号
att	Record	[out]	属性记录值
rtl	Integer	[out]	返回值：1=成功、2=空属性记录、0=失败、-1=该属性对应的图元已经被删除

#### GetDef 方法

描述 取缺省属性记录。

Visual Basic rtl = RegAtt.GetDef(att)

C++ HRESULT pRegAtt->GetDef(IRecord \*\*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
att	Record	[out]	缺省属性记录值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### Write 方法

描述 写属性。

Visual Basic rtl = RegAtt.Write(ri, att)

C++ HRESULT pRegAtt->Write(  
long ri, IRecord \*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	区实体号
att	Record	[in]	属性记录值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE



**WriteDef 方法**

描述 设置缺省属性记录值。

Visual Basic `rtl = RegAtt.WriteDef(att)`

C++ `HRESULT pRegAtt->WriteDef(IRecord *att, VARIANT_BOOL *rtl);`

参数说明：

参数	类型	出口/入口	描述
att	Record	[in]	缺省属性记录值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetID 方法**

描述 取属性 ID。

Visual Basic `lid = RegAtt.GetID(ri)`

C++ `HRESULT pRegAtt->GetID(long ri, long *rid);`

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	区实体号
rid	Long	[out]	返回属性 ID

**GetMaxID 方法**

描述 取最大属性 ID 值。

Visual Basic `max_id_val = RegAtt.GetMaxID()`

C++ `HRESULT pRegAtt->GetMaxID(long *max_id_val);`

参数说明：

参数	类型	出口/入口	描述
max_id_val	Long	[out]	返回最大属性 ID

**CopyAll 方法**

描述 拷贝所有属性记录到表工作区 ai1 中。

Visual Basic `attnum = RegAtt.CopyAll(ai1)`

C++ `HRESULT pRegAtt->CopyAll(ITblArea *ai1, long *attnum);`

参数说明：

参数	类型	出口/入口	描述
ai1	TblArea	[in]	目的工作区
attnum	Long	[out]	拷贝的记录数

**CopyList 方法**

描述 拷贝 lst 指定的记录到表工作区 ai1 中。

Visual Basic attnum = RegAtt.CopyList(lst, ai1, [changeStru], [rtnNo])  
C++ HRESULT pRegAtt->CopyList(  
IIDList \*lst, ITblArea \*ai1, short changeStru=1, short rtnNo=0, long \*attnum);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	属性记录列表
ai1	TblArea	[in]	目的工作区
changeStru	Integer	[in]	是否改变 ai1 的属性结构,使其和 RegAtt 的结构相同。 可选;缺省值:1
rtnNo	Integer	[in]	是否通过 lst 返回新的记录号。 可选;缺省值:0
attnum	Long	[out]	拷贝的记录数

### RepLstRcdFld 方法

描述 统改属性记录字段值。

Visual Basic nCount = RegAtt.RepLstRcdFld0(lst, repFldFlg, attExp, grpPad, grpInf, repAtt)  
C++ HRESULT pRegAtt->RepLstRcdFld0(  
IIDList \*lst, IRepFldFlag \*pRepFldFlg, BSTR attExp,  
IReg\_Pad \*grpPad, IReg\_Info \*grpInf, IRecord \*repAtt, long \*nCount);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	将进行替换的记录号列表对象,若 lst 为空则统改所有记录
repFldFlg	RepFldFlag	[in]	替换字段开关;repFldFlag[i]对应 i (从 0 开始) 号字段,1=替换,0=不替换
attExp	String	[in]	属性条件表达式字符串;可为空,即无属性条件
grpPad	Reg_Pad	[in]	图形参数开关,可为空
grpInf	Reg_Info	[in]	图形参数,可为空
repAtt	Record	[in]	替换值记录数据
nCount	Long	[out]	成功返回统改的记录数,失败返回 0

### RepLstRcdFld 方法

描述 字段统改 (弹出统改属性值条件窗口)。

Visual Basic rtl = RegAtt.RepLstRcdFld(lst, ttlName)  
C++ HRESULT pRegAtt->RepLstRcdFld(

IIDList \*lst, BSTR ttlName, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	统改记录号表，若为空则统改所有记录
ttlName	String	[in]	统改属性值条件窗口标题
rtl	Integer	[out]	成功返回 1，失败返回 0

**ModLstRcdFldVal 方法**

描述 统改字段域值。

Visual Basic rtl = RegAtt.ModLstRcdFldVal(lst, fldI, fldVal)

C++ HRESULT pRegAtt->ModLstRcdFldVal(  
IIDList \*lst, short fldI, VARIANT fldVal, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	统改记录号表，若为空则统改所有记录
fldI	Integer	[in]	统改字段号
fldVal	Variant	[in]	字段域值
rtl	Integer	[out]	成功返回 1，失败返回 0

**CalculateFldVal 方法**

描述 计算字段值。

Visual Basic rtl = RegAtt.CalculateFldVal(fldName, calExp, domainErrVal)

C++ HRESULT pRegAtt->CalculateFldVal(  
BSTR fldName,BSTR calExp,  
VARIANT domainErrVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fldName	String	[in]	计算字段名称
calExp	String	[in]	计算表达式
domainErrVal	Variant	[in]	表达式计算过程中定义域错时（如除 0 错），计算结果替换值，若 domainErrVal 不为空，则用 domainErrVal 作为 fldName 的字段值；若为空，则不修改当前记录的 fldName 字段值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### 3.89、NodAtt 对象

**描述** 结点属性对象 NodAtt 封装了操作结点属性的方法。NodAtt 对象不可创建，但能被内部实例化。可以通过区或网工作区对象的 nodAtt 属性、也可以通过查询区、网工作区对象的 INodAtt 接口来引用 NodAtt 对象。

#### 3.89.1、属性：

属性	描述
stru	结点属性结构；通过此属性可以操作结点属性结构
Record_Head	
可读写	

#### 3.89.2、方法：

##### Get 方法

**描述** 取结点 ni 的属性。

Visual Basic rtl = NodAtt.Get(ni, att)

C++ HRESULT pNodAtt->Get(  
long ni, IRecord \*\*att, short \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
ni	Long	[in]	结点实体号
att	Record	[out]	属性记录值
rtl	Integer	[out]	返回值：1=成功、2=空属性记录、0=失败、-1=该属性对应的图元已经被删除

##### GetDef 方法

**描述** 取缺省属性记录。

Visual Basic rtl = NodAtt.GetDef(att)

C++ HRESULT pNodAtt->GetDef(IRecord \*\*att, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
att	Record	[out]	缺省属性记录值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

##### Write 方法

**描述** 写属性。

Visual Basic rtl = NodAtt.Write(ni, att)  
 C++ HRESULT pNodAtt->Write(  
 long ni, IRecord \*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点实体号
att	Record	[in]	属性记录值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### WriteDef 方法

描述 设置缺省属性记录值。

Visual Basic rtl = NodAtt.WriteDef(att)  
 C++ HRESULT pNodAtt->WriteDef(IRecord \*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
att	Record	[in]	缺省属性记录值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetID 方法

描述 取属性 ID。

Visual Basic pid = NodAtt.GetID(ni)  
 C++ HRESULT pNodAtt->GetID(long ni, long \*nid);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点实体号
nid	Long	[out]	返回属性 ID

### GetMaxID 方法

描述 取最大属性 ID 值。

Visual Basic max\_id\_val = NodAtt.GetMaxID()  
 C++ HRESULT pNodAtt->GetMaxID(long \*max\_id\_val);

参数说明：

参数	类型	出口/入口	描述
max_id_val	Long	[out]	返回最大属性 ID

### CopyAll 方法

描述 拷贝所有属性记录到表工作区 ai1 中。

Visual Basic    attnum = NodAtt.CopyAll(ai1)

C++            HRESULT pNodAtt->CopyAll(ITblArea \*ai1, long \*attnum);

参数说明：

参数	类型	出口/入口	描述
ai1	TblArea	[in]	目的工作区
attnum	Long	[out]	拷贝的记录数

### CopyList 方法

描述            拷贝 lst 指定的记录到表工作区 ai1 中。

Visual Basic    attnum = NodAtt.CopyList(lst, ai1, [changeStru], [rtnNo])

C++            HRESULT pNodAtt->CopyList(  
IIDList \*lst, ITblArea \*ai1, short changeStru=1, short rtnNo=0, long \*attnum);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	属性记录列表
ai1	TblArea	[in]	目的工作区
changeStru	Integer	[in]	是否改变 ai1 的属性结构 ,使其和 NodAtt 的结构相同。 可选；缺省值：1
rtnNo	Integer	[in]	是否通过 lst 返回新的记录号。 可选；缺省值：0
attnum	Long	[out]	拷贝的记录数

### RepLstRcdFld0 方法

描述            统一改属性记录字段值。

Visual Basic    nCount = NodAtt.RepLstRcdFld0(lst, repFldFlg, attExp, grpPad, grpInf, repAtt)

C++            HRESULT pNodAtt->RepLstRcdFld0(  
IIDList \*lst, IRepFldFlag \*pRepFldFlg, BSTR attExp,  
INod\_Pad \*grpPad, INod\_Info \*grpInf, IRecord \*repAtt, long \*nCount);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	将进行替换的记录号列表对象，若 lst 为空则统一改所有记录
repFldFlg	RepFldFlag	[in]	替换字段开关；repFldFlag[i]对应 i（从 0 开始）号字段，1=替换，0=不替换
attExp	String	[in]	属性条件表达式字符串；可为空,即无属性条件

grpPad	Nod_Pad	[in]	图形参数开关，可为空
grpInf	Nod_Info	[in]	图形参数，可为空
repAtt	Record	[in]	替换值记录数据
nCount	Long	[out]	成功返回统改的记录数，失败返回 0

**RepLstRcdFld 方法**

描述 字段统改（弹出统改属性值条件窗口）。

Visual Basic rtl = NodAtt.RepLstRcdFld(lst, ttlName)

C++ HRESULT pNodAtt->RepLstRcdFld(  
IIDList \*lst, BSTR ttlName, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	统改记录号表，若为空则统改所有记录
ttlName	String	[in]	统改属性值条件窗口标题
rtl	Integer	[out]	成功返回 1，失败返回 0

**ModLstRcdFldVal 方法**

描述 统改字段域值。

Visual Basic rtl = NodAtt.ModLstRcdFldVal(lst, fldI, fldVal)

C++ HRESULT pNodAtt->ModLstRcdFldVal(  
IIDList \*lst, short fldI, VARIANT fldVal, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	统改记录号表，若为空则统改所有记录
fldI	Integer	[in]	统改字段号
fldVal	Variant	[in]	字段域值
rtl	Integer	[out]	成功返回 1，失败返回 0

**CalculateFldVal 方法**

描述 计算字段值。

Visual Basic rtl = NodAtt.CalculateFldVal(fldName, calExp, domainErrVal)

C++ HRESULT pNodAtt->CalculateFldVal(  
BSTR fldName, BSTR calExp,  
VARIANT domainErrVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fldName	String	[in]	计算字段名称

calExp	String	[in]	计算表达式
domainErrVal	Variant	[in]	表达式计算过程中定义域错时 (如除 0 错), 计算结果替换值, 若 domainErrVal 不为空, 则用 domainErrVal 作为 fldName 的字段值; 若为空, 则不修改当前记录的 fldName 字段值
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

### 3.90、NetAtt 对象

**描述** 望区属性对象 NetAtt 封装了操作网属性的方法。NetAtt 对象不可创建, 但能被内部实例化。可以通过网工作区对象的 att 属性或通过查询网工作区对象的 INetAtt 接口来引用 NetAtt 对象。

#### 3.90.1、属性：

属性	描述
stru	网属性结构；通过此属性可以操作网属性结构
Record_Head	可读写

#### 3.90.2、方法：

##### Get 方法

**描述** 取网 ni 的属性。

**Visual Basic** rtl = NetAtt.Get(ni, att)

**C++** HRESULT pNetAtt->Get(long ni, IRecord \*\*att, short \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
ni	Long	[in]	网实体号
att	Record	[out]	属性记录值
rtl	Integer	[out]	返回值：1=成功、2=空属性记录、0=失败、-1=该属性对应的图元已经被删除

##### GetDef 方法

**描述** 取缺省属性记录。

**Visual Basic** rtl = NetAtt.GetDef(att)

**C++** HRESULT pNetAtt->GetDef(IRecord \*\*att, VARIANT\_BOOL \*rtl);



参数说明：

参数	类型	出口/入口	描述
att	Record	[out]	缺省属性记录值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Write 方法**

描述 写属性。

Visual Basic rtl = NetAtt.Write(ni, att)

C++ HRESULT pNetAtt->Write(  
long ni, IRecord \*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	网实体号
att	Record	[in]	属性记录值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**WriteDef 方法**

描述 设置缺省属性记录值。

Visual Basic rtl = NetAtt.WriteDef(att)

C++ HRESULT pNetAtt-&gt;WriteDef(IRecord \*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
att	Record	[in]	缺省属性记录值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetID 方法**

描述 取属性 ID。

Visual Basic nid = NetAtt.GetID(ni)

C++ HRESULT pNetAtt-&gt;GetID(long ni, long \*nid);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	网实体号
nid	Long	[out]	返回属性 ID

**GetMaxID 方法**

描述 取最大属性 ID 值。

Visual Basic max\_id\_val = NetAtt.GetMaxID()

C++ HRESULT pNetAtt->GetMaxID(long \*max\_id\_val);

参数说明：

参数	类型	出口/入口	描述
max_id_val	Long	[out]	返回最大属性 ID

### CopyAll 方法

描述 拷贝所有属性记录到表工作区 ai1 中。

Visual Basic attnum = NetAtt.CopyAll(ai1)

C++ HRESULT pNetAtt->CopyAll(ITblArea \*ai1, long \*attnum);

参数说明：

参数	类型	出口/入口	描述
ai1	TblArea	[in]	目的工作区
attnum	Long	[out]	拷贝的记录数

### CopyList 方法

描述 拷贝 lst 指定的记录到表工作区 ai1 中。

Visual Basic attnum = NetAtt.CopyList(lst, ai1, [changeStru], [rtnNo])

C++ HRESULT pNetAtt->CopyList(  
IIDList \*lst, ITblArea \*ai1, short changeStru=1, short rtnNo=0, long \*attnum);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	属性记录列表
ai1	TblArea	[in]	目的工作区
changeStru	Integer	[in]	是否改变 ai1 的属性结构，使其和 NetAtt 的结构相同。 可选；缺省值：1
rtnNo	Integer	[in]	是否通过 lst 返回新的记录号。 可选；缺省值：0
attnum	Long	[out]	拷贝的记录数

### RepLstRcdFld0 方法

描述 统一属性记录字段值。

Visual Basic nCount = NetAtt.RepLstRcdFld0(lst, repFldFlg, attExp, grpPad, grpInf, repAtt)

C++ HRESULT pNetAtt->RepLstRcdFld0(  
IIDList \*lst, IRepFldFlag \*pRepFldFlg, BSTR attExp,  
INet\_Pad \*grpPad, INet\_Info \*grpInf, IRecord \*repAtt, long \*nCount);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	将进行替换的记录号列表对象，若 lst 为空则统改所有记录
repFldFlg	RepFldFlag	[in]	替换字段开关；repFldFlag[i]对应 i (从 0 开始) 号字段，1=替换，0=不替换
attExp	String	[in]	属性条件表达式字符串；可为空,即无属性条件
grpPad	Net_Pad	[in]	图形参数开关，可为空
grpInf	Net_Info	[in]	图形参数，可为空
repAtt	Record	[in]	替换值记录数据
nCount	Long	[out]	成功返回统改的记录数，失败返回 0

**RepLstRcdFld 方法**

描述 字段统改（弹出统改属性值条件窗口）。

Visual Basic rtl = NetAtt.RepLstRcdFld(lst, ttlName)

C++ HRESULT pNetAtt->RepLstRcdFld(  
IIDList \*lst, BSTR ttlName, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	统改记录号表，若为空则统改所有记录
ttlName	String	[in]	统改属性值条件窗口标题
rtl	Integer	[out]	成功返回 1，失败返回 0

**ModLstRcdFldVal 方法**

描述 统改字段域值。

Visual Basic rtl = NetAtt.ModLstRcdFldVal(lst, fldI, fldVal)

C++ HRESULT pNetAtt->ModLstRcdFldVal(  
IIDList \*lst, short fldI, VARIANT fldVal, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	统改记录号表，若为空则统改所有记录
fldI	Integer	[in]	统改字段号
fldVal	Variant	[in]	字段域值
rtl	Integer	[out]	成功返回 1，失败返回 0

**CalculateFldVal 方法**

描述 计算字段值。

Visual Basic    rtl = NetAtt.CalculateFldVal(fldName, calExp, domainErrVal)

C++            HRESULT pNetAtt->CalculateFldVal(  
                BSTR fldName, BSTR calExp,  
                VARIANT domainErrVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fldName	String	[in]	计算字段名称
calExp	String	[in]	计算表达式
domainErrVal	Variant	[in]	表达式计算过程中定义域错时（如除 0 错），计算结果替换值，若 domainErrVal 不为空 则用 domainErrVal 作为 fldName 的字段值；若为空，则不修改当前记录的 fldName 字段值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### 3.91、NodData 对象

描述        结点数据对象 NodData 封装了对结点数据的操作。NodData 对象不可创建，但能被内部实例化。可以通过区、网工作区对象的 nod 属性或通过查询区、网工作区对象的 INodData 接口来引用 NodData 对象。

#### 3.91.1、属性：

属性	描述
count	结点物理数
Long	
只读	
logCount	结点逻辑数
Long	
只读	
att	结点属性数据
NodAtt	
只读	
extDB	结点外挂数据库
ExtDbTool	
只读	

#### 3.91.2、方法：

## Get 方法

描述 取结点信息。

Visual Basic rtl = NodData.Get(ni, xy, dat)

C++ HRESULT pNodData->Get(  
long ni, ID\_Dot \*\*xy, ILONGList \*\*dat, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
xy	D_Dot	[out]	结点坐标
dat	LONGList	[out]	结点数据
rtl	Integer	[out]	返回值>0 表示成功；=0 表示失败；<0 表示已经被删除。若返回值不等于 0，那么返回值可能是下列宏定义的组合，表示该结点所拥有的数据类型，成功时组合值大于 0，被删除时组合值小于 0。 NOD_COMM：普通结点，拥有位置和连接的弧段信息。用_GetNod 函数读取。 NOD_GRAPH：图形结点，拥有图示化信息。用_GetNodGraph 函数读取。 NOD_LINK：连通方阵，表示弧段连通性的矩阵。用_GetNodLink 函数读取。 NOD_CENTER：网络中心，具有网络中心的信息。用_GetNodCenter 函数读取。 NOD_STOP：网络站点，具有网络站点的信息。用_GetNodStop 函数读取。 NOD_BARRIER：网络阻碍，具有网络阻碍点的信息。用_GetNodBarrier 函数读取。 NOD_TURNANG：弧段转角，具有各弧段转角信息。用_GetNodTurnAng 函数读取。

## Get1 方法

描述 取结点信息。

Visual Basic rtl = NodData.Get1(ni, xy, dat, infDat, inf)

C++ HRESULT pNodData->Get1(long ni, ID\_Dot \*\*xy,  
ILONGList \*\*dat, VARIANT \*infDat, IPnt\_Info \*\*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
xy	D_Dot	[out]	结点坐标
dat	LONGList	[out]	结点数据
infDat	Variant	[out]	结点图形数据
inf	Pnt_Info	[out]	结点图形信息
rtl	Integer	[out]	返回值说明见 Get 方法

### Get3D 方法

描述 取三维结点。

Visual Basic rtl = NodData.Get3D(ni, xyz, dat)

C++ HRESULT p NodData ->Get3D(  
long ni, ID\_3Dot \*\*xyz, ILONGList \*\*dat, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
xyz	D_3Dot	[out]	结点坐标
dat	LONGList	[out]	结点内容
rtl	Integer	[out]	返回值说明见 Get 方法

### Get3D1 方法

描述 取三维结点。

Visual Basic rtl = NodData.Get3D1(ni, xyz, dat, infDat, inf)

C++ HRESULT pNodData->Get3D1(long ni, ID\_3Dot \*\*xyz,  
ILONGList \*\*dat, VARIANT \*infDat, IPnt\_Info \*\*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
xyz	D_3Dot	[out]	结点坐标
dat	LONGList	[out]	结点数据
infDat	Variant	[out]	结点图形数据
inf	Pnt_Info	[out]	结点图形信息
rtl	Integer	[out]	返回值说明见 Get 方法

### GetGraph 方法

描述 取结点图形数据。

Visual Basic rtl = NodData.GetGraph(ni, gdat, inf)  
 C++ HRESULT pNodData->GetGraph (  
 long ni, VARIANT \*gdat, IPnt\_Info \*\*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
gdat	Variant	[out]	结点图形数据
inf	Pnt_Info	[out]	结点图形信息
rtl	Integer	[out]	返回值说明见 Get 方法

### GetLink 方法

描述 取结点的弧段连通值。

Visual Basic rtl = NodData.GetLink(ni, li0, li1, val, valType)  
 C++ HRESULT pNodData->GetLink (long ni, long li0, long li1,  
 double \*val, Enum\_NodLink\_Type \*valType, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
li0	Long	[in]	前一条弧段
li1	Long	[in]	后一条弧段
val	Double	[out]	弧段连通值，连通值范围由连通值数据类型决定
valType	Enum_NodLink_Type	[out]	连通值数据类型
rtl	Integer	[out]	返回值说明见 Get 方法

### GetStop 方法

描述 取网络终止结点数据。

Visual Basic rtl = NodData.GetStop(ni, nsinf)  
 C++ HRESULT pNodData->GetStop (  
 long ni, INod\_Stop\_Info \*\*nsinf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
nsinf	Nod_Stop_Info	[out]	网络终止结点数据
rtl	Integer	[out]	返回值说明见 Get 方法

**GetBarrier 方法**

描述 取网络障碍结点数据。

Visual Basic rtl = NodData.GetBarrier(ni, nbinf)

C++ HRESULT pNodData->GetBarrier (  
long ni, INod\_Barrier\_Info \*\*nbinf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
nbinf	Nod_Barrier_Info	[out]	网络障碍结点数据
rtl	Integer	[out]	返回值说明见 Get 方法

**GetTurnAng 方法**

描述 取结点弧段转角数据。

Visual Basic rtl = NodData.GetTurnAng(ni, li0, li1, val)

C++ HRESULT pNodData->GetTurnAng (  
long ni, long li0, long li1, float \*val, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
li0	Long	[in]	前一条弧段
li1	Long	[in]	后一条弧段
val	Single	[out]	弧段转角值
rtl	Integer	[out]	返回值说明见 Get 方法

**GetCenter 方法**

描述 取第 ni 号网络中心结点数据。

Visual Basic rtl = NodData.GetCenter(ni, ncinf)

C++ HRESULT pNodData->GetCenter(long ni, INod\_Center\_Info \*\*ncinf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
ncinf	Nod_Center_Info	[out]	网络中心结点信息
rtl	Integer	[out]	返回值说明见 Get 方法

**GetExistFlag 方法**

描述 取结点存在标志。

Visual Basic rtl = NodData.GetExistFlag(nodi)



C++ HRESULT pNodData->GetExistFlag(long nodi, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
nodi	Long	[in]	结点号
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

### GetInfo 方法

描述 取结点图形信息。

Visual Basic rtl = NodData.GetInfo(ni, inf)

C++ HRESULT pNodData->GetInfo(long ni, IPnt\_Info \*\*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
inf	Pnt_Info	[out]	结点图形信息
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

### UpdateInfo 方法

描述 更新结点图形信息。

Visual Basic rtl = NodData.UpdateInfo(ni, inf)

C++ HRESULT pNodData-> UpdateInfo(long ni, IPnt\_Info \*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
inf	Pnt_Info	[in]	结点图形信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### UpdateGraph 方法

描述 更新结点图形数据。若结点没有图形信息，则返回失败；  
有则该函数更新结点图形范围。

Visual Basic rtl = NodData.UpdateGraph(ni, gdat, inf)

C++ HRESULT pNodData-> UpdateGraph(  
long ni, VARIANT gdat, IPnt\_Info \*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
gdat	Variant	[in]	结点图形数据
inf	Pnt_Info	[in]	结点图形信息

rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除
-----	---------	-------	-----------------------

**UpdateLink 方法**

**描述** 更新结点连通值。如果 ni 号结点已经有连通方阵，那么该函数更新原来的连通方阵，否则就将连通方阵添加到 ni 号结点，并改变结点数据类型（位或 NOD\_LINK）。

**Visual Basic** rtl = NodData.UpdateLink(ni, li0, li1, val)

**C++** HRESULT pNodData->UpdateLink (  
long ni, long li0, long li1, double val, short \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
li0	Long	[in]	前一条弧段
li1	Long	[in]	后一条弧段
val	Double	[in]	弧段连通值，连通值的取值取决于 _SetLinkValType 函数所设类型
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**UpdateCenter 方法**

**描述** 更新网络中心结点数据。若结点没有图形信息，则返回失败；有则该函数更新结点图形范围。

**Visual Basic** rtl = NodData.UpdateCenter(ni, ncinf)

**C++** HRESULT pNodData->UpdateCenter (  
long ni, INod\_Center\_Info \*ncinf, short \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
ncinf	Nod_Center_Info	[in]	网络中心结点数据
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**UpdateStop 方法**

**描述** 更新网络终止结点数据。如果 ni 号结点已经有网络终止结点数据，那么该函数更新原来的网络终止结点数据；否则就将网络终止结点数据添加到 ni 号结点，并改变结点数据类型（位或 NOD\_STOP）。

**Visual Basic** rtl = NodData.UpdateStop(ni, nsinf)

**C++** HRESULT pNodData->UpdateStop (  
long ni, INod\_Stop\_Info \*nsinf, short \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
nsinf	Nod_Stop_Info	[in]	网络终止结点数据
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**UpdateBarrier 方法**

**描述** 更新网络障碍结点数据。如果 ni 号结点已经有网络障碍结点数据，那么该函数更新原来的网络障碍结点数据，否则就将网络障碍结点数据添加到 ni 号结点，并改变结点数据类型（位或 NOD\_BARRIER）。

**Visual Basic** rtl = NodData.UpdateBarrier(ni, nbinf)

**C++** HRESULT pNodData->UpdateBarrier (  
long ni, INod\_Barrier\_Info \*nbinf, short \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
nbinf	Nod_Barrier_Info	[in]	网络障碍结点数据
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**UpdateTurnAng 方法**

**描述** 更新网络转角结点数据。

**Visual Basic** rtl = NodData.UpdateTurnAng(ni, li0, li1, val)

**C++** HRESULT pNodData->UpdateTurnAng (  
long ni, long li0, long li1, float val, short \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
li0	Long	[in]	前一条弧段
li1	Long	[in]	后一条弧段
val	Single	[in]	弧段转角值
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**GetType 方法**

**描述** 取结点类型。

**Visual Basic** nodType = NodData.GetType(ni)

**C++** HRESULT pNodData->GetType(long ni, Enum\_Nod\_Type \*nodType);

**参数说明：**

参数	类型	出口/入口	描述
----	----	-------	----

ni	Long	[in]	结点号
nodType	Enum_Nod_Type	[out]	返回结点类型

**SetLinkValType 方法**

描述 设置连通值类型。

Visual Basic rtl = NodData.SetLinkValType(ni, valType)

C++ HRESULT pNodData-> SetLinkValType (  
long ni, Enum\_NodLink\_Type valType, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
valType	Enum_NodLink_Type	[in]	结点连通值数据类型，可以是下列常量之一： BIT_TYPE 、 BYTE_TYPE 、 SHORT_TYPE、 LONG_TYPE、 FLOAT_TYPE、 DOUBLE_TYPE
rtl	Integer	[out]	成功返回 1，失败返回 0

**GetLinkValType 方法**

描述 取连通值类型。

Visual Basic rtl = NodData.GetLinkValType(ni, valType)

C++ HRESULT pNodData-> GetLinkValType (  
long ni, Enum\_NodLink\_Type \*valType, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
valType	Enum_NodLink_Type	[out]	结点连通值数据类型，可以是下列常量之一： BIT_TYPE 、 BYTE_TYPE 、 SHORT_TYPE、 LONG_TYPE、 FLOAT_TYPE、 DOUBLE_TYPE
rtl	Integer	[out]	成功返回 1，失败返回 0

**GetDatLen 方法**

描述 取结点数据长度（字节长度）。

Visual Basic rtl = NodData.GetDatLen (ni, nodLen, infLen)

C++ HRESULT pNodData->GetDatLen(

```
long ni, short *nodLen, short *infLen, short *rtl);
```

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	点实体号
nodLen	Integer	[out]	结点数据长度
infLen	Integer	[out]	结点信息数据长度(字节为单位)
rtl	Integer	[out]	返回结点数据长度

### GetPos3D 方法

描述 取三维结点坐标。

Visual Basic rtl = NodData.GetPos3D(ni, xyz)

C++ HRESULT pNodData-> GetPos3D(long ni, ID\_3Dot \*\*xyz, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
xyz	D_3Dot	[out]	三维结点坐标
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

### Append3D 方法

描述 添加一个三维结点。

Visual Basic nodNo = NodData.Append3D(xyz, dat)

C++ HRESULT pNodData-> Append3D(  
ID\_3Dot \*xyz, ILONGList \*dat, long \*nodNo);

参数说明：

参数	类型	出口/入口	描述
xyz	D_3Dot	[in]	结点坐标
dat	LONGList	[in]	结点结点数据
nodNo	Long	[out]	成功返回结点号 (>0), 失败返回 0

### Append3D1 方法

描述 添加一个三维结点。

Visual Basic nodNo = NodData.Append3D1(xyz, dat, infDat, inf)

C++ HRESULT pNodData-> Append3D1(ID\_3Dot \*xyz,  
ILONGList \*dat, VARIANT infDat, IPnt\_Info \*inf, long \*nodNo);

参数说明：

参数	类型	出口/入口	描述
xyz	D_3Dot	[in]	结点坐标

dat	LONGList	[in]	结点结点数据
infDat	Variant	[in]	结点内容
inf	Pnt_Info	[in]	结点信息
nodNo	Long	[out]	成功返回结点号 (>0), 失败返回 0

**Update3D 方法**

描述 更新三维结点。

Visual Basic rtl = NodData.Update3D(ni, xyz, dat)

C++ HRESULT pNodData-> Update3D(  
long ni, ID\_3Dot \*xyz, ILONGList \*dat, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
xyz	D_3Dot	[in]	结点坐标
dat	LONGList	[in]	结点数据
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**Update3D1 方法**

描述 更新三维结点。

Visual Basic rtl = NodData.Update3D1(ni, xyz, dat, infDat, inf)

C++ HRESULT pNodData-> Update3D1(long ni, ID\_3Dot \*xyz,  
ILONGList \*dat, VARIANT infDat, IPnt\_Info \*inf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
xyz	D_3Dot	[in]	结点坐标
dat	LONGList	[in]	结点数据
infDat	Variant	[in]	结点内容
inf	Pnt_Info	[in]	结点信息
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**UpdatePos3D 方法**

描述 更新三维结点坐标。

Visual Basic rtl = NodData.UpdatePos3D(ni, xyz)

C++ HRESULT pNodData-> UpdatePos3D (  
long ni, ID\_3Dot \*xyz, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	成功返回点号 (>0), 失败返回 0
xyz	D_3Dot	[in]	点坐标
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**GetPos 方法**

描述 取结点坐标。

Visual Basic rtl = NodData.GetPos(ni, xy)

C++ HRESULT pNodData-> GetPos(long ni, ID\_Dot \*\*xy, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
xy	D_Dot	[out]	结点坐标
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**Append 方法**

描述 添加一个结点。

Visual Basic nodNo = NodData.Append(xy, dat)

C++ HRESULT pNodData-> Append (  
ID\_Dot \*xy, ILONGList \*dat, long \*nodNo);

参数说明：

参数	类型	出口/入口	描述
xy	D_Dot	[in]	结点坐标
dat	LONGList	[in]	结点结点数据
nodNo	Long	[out]	成功返回结点号 (>0), 失败返回 0

**Append1 方法**

描述 添加一个结点。

Visual Basic nodNo = NodData.Append1(xy, dat, infDat, inf)

C++ HRESULT pNodData-> Append1(ID\_Dot \*xy,  
ILONGList \*dat, VARIANT infDat, IPnt\_Info \*inf, long \*nodNo);

参数说明：

参数	类型	出口/入口	描述
xy	D_Dot	[in]	结点坐标
dat	LONGList	[in]	结点结点数据
infDat	Variant	[in]	结点内容
inf	Pnt_Info	[in]	结点信息

nodNo	Long	[out]	成功返回结点号 (>0), 失败返回 0
-------	------	-------	----------------------

**Update 方法**

描述 更新结点。

Visual Basic rtl = NodData.Update(ni, xy, dat)

C++ HRESULT pNodData-> Update(  
long ni, ID\_Dot \*xy, ILONGList \*dat, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
xy	D_Dot	[in]	结点坐标
dat	LONGList	[in]	结点数据
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**Update1 方法**

描述 更新结点。

Visual Basic rtl = NodData.Update1(ni, xy, dat, infDat, inf)

C++ HRESULT pNodData-> Update1(long ni, ID\_Dot \*xy,  
ILONGList \*dat, VARIANT infDat, IPnt\_Info \*inf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
xy	D_Dot	[in]	结点坐标
dat	LONGList	[in]	结点数据
infDat	Variant	[in]	结点内容
inf	Pnt_Info	[in]	结点信息
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**UpdatePos 方法**

描述 更新结点坐标。

Visual Basic rtl = NodData.UpdatePos(ni, xy)

C++ HRESULT pNodData-> UpdatePos (  
long ni, ID\_Dot \*xy, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	成功返回点号 (>0), 失败返回 0
xy	D_Dot	[in]	点坐标



rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

**Del 方法**

描述 从逻辑上删除第 ni 号结点。

Visual Basic rtl = NodData.Del(ni)

C++ HRESULT pNodData-> Del(long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DelGraph 方法**

描述 从逻辑上删除第 ni 号结点的图形信息。

Visual Basic rtl = NodData.DelGraph(ni)

C++ HRESULT pNodData-> DelGraph(long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DelLink 方法**

描述 从逻辑上删除第 ni 号结点的连通矩阵信息。

Visual Basic rtl = NodData.DelLink(ni)

C++ HRESULT pNodData->DelLink(long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DelTurnAng 方法**

描述 从逻辑上删除第 ni 号结点的转角矩阵信息。

Visual Basic rtl = NodData.DelTurnAng(ni)

C++ HRESULT pNodData-> DelTurnAng (long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DelCenter 方法**

描述 从逻辑上删除第 ni 号结点的中心结点信息。

Visual Basic rtl = NodData.DelCenter(ni)

C++ HRESULT pNodData->DelCenter(long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DelStop 方法**

描述 从逻辑上删除第 ni 号结点的终止结点信息。

Visual Basic rtl = NodData.DelStop(ni)

C++ HRESULT pNodData->DelStop(long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DelBarrier 方法**

描述 从逻辑上删除第 ni 号结点的障碍结点信息。

Visual Basic rtl = NodData.DelBarrier(ni)

C++ HRESULT pNodData->DelBarrier(long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UnDel 方法**

描述 恢复被删除的结点。

Visual Basic rtl = NodData.UnDel(ni)

C++ HRESULT pNodData->UnDel(long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UnDelGraph 方法**

描述 恢复被删除的结点图形信息。

Visual Basic rtl = NodData.UnDelGraph(ni)

C++ HRESULT pNodData-> UnDelGraph (long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UnDelLink 方法**

描述 恢复被删除的结点连通矩阵信息。

Visual Basic rtl = NodData.UnDelLink(ni)

C++ HRESULT pNodData-> UnDelLink (long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UnDelTurnAng 方法**

描述 恢复被删除的结点转角矩阵信息。

Visual Basic rtl = NodData.UnDelTurnAng(ni)

C++ HRESULT pNodData->UnDelTurnAng (long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UnDelCenter 方法**

描述 恢复被删除的结点中心结点信息。

Visual Basic rtl = NodData.UnDelCenter(ni)

C++ HRESULT pNodData-> UnDelCenter (long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UnDelStop 方法**

描述 恢复被删除的结点终止结点信息。

Visual Basic rtl = NodData.UnDelStop(ni)

C++ HRESULT pNodData->UnDelStop(long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UnDelBarrier 方法**

描述 恢复被删除的结点障碍结点信息。

Visual Basic rtl = NodData.UnDelBarrier(ni)

C++ HRESULT pNodData-> UnDelBarrier (long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	结点号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetRadius 方法**

描述 取结点平差半径。

Visual Basic rtl = NodData.GetRadius(rad)

C++ HRESULT pNodData-> GetRadius (double \*rad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rad	Double	[out]	结点平差半径
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SetRadius 方法**

描述 设置结点平差半径。

Visual Basic rtl = NodData.SetRadius(rad)

C++ HRESULT pNodData->SetRadius (double rad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rad	Double	[in]	结点平差半径
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetFirstNo 方法**

描述 取第一个分类结点号，必须与 GetNextNo 方法配合使用。

Visual Basic nodNo = NodData.GetFirstNo(nodType)

C++ HRESULT pNodData->GetFirstNo(  
Enum\_Nod\_Type nodType, long \*nodNo);

参数说明：

参数	类型	出口/入口	描述
nodType	Enum_Nod_Type	[in]	结点数据类型
nodNo	Long	[out]	成功返回符合 nodType 类型的下一个结点号，失败返回 0

**GetNextNo 方法**

描述 取下一个分类结点号；必须与 GetFirstNo 方法配合使用。

Visual Basic nodNo = NodData.GetNextNo()

C++ HRESULT pNodData->GetNextNo(long \*nodNo);

参数说明：

参数	类型	出口/入口	描述
nodNo	Long	[out]	成功返回符合 nodType 类型的下一个结点号，失败返回 0

**GetClassNum 方法**

描述 取分类结点数。

Visual Basic rtl = NodData.GetClassNum(nodType, nodNum)

C++ HRESULT pNodData->GetClassNum(  
Enum\_Nod\_Type nodType, long \*nodNum, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
nodType	Enum_Nod_Type	[in]	结点类型
nodNum	Long	[out]	结点数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetClassLst 方法**

描述 取分类结点表。

Visual Basic Set lst = NodData.GetClassLst(nodType)

C++ HRESULT pNodData->GetClassLst (  
Enum\_Nod\_Type nodType, IIDList \*\*lst);

参数说明：

参数	类型	出口/入口	描述
nodType	Enum_Nod_Type	[in]	结点类型
lst	IDList	[out]	分类结点表

**ClearDat 方法**

**描述** 清除工作区的结点数据，且被清除的数据不能恢复，故应该慎用。一般只有在重建拓扑结构之前才调用该函数。如果结点数据中包含有普通结点以外的信息，如图形数据，中心结点数据等，由于包含的信息比较多，这些数据被删除后，重新建立比较费时，所以系统弹出对话框询问是否真要删除。

Visual Basic rtl = Noddata.ClearDat()

C++ HRESULT pNodData->ClearDat(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Near 方法**

**描述** 查找最近的结点。

Visual Basic nodNo = NodData.Near(xy, rtnDis)

C++ HRESULT pNodData->Near(  
ID\_Dot \*xy, double \*rtnDis, long \*nodNo);

参数说明：

参数	类型	出口/入口	描述
xy	D_Dot	[in]	点位置，为图形实际坐标
rtnDis	Double	[out]	点跟结点的距离 可选；缺省值：Nothing
nodNo	Long	[out]	成功返回结点号 (>0)，失败返回 0

**3.92、LinData 对象**

**描述** 线数据对象 LinData 封装了对线数据的操作。LinData 对象不可创建，但能被内部实例化。可以通过区、网工作区对象的 lin 属性或通过查询区、网工作区对象的 ILinData 接口来引用 LinData 对象。

**3.92.1、属性：**

属性	描述
----	----

count	线物理数
Long	
只读	
logCount	线逻辑数
Long	
只读	
att	线属性数据
LinAtt	
只读	
extDB	线外挂数据库
ExtDbTool	
只读	
layer	线图层
Layer	
只读	

### 3.92.2、方法：

#### GetRect 方法

描述 取线实体图形范围。

Visual Basic rtl = LinData.GetRect(li, rc)

C++ HRESULT pLinData ->GetRect(long li, ID\_Rect \*\*rc, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
rc	D_Rect	[out]	图元的图形范围
rtl	Integer	[out]	返回：1/0/-1=存在/不存在/被删除

#### GetRectLayer 方法

描述 取线实体图形范围和所在图层号。

Visual Basic rtl = LinData.GetRectLayer(li, rc, layer)

C++ HRESULT pLinData ->GetRectLayer(  
long li, ID\_Rect \*\*rc, short \*layer, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
rc	D_Rect	[out]	线图元的图形范围
layer	Integer	[out]	线图元所在的图层号

rtl	Integer	[out]	返回：1/0/-1=存在/不存在/被删除
-----	---------	-------	----------------------

**GetLayer 方法**

描述 取线实体所在图层号。

Visual Basic rtl = LinData.GetLayer(li, layer)

C++ HRESULT pLinData -> GetLayer(long li, short \*layer, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
layer	Integer	[out]	线图元所在的图层号
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**GetExistFlag 方法**

描述 取线存在标志。

Visual Basic flg = LinData.GetExistFlag(li)

C++ HRESULT pLinData -> GetExistFlag(long li, short \*flg);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
flg	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**GetAllExistNo 方法**

描述 取所有没有被删除的线号。

Visual Basic Set lst = LinData.GetAllExistNo()

C++ HRESULT pLinData -> GetAllExistNo(IIDList \*\*lst);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[out]	没有被删除的线列表

**GetInfo 方法**

描述 取第 li 条线图形信息。

Visual Basic rtl = LinData.GetInfo(li, inf)

C++ HRESULT pLinData -> GetInfo(long li, ILin\_Info \*\*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
inf	Lin_Info	[out]	线图形信息



rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

**UpdateInfo 方法**

描述 更新第 li 条线图形信息。

Visual Basic rtl = LinData.UpdateInfo(li, inf)

C++ HRESULT pLinData -> UpdateInfo(long li, ILin\_Info \*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
inf	Lin_Info	[in]	线图形信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetDatLen 方法**

描述 取线数据长度（字节长度）。

Visual Basic datLen = LinData.GetDatLen(li)

C++ HRESULT pLinData -> GetDatLen(long li, long \*datLen);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
datLen	Long	[out]	返回线数据长度

**Del 方法**

描述 从逻辑上删除第 li 条线实体。

Visual Basic rtl = LinData.Del(li)

C++ HRESULT pLinData -> Del(long li, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	删除的线实体号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UnDel 方法**

描述 恢复被删除的线。

Visual Basic rtl = LinData.UnDel(li)

C++ HRESULT pLinData -> UnDel(long li, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	恢复的线实体号

rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

**SavePart 方法**

**描述** 保存部分线实体。

**Visual Basic** rtl = LinData.SavePart(pad, inf, [layerOnOffPad]);

**C++** HRESULT pLinData -> SavePart(  
ILin\_Pad \*pad, ILin\_Info \*inf,  
ILayerOnOffPad \*layerOnOffPad=NULL, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
pad	Lin_Pad	[in]	线实体参数开关
inf	Lin_Info	[in]	线实体图形信息
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空， 则只保存处于打开状态的图层。 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CopyPart 方法**

**描述** 拷贝部分线到 ai1 工作区。

**Visual Basic** linNum = LinData.CopyPart(pad, inf, ai1, [layerOnOffPad])

**C++** HRESULT pLinData -> CopyPart (  
ILin\_Pad \*pad, ILin\_Info \*inf, IWorkArea \*ai1,  
ILayerOnOffPad \*layerOnOffPad=NULL, long \*linNum);

**参数说明：**

参数	类型	出口/入口	描述
pad	Lin_Pad	[in]	线实体参数开关
inf	Lin_Info	[in]	线实体图形信息
ai1	WorkArea	[in]	目的工作区
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空， 则只保存处于打开状态的图层。 可选；缺省值：Nothing
linNum	Long	[out]	成功返回拷贝的点数，失败返回 0

**CopyAll 方法**

**描述** 拷贝所有线实体到 ai1 工作区。

**Visual Basic** linNum = LinData.CopyAll(ai1)

**C++** HRESULT pLinData -> CopyAll (IWorkArea \*ai1, long \*linNum);

参数说明：

参数	类型	出口/入口	描述
ai1	WorkArea	[in]	目的工作区
linNum	Long	[out]	成功返回拷贝的线数，失败返回 0

**CopyList 方法**

描述 拷贝列表中的线到工作区 ai1。

Visual Basic linNum = LinData.CopyList(lst, ai1, [rtnNo])

C++ HRESULT pLinData -> CopyList(  
IIDList \*lst, IWorkArea \*ai1, short rtnNo=0, long \*linNum);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	要拷贝的线图元列表
ai1	WorkArea	[in]	目的工作区
rtnNo	Integer	[in]	是否通过 lst 返回新的实体号
linNum	Long	[out]	成功返回拷贝的线数，失败返回 0

**GetDimension 方法**

描述 取 li 号线的维数。

Visual Basic dimension = LinData.GetDimension(li)

C++ HRESULT pLinData -&gt; GetDimension(long li, short \*dimension);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
dimension	Integer	[out]	返回线的维数：2=2 维，3=3 维， 0=失败，-2=2 维但被删除，-3=3 维 但被删除

**Get 方法**

描述 取二维线。

Visual Basic rtl = LinData.Get(li, pos, inf, dim)

C++ HRESULT pLinData -> Get(  
long li, ID\_DotSet \*\*pos, ILin\_Info \*\*inf, short, \*dim, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
pos	D_DotSet	[out]	线点坐标

inf	Lin_Info	[out]	线图形信息
dim	Integer	[out]	线实际维数
rtl	Integer	[out]	返回值：1/0/-1 = 成功/失败/被删除

**Append 方法**

描述 添加一条二维线。

Visual Basic linNo = LinData.Append(pos, inf)

C++ HRESULT pLinData -> Append(  
ID\_DotSet \*pos, ILin\_Info \*inf, long \*linNo);

参数说明：

参数	类型	出口/入口	描述
pos	D_DotSet	[in]	线坐标
inf	Lin_Info	[in]	线图形点信息
linNo	Long	[out]	成功返回线实体号(>0)，失败返回 0

**Update 方法**

描述 更新第 li 号线实体。

Visual Basic rtl = LinData.Update(li, pos, inf)

C++ HRESULT pLinData -> Update(  
long li, ID\_DotSet \*pos, ILin\_Info \*inf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
pos	D_DotSet	[in]	线坐标
inf	Lin_Info	[in]	线信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Get3D 方法**

描述 取三维线。

Visual Basic rtl = LinData.Get3D(pi, pos, inf, dim)

C++ HRESULT pLinData -> Get3D(  
long li, ID\_3DotSet \*\*pos, ILin\_Info \*\*inf, short \*dim, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
pos	D_3DotSet	[out]	线坐标
inf	Lin_Info	[out]	线图形信息

dim	Integer	[out]	线实际维数
rtl	Integer	[out]	返回值：1/0/-1 = 成功/失败/已删除

**Append3D 方法**

描述 添加一条三维线。

Visual Basic linNo = LinData.Append3D(pos, inf)

C++ HRESULT pLinData -> Append3D(  
ID\_3DotSet \*pos, ILin\_Info \*inf, long \*linNo);

参数说明：

参数	类型	出口/入口	描述
pos	D_3DotSet	[in]	线坐标
inf	Lin_Info	[in]	线图形信息
linNo	Long	[out]	成功返回线号 (>0), 失败返回 0

**Update3D 方法**

描述 更新线。

Visual Basic rtl = LinData.Update3D(li, pos, inf)

C++ HRESULT pLinData -> Update3D(  
long li, ID\_3DotSet \*pos, ILin\_Info \*inf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
pos	D_3DotSet	[in]	线坐标
inf	Lin_Info	[in]	线信息
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**ModParam 方法**

描述 统改线参数。

Visual Basic rtl = LinData.ModParam(pad0, inf0, pad1, inf1, [layerOnOffPad])

C++ HRESULT pLinData -> ModParam(  
ILin\_Pad \*pad0, ILin\_Info \*inf0, ILin\_Pad \*pad1, ILin\_Info \*inf1,  
ILayerOnOffPad \*layerOnOffPad=NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad0	Lin_Pad	[in]	查找线实体参数开关
inf0	Lin_Info	[in]	源线实体参数
pad1	Lin_Pad	[in]	修改线实体参数开关

infl	Lin_Info	[in]	新的线实体参数
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空， 则只修改处于打开状态的图层。 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetLength 方法**

描述 取线长度。

Visual Basic rtl = LinData.GetLength(li, length)

C++ HRESULT pLinData -> GetLength(  
long li, double \*length, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
length	Double	[out]	线长度
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Near 方法**

描述 查找最近的线。

Visual Basic linNo = LinData.Near(xy, [layerOnOffPad], [dis])

C++ HRESULT pLinData -> Near(  
ID\_Dot \*xy, ILayerOnOffPad \*layerOnOffPad=NULL,  
double \*dis=NULL, long \*linNo);

参数说明：

参数	类型	出口/入口	描述
xy	D_Dot	[in]	点位置，为图形实际坐标
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空，则只在 处于打开状态的图层中查找。 可选；缺省值：Nothing
dis	Double	[out]	点线距离
linNo	Long	[out]	成功返回线号 (>0)，失败返回 0

**GetTop 方法**

描述 取线拓扑结构数据。

Visual Basic rtl = LinData.GetTop(li, top)

C++ HRESULT pLinData -> GetTop(long li, ILin\_Top \*\*top, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
top	Lin_Top	[out]	线拓扑数据
rtl	Integer	[out]	返回：1/0/-1 = 成功/失败/已删除

**WriteTop 方法**

描述 写线拓扑结构数据。

Visual Basic rtl = LinData.WriteTop(li, top)

C++ HRESULT pLinData -> WriteTop (long li, ILin\_Top \*top, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
top	Lin_Top	[in]	线拓扑数据
rtl	Integer	[out]	返回：1/0/-1 = 成功/失败/已删除

**3.93、PathData 对象**

描述 路径数据对象 PathData 封装了对网络路径的操作。PathData 对象不可创建，但能被内部实例化。可以通过网工作区对象的 path 属性或通过查询网工作区对象的 IPathData 接口来引用 PathData 对象。

**3.93.1、属性：**

属性	描述
count	物理路径数
Long	
只读	
logCount	逻辑路径数
Long	
只读	

**3.93.2、方法：****Append 方法**

描述 添加一条网络路径。

Visual Basic pathNo = PathData.Append(nPath, aPath, inf)

C++ HRESULT pPathData->Append(  
IIDList \*nPath, IIDList \*aPath, ILin\_Info \*inf, long \*pathNo);

参数说明：

参数	类型	出口/入口	描述
nPath	IDList	[in]	结点 ID 列表
aPath	IDList	[in]	线（弧段）ID 列表
inf	Lin_Info	[in]	线（弧段）图形信息
pathNo	Long	[out]	成功返回网络路径号，失败返回 0

### Update 方法

描述 更新网络路径。

Visual Basic rtl = PathData.Update(pi, nPath, aPath, inf)

C++ HRESULT pPathData->Update(  
long pi, IIDList \*nPath, IIDList \*aPath, ILin\_Info \*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	网络路径号
nPath	IDList	[in]	结点 ID 列表
aPath	IDList	[in]	线（弧段）ID 列表
inf	Lin_Info	[in]	线（弧段）图形信息
rtl	Integer	[out]	成功返回 1，失败返回 0

### UpdateInfo 方法

描述 更新网络路径信息。

Visual Basic rtl = PathData.UpdateInfo(pi, inf)

C++ HRESULT pPathData->UpdateInfo(long pi, ILin\_Info \*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	网络路径号
inf	Lin_Info	[in]	线（弧段）图形信息
rtl	Integer	[out]	成功返回 1，失败返回 0

### Del 方法

描述 从逻辑上删除第 pi 条网络路径。

Visual Basic rtl = PathData.Del(pi)

C++ HRESULT pPathData->Del(long pi, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	网络路径号



rtl	Integer	[out]	成功返回 1，失败返回 0
-----	---------	-------	---------------

**UnDel 方法**

描述 恢复被删除的网络路径。

Visual Basic rtl = PathData.UnDel(pi)

C++ HRESULT pPathData->UnDel(long pi, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	网络路径号
rtl	Integer	[out]	成功返回 1，失败返回 0

**Get 方法**

描述 取网络路径。

Visual Basic rtl = PathData.Get(pi, nPath, aPath, inf)

C++ HRESULT pPathData->Get(long pi,  
IIDList \*\*nPath, IIDList \*\*aPath, IILin\_Info \*\*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	网络路径号
nPath	IDList	[out]	结点 ID 列表
aPath	IDList	[out]	线（弧段）ID 列表
inf	Lin_Info	[out]	线（弧段）图形信息
rtl	Integer	[out]	成功返回 1，失败返回 0

**GetLen 方法**

描述 取网络数据块的项数。

Visual Basic rtl = PathData.GetLen(pi, nLen, aLen)

C++ HRESULT pPathData->GetLen(long pi,  
long \*nLen, long \*aLen, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	网络路径号
nLen	Long	[out]	结点数
aLen	Long	[out]	线（弧段）数
rtl	Integer	[out]	成功返回 1，失败返回 0

## 3.94、WorkArea 对象

**描述** WorkArea 对象封装了所有工作区的公共操作。WorkArea 对象既不可创建，也不能被实例化。定义 WorkArea 对象只是为了给具体的工作区（如：点、线、区、网、表工作区）提供一个接口声明规范。

### 3.94.1、属性：

属性	描述
Type	工作区类型
Enum_Area_Type	
只读	
Name	工作区名称
String	
可读写	
Changed	工作区已经改变标志
Boolean	
可读写	
ShowErrMsg	显示错误信息标志
Boolean	
可读写	
Warning	对不能后悔的操作是否给出警告提示
Boolean	
可读写	
Compact	保存工作区时压缩数据标志
Boolean	
可读写	
Empty	工作区是否为空标志
Boolean	
只读	
LastErrCode	最后错误码
Long	
只读	
rect	图形范围
D_Rect	
只读	

## 3.94.2、方法：

**RefreshRect 方法**

描述 重新整理、更新图形范围。

Visual Basic WorkArea.RefreshRect

C++ HRESULT pWorkArea->RefreshRect();

参数说明： 无

**GetAreaFClass 方法**

描述 取工作区文件分类码。

Visual Basic fclass = WorkArea.GetAreaFClass()

C++ HRESULT pWorkArea->GetAreaFClass(short \*fclass);

参数说明：

参数	类型	出口/入口	描述
fclass	Integer	[out]	返回工作区文件分类码

**SetAreaFClass 方法**

描述 设工作区文件分类码。

Visual Basic rtl = WorkArea.SetAreaFClass(fclass)

C++ HRESULT pWorkArea->SetAreaFClass(short fclass, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fclass	Integer	[in]	工作区文件分类码
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetAreaAlias 方法**

描述 取工作区文件别名。

Visual Basic rtl = WorkArea.GetAreaAlias(fAlias)

C++ HRESULT pWorkArea->GetAreaAlias(  
BSTR \*fAlias, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fAlias	String	[out]	工作区文件别名
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SetAreaAlias 方法**

描述 设工作区文件别名。

Visual Basic    rtl = WorkArea.SetAreaAlias(fAlias)  
C++             HRESULT pWorkArea-> SetAreaAlias(  
                  BSTR fAlias, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fAlias	String	[in]	工作区文件别名
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## Load 方法

描述            加载文件到工作区。

Visual Basic    rtl = WorkArea.Load([fName], [szDSN], [szLOG], [szPSW])  
C++             HRESULT pWorkArea-> Load(  
                  BSTR fName=NULL, BSTR szDSN=NULL,  
                  BSTR szLOG=NULL,BSTR szPSW=NULL,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fName	String	[in]	文件名，若为本地数据，则必须带路径。若为网络数据，则不带路径，且不允许数据名称有后缀，如“.wp”，该函数自动核对文件和工作区类型。 可选；缺省值：Nothing
szDSN	String	[in]	ODBC 数据源名称，若=NULL，则 datName 为本地数据，否则是网络数据，数据通过 szDSN 指定的数据源进行存取。 可选；缺省值：Nothing
szLOG	String	[in]	登录号（登录名），即网络数据库的登录号。 可选；缺省值：Nothing
szPSW	String	[in]	登录口令，即网络数据库的登录口令。 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## Save 方法

描述            将工作区存到文件。

Visual Basic    rtl = WorkArea.Save([fName], [szDSN], [szLOG], [szPSW])

C++ HRESULT pWorkArea-> Save (  
BSTR fName=NULL, BSTR szDSN=NULL,  
BSTR szLOG=NULL,BSTR szPSW=NULL,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fName	String	[in]	文件名，若为本地数据，则必须带路径。若为网络数据，则不带路径，且不允许数据名称有后缀，如“.wp”，该函数自动核对文件和工作区类型。 可选；缺省值：Nothing
szDSN	String	[in]	ODBC 数据源名称，若=NULL，则 datName 为本地数据，否则是网络数据，数据通过 szDSN 指定的数据源进行存取。 可选；缺省值：Nothing
szLOG	String	[in]	登录号（登录名），即网络数据库的登录号。 可选；缺省值：Nothing
szPSW	String	[in]	登录口令，即网络数据库的登录口令。 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### SaveAs 方法

描述 将工作区换名存到文件。

Visual Basic rtl = WorkArea.SaveAs()

C++ HRESULT pWorkArea->SaveAs(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### Clear 方法

描述 清除工作区中所有内容，包括图形信息和属性信息，要慎重使用。

Visual Basic rtl = WorkArea.Clear()

C++ HRESULT pWorkArea-> Clear (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

**Clone 方法**

描述 克隆工作区。

Visual Basic Set NewArea = WorkArea.Clone()

C++ HRESULT pWorkArea->Clone(IWorkArea \*\*pINewArea);

参数说明：

参数	类型	出口/入口	描述
NewArea	WorkArea	[out]	返回克隆的新工作区

**3.95、VectWorkArea 对象**

描述 VectWorkArea 对象封装了矢量工作区的公共操作。VectWorkArea 对象既不可创建，也不能被实例化。定义 VectWorkArea 对象只是为了给具体的矢量工作区（点、线、区、网工作区）提供一个接口声明规范。

**3.95.1、属性：无****3.95.2、方法：****SetMMDPath 方法**

描述 设置多媒体数据路径。

Visual Basic rtl = VectWorkArea.SetMMDPath(mmdPath)

C++ HRESULT pVectWorkArea->SetMMDPath(  
BSTR mmdPath, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
mmdPath	String	[in]	多媒体数据路径
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetMMDPath 方法**

描述 取多媒体数据路径。

Visual Basic rtl = VectWorkArea.GetMMDPath(mmdPath)

C++ HRESULT pVectWorkArea->GetMMDPath(  
BSTR \*mmdPath, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

mmdPath	String	[out]	多媒体数据路径
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ModMMDPath 方法**

描述 修改多媒体数据路径。

Visual Basic rtl = VectWorkArea.ModMMDPath()

C++ HRESULT pVectWorkArea->ModMMDPath (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetTicNum 方法**

描述 取工作区中包含的 Tic 点数。

Visual Basic num = VectWorkArea.GetTicNum()

C++ HRESULT pVectWorkArea->GetTicNum([out,retval]short \*num);

参数说明：

参数	类型	出口/入口	描述
num	Integer	[out]	返回工作区中包含的 Tic 点数

**GetTicList 方法**

描述 取 Tic 点列表。

Visual Basic rtl = VectWorkArea.GetTicList(lst, rtl)

C++ HRESULT pVectWorkArea->GetTicList(  
ITic\_DotSet \*\*lst, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
lst	Tic_DotSet	[out]	工作区中包含的 Tic 点列表
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**AppendTicDot 方法**

描述 添加 Tic 点。

Visual Basic rtl = VectWorkArea.AppendTicDot(tic)

C++ HRESULT pVectWorkArea-> AppendTicDot(  
ITic\_Dot \*tic, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
tic	Tic_Dot	[in]	添加的 Tic 点

rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

**ReplaceTicList 方法**

**描述** 将 lst 中的 TIC 点替换 ai 工作区中原来的 TIC 点。

**Visual Basic** rtl = VectWorkArea.ReplaceTicList(lst)

**C++** HRESULT pVectWorkArea->ReplaceTicList(  
ITic\_DotSet \*lst, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
lst	Tic_DotSet	[in]	准备替换的 TIC 点集
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ClearAllTicDot 方法**

**描述** 清除工作区中所有 Tic 点。

**Visual Basic** rtl = VectWorkArea.ClearAllTicDot()

**C++** HRESULT pVectWorkArea->ClearAllTicDot(VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CopyTicDots 方法**

**描述** 拷贝工作区中所有 TIC 点到 ai 工作区。

**Visual Basic** rtl = VectWorkArea.CopyTicDots(ai)

**C++** HRESULT pVectWorkArea->CopyTicDots(  
IWorkArea \*ai, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
ai	WorkArea	[in]	目的工作区对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**3.96、PntArea 对象**

**描述** PntArea 对象封装了对点工作区的操作。

**3.96.1、属性：**

属性	描述
----	----



count	点工作区包含的点物理数
Long	
只读	
logCount	点工作区包含的点逻辑数
Long	
只读	
att	点工作区属性数据
PntAtt	
只读	
extDB	点工作区外挂数据库
ExtDbTool	
只读	
layer	点工作区图层
Layer	
只读	
Type	工作区类型
Enum_Area_Type	
只读	
Name	工作区名称
String	
可读写	
Changed	工作区已经改变标志
Boolean	
可读写	
ShowErrMsg	显示错误信息标志
Boolean	
可读写	
Warning	对不能后悔的操作是否给出警告提示
Boolean	
可读写	
Compact	保存工作区时压缩数据标志
Boolean	
可读写	
Empty	工作区是否为空标志
Boolean	
只读	

LastErrCode	最后错误码
Long	
只读	
rect	图形范围
D_Rect	
只读	

### 3.96.2、方法：

#### GetRect 方法

描述 取点实体图形范围。

Visual Basic rtl = PntArea.GetRect(pi, rc)

C++ HRESULT pPntArea->GetRect(long pi, ID\_Rect \*\*rc, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	点实体号
rc	D_Rect	[out]	图元的图形范围
rtl	Integer	[out]	返回：1/0/-1=存在/不存在/被删除

#### GetRectLayer 方法

描述 取点实体图形范围和所在图层号。

Visual Basic rtl = PntArea.GetRectLayer(pi, rc, layer)

C++ HRESULT pPntArea->GetRectLayer(  
long pi, ID\_Rect \*\*rc, short \*layer, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	点实体号
rc	D_Rect	[out]	图元的图形范围
layer	Integer	[out]	图元所在的图层号
rtl	Integer	[out]	返回：1/0/-1=存在/不存在/被删除

#### GetLayer 方法

描述 取点实体所在图层号。

Visual Basic rtl = PntArea.GetLayer(pi, lay)

C++ HRESULT pPntArea->GetLayer(long pi, short \*lay, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	点实体号

lay	Integer	[out]	图元所在的图层号
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**GetExistFlag 方法**

描述 取点存在标志。

Visual Basic rtl = PntArea.GetExistFlag(pi)

C++ HRESULT pPntArea->GetExistFlag(long pi, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	点实体号
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**GetAllExistNo 方法**

描述 取所有没有被删除的点号。

Visual Basic Set lst = PntArea.GetAllExistNo()

C++ HRESULT pPntArea->GetAllExistNo(IIDList \*\*lst);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[out]	没有被删除的点号列表

**GetInfo 方法**

描述 取 pi 点图形信息。

Visual Basic rtl = PntArea.GetInfo(pi, inf)

C++ HRESULT pPntArea->GetInfo(long pi, IPnt\_Info \*\*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	点实体号
inf	Pnt_Info	[out]	点图形信息
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**UpdateInfo 方法**

描述 更新 pi 点图形信息。

Visual Basic rtl = PntArea.UpdateInfo(pi, inf)

C++ HRESULT pPntArea->UpdateInfo(long pi, IPnt\_Info \*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	点实体号

inf	Pnt_Info	[in]	点图形信息
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**GetType 方法**

描述 取 pi 点类型。

Visual Basic rtl = PntArea.GetType(pi, pntType)

C++ HRESULT pPntArea->GetType(  
long pi, Enum\_Pnt\_Type \*pntType, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	点实体号
pntType	Enum_Pnt_Type	[out]	点类型
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**GetDatLen 方法**

描述 取点数据长度（字节长度）。

Visual Basic retLen = PntArea.GetDatLen(pi)

C++ HRESULT pPntArea->GetDatLen(long pi, long \*retLen);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	点实体号
retLen	Long	[out]	返回点数据长度

**Get3D 方法**

描述 取点详细信息。

Visual Basic rtl = PntArea.Get3D(pi, xyz, NoteDat, inf)

C++ HRESULT pPntArea->Get3D(  
long pi, ID\_3Dot \*\*xyz, BSTR \*NoteDat, IPnt\_Info \*\*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	点实体号
xyz	D_3Dot	[out]	pi 点坐标
NoteDat	String	[out]	pi 点内容
inf	Pnt_Info	[out]	pi 点图形信息
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**GetPos3D 方法**

描述 取 pi 点坐标。

Visual Basic rtl = PntArea.GetPos3D(pi, xyz)

C++ HRESULT pPntArea->GetPos3D(long pi, ID\_3Dot \*\*xyz, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	点实体号
xyz	D_3Dot	[out]	pi 点坐标
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**Append3D 方法**

描述 添加一个点。

Visual Basic pi = PntArea.Append3D(xyz, noteDat, inf)

C++ HRESULT pPntArea->Append3D(  
ID\_3Dot \*xyz, BSTR noteDat, IPnt\_Info \*inf, long \*pi);

参数说明：

参数	类型	出口/入口	描述
xyz	D_3Dot	[in]	点坐标
noteDat	String	[in]	点内容
inf	Pnt_Info	[in]	点信息
pi	Long	[out]	成功返回点号 (>0), 失败返回 0

**Update3D 方法**

描述 更新 pi 点。

Visual Basic rtl = PntArea.Update3D(pi, xyz, noteDat, inf)

C++ HRESULT pPntArea->Update3D(  
long pi, ID\_3Dot \*xyz, BSTR noteDat, IPnt\_Info \*inf, long \*pi);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	更新点实体号
xyz	D_3Dot	[in]	点坐标
noteDat	String	[in]	点内容
inf	Pnt_Info	[in]	点信息
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**UpdatePos3D 方法**

描述 更新 pi 点坐标。

Visual Basic rtl = PntArea.UpdatePos3D(pi, xyz)  
 C++ HRESULT pPntArea->UpdatePos3D (  
 long pi, ID\_3Dot \*xyz, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	成功返回点号 (>0), 失败返回 0
xyz	D_3Dot	[in]	点坐标
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

### Get 方法

描述 取点详细信息。

Visual Basic rtl = PntArea.Get(pi, xy, NoteDat, inf)  
 C++ HRESULT pPntArea->Get(  
 long pi, ID\_Dot \*\*xy, BSTR \*NoteDat, IPnt\_Info \*\*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	点实体号
xy	D_Dot	[out]	pi 点坐标
NoteDat	String	[out]	pi 点内容
inf	Pnt_Info	[out]	pi 点图形信息
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

### GetPos 方法

描述 取 pi 点坐标。

Visual Basic rtl = PntArea.GetPos(pi, xy)  
 C++ HRESULT pPntArea->GetPos(long pi, ID\_Dot \*\*xy, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	点实体号
xy	D_Dot	[out]	pi 点坐标
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

### Append 方法

描述 添加一个点。

Visual Basic pi = PntArea.Append(xy, noteDat, inf)  
 C++ HRESULT pPntArea->Append(  
 ID\_Dot \*xy, BSTR noteDat, IPnt\_Info \*inf, long \*pi);

参数说明：

参数	类型	出口/入口	描述
xy	D_Dot	[in]	点坐标
noteDat	String	[in]	点内容
inf	Pnt_Info	[in]	点信息
pi	Long	[out]	成功返回点号 (>0), 失败返回 0

**Update 方法**

描述 更新 pi 点。

Visual Basic rtl = PntArea.Update(pi, xy, noteDat, inf)

C++ HRESULT pPntArea->Update(  
long pi, ID\_Dot \*xy, BSTR noteDat, IPnt\_Info \*inf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	点实体号
xy	D_Dot	[in]	点坐标
noteDat	String	[in]	点内容
inf	Pnt_Info	[in]	点信息
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**UpdatePos 方法**

描述 更新 pi 点坐标。

Visual Basic rtl = PntArea.UpdatePos(pi, xy)

C++ HRESULT pPntArea->UpdatePos(  
long pi, ID\_Dot \*xy, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	更新点实体号
xy	D_Dot	[in]	点坐标
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**Del 方法**

描述 从逻辑上删除第 pi 号点实体。

Visual Basic rtl = PntArea.Del(pi)

C++ HRESULT pPntArea-> Del(long pi, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

pi	Long	[in]	删除的点实体号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UnDel 方法**

描述 恢复被删除的点。

Visual Basic rtl = PntArea.UnDel(pi)

C++ HRESULT pPntArea->UnDel(long pi, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	恢复的点实体号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SavePart 方法**

描述 保存部分点实体。

Visual Basic rtl = PntArea.SavePart(pad, inf, [layerOnOffPad])

C++ HRESULT pPntArea-> SavePart(  
IPnt\_Pad \*pad, IPnt\_Info \*inf,  
ILayerOnOffPad \*layerOnOffPad=NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点实体参数开关。
inf	Pnt_Info	[in]	点实体图形信息
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空， 则只保存处于打开状态的图层。 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CopyPart 方法**

描述 拷贝部分点到 ai1 工作区。

Visual Basic nCount = PntArea.CopyPart(pad, inf, ai1, [layerOnOffPad])

C++ HRESULT pPntArea-> CopyPart (  
IPnt\_Pad \*pad, IPnt\_Info \*inf, IPntArea \*ai1,  
ILayerOnOffPad \*layerOnOffPad=NULL, long \*nCount);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点实体参数开关 可选。缺省值：Nothing



inf	Pnt_Info	[in]	点实体图形信息
ai1	PntArea	[in]	目的工作区
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空， 则只保存处于打开状态的图层。 可选；缺省值：Nothing
nCount	Long	[out]	成功返回拷贝的点数，失败返回 0

**CopyAll 方法**

描述 拷贝所有点实体到 ai1 工作区。

Visual Basic nCount = PntArea.CopyAll(ai1)

C++ HRESULT pPntArea-> CopyAll (IPntArea \*ai1, long \*nCount);

参数说明：

参数	类型	出口/入口	描述
ai1	PntArea	[in]	目的工作区
nCount	Long	[out]	成功返回拷贝的点数，失败返回 0

**CopyList 方法**

描述 拷贝列表中的点到工作区 ai1。

Visual Basic nCount = PntArea.CopyList(lst, ai1, [rtnNo])

C++ HRESULT pPntArea-> CopyList(  
IIDList \*lst, IPntArea \*ai1, short rtnNo=0, long \*nCount);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	要拷贝的图元列表
ai1	PntArea	[in]	目的工作区
rtnNo	Integer	[in]	是否通过 lst 返回新的实体号
nCount	Long	[out]	成功返回拷贝的点数，失败返回 0

**GetAutoGetBinDatFlag 方法**

描述 取自动取二进制数据字段标志。

Visual Basic flg = PntArea.GetAutoGetBinDatFlag()

C++ HRESULT pPntArea-> GetAutoGetBinDatFlag(short \*flg);

参数说明：

参数	类型	出口/入口	描述
flg	Integer	[out]	返回值：TRUE/FALSE=自动/不自动

**SetAutoGetBinDatFlag 方法**

描述 设自动取二进制数据字段标志。

Visual Basic preFlg = PntArea.SetAutoGetBinDatFlag(flag)

C++ HRESULT pPntArea-> SetAutoGetBinDatFlag(short flg, short \*preFlg);

参数说明：

参数	类型	出口/入口	描述
flg	Integer	[in]	自动取二进制数据字段标志
preFlg	Integer	[out]	返回值：TRUE/FALSE=自动/不自动

**ModParam 方法**

描述 统改点参数。

Visual Basic rtl = PntArea.ModParam(pad0, inf0, pad1, inf1, [layerOnOffPad])

C++ HRESULT pPntArea-> ModParam(  
IPnt\_Pad \*pad0, IPnt\_Info \*inf0, IPnt\_Pad \*pad1, IPnt\_Info \*inf1,  
ILayerOnOffPad \*layerOnOffPad=NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad0	Pnt_Pad	[in]	查找点实体参数开关
inf0	Pnt_Info	[in]	源点实体参数
pad1	Pnt_Pad	[in]	修改点实体参数开关
inf1	Pnt_Info	[in]	新的点实体参数
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空， 则只修改处于打开状态的图层。 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetMapParam 方法**

描述 取地图参数。

Visual Basic rtl = PntArea.GetMapParam(mpar)

C++ HRESULT pPntArea-> GetMapParam(  
IMap\_Para \*\*mpar, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
mpar	Map_Para	[out]	地图参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SetMapParam 方法**

描述 设地图参数。

Visual Basic rtl = PntArea.SetMapParam(mpar)

C++ HRESULT pPntArea-> SetMapParam(  
IMap\_Para \*mpar, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
mpar	Map_Para	[in]	地图参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ModMapParam 方法**

描述 修改地图参数。

Visual Basic rtl = PntArea.ModMapParam()

C++ HRESULT pPntArea-> ModMapParam(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CopyMapParam 方法**

描述 拷贝地图参数到工作区 ai1。

Visual Basic rtl = PntArea.CopyMapParam(ai1)

C++ HRESULT pPntArea-> CopyMapParam(  
IWorkArea \*ai1, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai1	WorkArea	[in]	目的工作区
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Transform 方法**

描述 图形变换。

Visual Basic rtl = PntArea.Transform(tp, [chgParam])

C++ HRESULT pPntArea-> Transform(  
ITran\_Par \*tp, short chgParam=0, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
tp	Tran_Par	[in]	图形变换参数

chgParam	Integer	[in]	改变点图形参数标志,1/0=改变/不改变;若改变点图形参数标志为1,而且变换包括了旋转和缩放,那么点图形参数中的角度值作相应的旋转,参数中的大小值作相应的缩放。 可选;缺省值:0
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

**Near 方法**

描述 查找最近点。

Visual Basic pntNo = PntArea.Near(xy, [layerOnOffPad], [dis])

C++ HRESULT pPntArea-> Near(  
ID\_Dot \*xy, ILayerOnOffPad \*layerOnOffPad=NULL,  
double \*dis=NULL, long \*pntNo);

参数说明:

参数	类型	出口/入口	描述
xy	D_Dot	[in]	点位置,为图形实际坐标
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志,若此实体不为空,则只在处于打开状态的图层中查找。 可选;缺省值:Nothing
dis	Double	[out]	两点之间的距离
pntNo	Long	[out]	成功返回 TRUE,失败返回 FALSE

**dbGetAreaDSType 方法**

描述 取工作区数据源类型。

Visual Basic type = PntArea.dbGetAreaDSType()

C++ HRESULT pPntArea-> dbGetAreaDSType(Enum\_GisDS\_Type \*type);

参数说明:

参数	类型	出口/入口	描述
type	Enum_GisDS_Type	[out]	返回工作区数据源类型: PC_GIS/NET_GIS=单机数据源/ 网络数据源

**dbGetAreaDSName 方法**

描述 取工作区数据源名称。

Visual Basic rtl = PntArea.dbGetAreaDSName(szDSN)

C++ HRESULT pPntArea->dbGetAreaDSName(BSTR \*szDSN, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
szDSN	String	[out]	工作区数据源名称
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dbLockAll 方法

描述 全锁工作区中的数据。

Visual Basic rtl = PntArea.dbLockAll()

C++ HRESULT pPntArea-> dbLockAll(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dbUnLockAll 方法

描述 解除全锁。

Visual Basic rtl = PntArea.dbUnLockAll()

C++ HRESULT pPntArea-> dbUnLockAll(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dbLockRect 方法

描述 锁工作区中矩形内的数据。

Visual Basic rtl = PntArea.dbLockRect(rc)

C++ HRESULT pPntArea-> dbLockRect(ID\_Rect \*rc, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rc	D_Rect	[in]	封锁的数据范围
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dbUnLockRect 方法

描述 解除矩形锁。

Visual Basic rtl = PntArea.dbUnLockRect()

C++ HRESULT pPntArea-> dbUnLockRect(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

**dbGetLockRange 方法**

描述 取被自己封锁的数据范围。

Visual Basic rtl = PntArea.dbGetLockRange(rc)

C++ HRESULT pPntArea-> dbGetLockRange(ID\_Rect \*\*rc, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
rc	D_Rect	[out]	被自己封锁的数据范围
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbGetLockNumByOther 方法**

描述 取被其他用户封锁的矩形块数。

Visual Basic num = PntArea.dbGetLockNumByOther()

C++ HRESULT pPntArea-> dbGetLockNumByOther(long \*num);

参数说明：

参数	类型	出口/入口	描述
num	Long	[out]	返回值：-1-失败，0-其他用户未封锁， >0-其他用户封锁的矩形块数

**dbGetLockRangeByOther 方法**

描述 取被其他用户封锁的矩形范围。

Visual Basic num = PntArea.dbGetLockRangeByOther(rc)

C++ HRESULT pPntArea-> dbGetLockRangeByOther(  
ID\_RectSet \*\*rc, long \*num);

参数说明：

参数	类型	出口/入口	描述
rc	D_RectSet	[out]	被其他用户封锁的矩形范围
num	Long	[out]	返回值：-1-失败，0-其他用户未封锁， >0-其他用户封锁的矩形块数

**dbLockSingle 方法**

描述 封锁单个图元。

Visual Basic rtl = PntArea.dbLockSingle(i)

C++ HRESULT pPntArea-> dbLockSingle(long i, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

i	Long	[in]	封锁的图元号
rtl	Boolean	[out]	返回值：1/0 = 成功/失败

**dbUnLockSingle 方法**

描述 取消单个图元的封锁。

Visual Basic rtl = PntArea.dbUnLockSingle(i)

C++ HRESULT pPntArea-> dbUnLockSingle(long i, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
i	Long	[in]	取消封锁的图元号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbBeginTransaction 方法**

描述 事务开始。

Visual Basic rtl = PntArea.dbBeginTransaction([newTrans])

C++ HRESULT pPntArea-> dbBeginTransaction(short newTrans=0, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
newTrans	Integer	[in]	newTrans=0 时只增加嵌套次数，=1 时则强制提交以前开始的事务，并开始一个新的事务。 可选；缺省值：0
rtl	Integer	[out]	成功返回事务嵌套次数，失败返回 0

**dbRollbackTransaction 方法**

描述 事务回滚，忽略事务嵌套次数，回滚到最外层事务开始的地方，并将嵌套次数设为 0。

Visual Basic rtl = PntArea.dbRollbackTransaction()

C++ HRESULT pPntArea-> dbRollbackTransaction(short \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Integer	[out]	成功返回 TRUE，失败返回 FALSE

**dbCommitTransaction 方法**

描述 事务提交。

Visual Basic rtl = PntArea.dbCommitTransaction([commitNow])

C++ HRESULT pPntArea-> dbCommitTransaction(

```
short commitNow=0, short *rtl);
```

参数说明：

参数	类型	出口/入口	描述
commitNow	Integer	[in]	commitNow!=0 则马上提交并将嵌套次数置 0，否则只有嵌套次数减到 0 时才提交。 可选；缺省值：0
rtl	Integer	[out]	成功返回 TRUE，失败返回 FALSE

### dbCreateFieldIndex 方法

描述 创建属性字段索引（只有拥有者才能创建索引）

Visual Basic rtl = PntArea.dbCreateFieldIndex(stru,  
ndxFldFlag, IsUnique, IsClustered, IsPadIndex)

C++ HRESULT pPntArea->dbCreateFieldIndex(  
IRecord\_Head \*stru, IRepFldFlag \*ndxFldFlag, short IsUnique,  
short IsClustered, short IsPadIndex, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
stru	Record_Head	[in]	属性结构（主要用其中的字段名称）
ndxFldFlag	RepFldFlag	[in]	创建字段索引标识（用来对应于创建索引的字段，ndxFldFlag[i]=1：表示字段 i 创建索引；=0：表示字段 i 不创建索引）
IsUnique	Integer	[in]	创建唯一索引标识：1/0-是/否
IsClustered	Integer	[in]	创建聚簇索引标识：1/0-是/否
IsPadIndex	Integer	[in]	1/0-是/否
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### PartToList 方法

描述 拷贝部分点到列表当中。

Visual Basic Set lst = PntArea.PartToList(pad, inf, [layerOnOffPad])

C++ HRESULT pPntArea->PartToList(IPnt\_Pad \*pad, IPnt\_Info  
\*inf, ILayerOnOffPad \*layerOnOffPad=NULL, IIDList \*\*lst);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点实体参数开关
inf	Pnt_Info	[in]	点实体图形信息



layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空， 则只保存处于打开状态的图层。 可选；缺省值：Nothing
lst	IDList	[out]	目标列表

### AppendFile 方法

描述 添加一个文件到工作区中。

Visual Basic rtl = PntArea.AppendFile([fName], [szDSN], [szLOG], [szPSW])

C++ HRESULT pPntArea->AppendFile(BSTR fname= NULL,BSTR szDSN= NULL,  
BSTR szLOG= NULL, BSTR szPSW= NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fName	String	[in]	文件名字符串。 可选；如果缺省则弹出对话框选择文件。
szDSN	String	[in]	ODBC 数据源名称。 可选；缺省值：NULL
szLOG	String	[in]	登录号（登录名）。 可选；缺省值：NULL
szPSW	String	[out]	登录口令。 可选；缺省值：NULL
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetFileName 方法

描述 取工作区对应的文件名（不带路径）。

Visual Basic rtl = PntArea.GetFileName(fName)

C++ HRESULT pPntArea->GetFileName(BSTR \*fname, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fName	String	[out]	工作区对应的文件名
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## 3.97、LinArea 对象

描述 LinArea 对象封装了对线工作区的操作。

### 3.97.1、属性：

属性	描述
count	线工作区包含的线物理数
Long	
只读	
logCount	线工作区包含的线逻辑数
Long	
只读	
att	线工作区属性数据
PntAtt	
只读	
extDB	线工作区外挂数据库
ExtDbTool	
只读	
layer	线工作区图层
Layer	
只读	
Type	工作区类型
Enum_Area_Type	
只读	
Name	工作区名称
String	
可读写	
Changed	工作区已经改变标志
Boolean	
可读写	
ShowErrMsg	显示错误信息标志
Boolean	
可读写	
Warning	对不能后悔的操作是否给出警告提示
Boolean	
可读写	
Compact	保存工作区时压缩数据标志
Boolean	
可读写	
Empty	工作区是否为空标志
Boolean	
只读	

LastErrCode	最后错误码
Long	
只读	
rect	图形范围
D_Rect	
只读	

### 3.97.2、方法：

#### GetRect 方法

描述 取线实体图形范围。

Visual Basic rtl = LinArea.GetRect(li, rc)

C++ HRESULT pLinArea->GetRect(long li, ID\_Rect \*\*rc, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
rc	D_Rect	[out]	图元的图形范围
rtl	Integer	[out]	返回：1/0/-1=存在/不存在/被删除

#### GetRectLayer 方法

描述 取线实体图形范围和所在图层号。

Visual Basic rtl = LinArea.GetRectLayer(li, rc, layer)

C++ HRESULT pLinArea->GetRectLayer(  
long li, ID\_Rect \*\*rc, short \*layer, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
rc	D_Rect	[out]	线图元的图形范围
layer	Integer	[out]	线图元所在的图层号
rtl	Integer	[out]	返回：1/0/-1=存在/不存在/被删除

#### GetLayer 方法

描述 取线实体所在图层号。

Visual Basic rtl = LinArea.GetLayer(li, layer)

C++ HRESULT pLinArea->GetLayer(long li, short \*layer, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号

layer	Integer	[out]	线图元所在的图层号
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**GetExistFlag 方法**

描述 取线存在标志。

Visual Basic rtl = LinArea.GetExistFlag(li)

C++ HRESULT pLinArea->GetExistFlag(long li, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**GetAllExistNo 方法**

描述 取所有没有被删除的线号。

Visual Basic Set lst = LinArea.GetAllExistNo()

C++ HRESULT pLinArea->GetAllExistNo(IDList \*\*lst);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[out]	没有被删除的线列表

**GetInfo 方法**

描述 取第 li 条线图形信息。

Visual Basic rtl = LinArea.GetInfo(li, inf)

C++ HRESULT pLinArea->GetInfo(long li, ILin\_Info \*\*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
inf	Lin_Info	[out]	线图形信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UpdateInfo 方法**

描述 更新第 li 条线图形信息。

Visual Basic rtl = LinArea.UpdateInfo(li, inf)

C++ HRESULT pLinArea->UpdateInfo(long li, ILin\_Info \*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号

inf	Lin_Info	[in]	线图形信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetDatLen 方法**

描述 取线数据长度（字节长度）。

Visual Basic retLen = LinArea.GetDatLen(li)

C++ HRESULT pLinArea->GetDatLen(long li, long \*retLen);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
retLen	Long	[out]	返回线数据长度

**Del 方法**

描述 从逻辑上删除第 li 条线实体。

Visual Basic rtl = LinArea.Del(li)

C++ HRESULT pLinArea-> Del(long li, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	删除的线实体号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UnDel 方法**

描述 恢复被删除的线。

Visual Basic rtl = LinArea.UnDel(li)

C++ HRESULT pLinArea->UnDel(long li, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	恢复的线实体号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SavePart 方法**

描述 保存部分线实体。

Visual Basic rtl = LinArea.SavePart(pad, inf, [layerOnOffPad])

C++ HRESULT pLinArea-> SavePart(  
ILin\_Pad \*pad, ILin\_Info \*inf,  
ILayerOnOffPad \*layerOnOffPad=NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Lin_Pad	[in]	线实体参数开关
inf	Lin_Info	[in]	线实体图形信息
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志,若此实体不为空,则只保存处于打开状态的图层。 可选;缺省值:Nothing
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

**CopyPart 方法**

描述 拷贝部分线到 ai1 工作区。

Visual Basic linNum = LinArea.CopyPart(pad, inf, ai1, [layerOnOffPad])

C++ HRESULT pLinArea-> CopyPart (  
ILin\_Pad \*pad, ILin\_Info \*inf, IWorkArea \*ai1,  
ILayerOnOffPad \*layerOnOffPad=NULL, long \*linNum);

参数说明:

参数	类型	出口/入口	描述
pad	Lin_Pad	[in]	线实体参数开关
inf	Lin_Info	[in]	线实体图形信息
ai1	WorkArea	[in]	目的工作区
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志,若此实体不为空,则只拷贝处于打开状态的图层。 可选;缺省值:Nothing
linNum	Long	[out]	成功返回拷贝的线数,失败返回 0

**CopyAll 方法**

描述 拷贝所有线实体到 ai1 工作区。

Visual Basic linNum = LinArea.CopyAll(ai1)

C++ HRESULT pLinArea-> CopyAll (IWorkArea \*ai1, long \*linNum);

参数说明:

参数	类型	出口/入口	描述
ai1	WorkArea	[in]	目的工作区
linNum	Long	[out]	成功返回拷贝的线数,失败返回 0

**CopyList 方法**

描述 拷贝列表中的线到工作区 ai1。

Visual Basic linNum = LinArea.CopyList(lst, ai1, [rtnNo])

C++ HRESULT pLinArea->CopyList(  
IIDList \*lst, IWorkArea \*ai1, short rtnNo, long \*linNum);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	要拷贝的线图元列表
ai1	WorkArea	[in]	目的工作区
rtnNo	Integer	[in]	是否通过 lst 返回新的实体号
linNum	Long	[out]	成功返回拷贝的线数，失败返回 0

### GetDimension 方法

描述 取 li 号线的维数。

Visual Basic dimension = LinArea.GetDimension(li)

C++ HRESULT pLinArea->GetDimension(long li, short \*dimension);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	要拷贝的线图元列表
li	Long	[in]	线实体号
dimension	Integer	[out]	返回线的维数：2=2 维，3=3 维，0=失败，-2=2 维但被删除，-3=3 维但被删除

### Get 方法

描述 取二维线。

Visual Basic rtl = LinArea.Get(li, pos, inf, dim)

C++ HRESULT pLinArea->Get(  
long li, ID\_DotSet \*\*pos, ILin\_Info \*\*inf, short, \*dim, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
pos	D_DotSet	[out]	线坐标
inf	Lin_Info	[out]	线图形信息
dim	Integer	[out]	线实际维数
rtl	Integer	[out]	返回值：1/0/-1 = 成功/失败/被删除

### Append 方法

描述 添加一条二维线。

Visual Basic linNo = LinArea.Append(pos, inf)

C++ HRESULT pLinArea-> Append(  
ID\_DotSet \*pos, ILin\_Info \*inf, long \*linNo);

参数说明：

参数	类型	出口/入口	描述
pos	D_DotSet	[in]	线坐标
inf	Lin_Info	[in]	线图形信息
linNo	Long	[out]	成功返回线实体号 (>0), 失败返回 0

### Update 方法

描述 更新第 li 号线实体。

Visual Basic rtl = LinArea.Update(li, pos, inf)

C++ HRESULT pLinArea-> Update(  
long li, ID\_DotSet \*pos, ILin\_Info \*inf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
pos	D_DotSet	[in]	线坐标
inf	Lin_Info	[in]	线信息
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

### Get3D 方法

描述 取三维线。

Visual Basic rtl = LinArea.Get3D(li, pos, inf, dim)

C++ HRESULT pLinArea->Get3D(  
long li, ID\_3DotSet \*\*pos, ILin\_Info \*\*inf, short \*dim, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
pos	D_3DotSet	[out]	线坐标
inf	Lin_Info	[out]	线图形信息
dim	Integer	[out]	线实际维数
rtl	Integer	[out]	返回值：1/0/-1 = 成功/失败/已删除

### Append3D 方法

描述 添加一条三维线。

Visual Basic linNo = LinArea.Append3D(pos, inf)

C++ HRESULT pLinArea-> Append3D(



---

ID\_3DotSet \*pos, ILin\_Info \*inf, long \*linNo);

参数说明：

参数	类型	出口/入口	描述
pos	D_3DotSet	[in]	线坐标
inf	Lin_Info	[in]	线图形信息
li	Long	[out]	成功返回线号 (>0), 失败返回 0

---

**Update3D 方法**

描述 更新线。

Visual Basic rtl = LinArea.Update3D(li, pos, inf)

C++ HRESULT pLinArea-> Update3D(  
long li, ID\_3DotSet \*pos, ILin\_Info \*inf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
pos	D_3DotSet	[in]	线坐标
inf	Lin_Info	[in]	线信息
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

---

**GetAutoGetBinDatFlag 方法**

描述 取自动取二进制数据字段标志。

Visual Basic flg = LinArea.GetAutoGetBinDatFlag()

C++ HRESULT pLinArea-&gt; GetAutoGetBinDatFlag(short \*flg);

参数说明：

参数	类型	出口/入口	描述
flg	Integer	[out]	返回值：TRUE/FALSE=自动/不自动

---

**SetAutoGetBinDatFlag 方法**

描述 设自动取二进制数据字段标志。

Visual Basic preFlg = LinArea.SetAutoGetBinDatFlag(flgl)

C++ HRESULT pLinArea-&gt; SetAutoGetBinDatFlag(short flgl, short \*preFlgl);

参数说明：

参数	类型	出口/入口	描述
flgl	Integer	[in]	自动取二进制数据字段标志
preFlgl	Integer	[out]	返回值：TRUE/FALSE=自动/不自动

---

**ModParam 方法**

描述 统改线参数。

Visual Basic rtl = LinArea.ModParam(pad0, inf0, pad1, inf1, [layerOnOffPad])

C++ HRESULT pLinArea-> ModParam(  
ILin\_Pad \*pad0, ILin\_Info \*inf0, ILin\_Pad \*pad1, ILin\_Info \*inf1,  
ILayerOnOffPad \*layerOnOffPad=NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad0	Lin_Pad	[in]	查找线实体参数开关
inf0	Lin_Info	[in]	源线实体参数
pad1	Lin_Pad	[in]	修改线实体参数开关
inf1	Lin_Info	[in]	新的线实体参数
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空， 则只修改处于打开状态的图层。 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetMapParam 方法**

描述 取地图参数。

Visual Basic rtl = LinArea.GetMapParam(mpar)

C++ HRESULT pLinArea-> GetMapParam(  
IMap\_Para \*\*mpar, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
mpar	Map_Para	[out]	地图参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SetMapParam 方法**

描述 设地图参数。

Visual Basic rtl = LinArea.SetMapParam(mpar)

C++ HRESULT pLinArea-> SetMapParam(  
IMap\_Para \*mpar, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
mpar	Map_Para	[in]	地图参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ModMapParam 方法**

描述 修改地图参数。

Visual Basic rtl = LinArea.ModMapParam()

C++ HRESULT pLinArea-> ModMapParam(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CopyMapParam 方法**

描述 拷贝地图参数到工作区 ai1。

Visual Basic rtl = LinArea.CopyMapParam(ai1)

C++ HRESULT pLinArea-> CopyMapParam(  
IWorkArea \*ai1, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai1	WorkArea	[in]	目的工作区
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Transform 方法**

描述 图形变换。

Visual Basic rtl = LinArea.Transform(tp)

C++ HRESULT pLinArea-> Transform(ITran\_Par \*tp, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
tp	Tran_Par	[in]	图形变换参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetLength 方法**

描述 取线长度。

Visual Basic rtl = LinArea.GetLength(li, length)

C++ HRESULT pLinArea-> GetLength(  
long li,double \*length,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线实体号
length	Double	[out]	线长度
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Near 方法**

描述 查找最近的线。

Visual Basic linNo = LinArea.Near(xy, [layerOnOffPad], [dis])

C++ HRESULT pLinArea-> Near(  
ID\_Dot \*xy, ILayerOnOffPad \*layerOnOffPad=NULL,  
double \*dis=NULL, long \*linNo);

参数说明：

参数	类型	出口/入口	描述
xy	D_Dot	[in]	点位置，为图形实际坐标
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空，则只在处于打开状态的图层中查找。 可选；缺省值：Nothing
dis	Double	[out]	点线距离
linNo	Long	[out]	成功返回 TRUE，失败返回 FALSE

**dbGetAreaDSType 方法**

描述 取工作区数据源类型。

Visual Basic type = LinArea.dbGetAreaDSType()

C++ HRESULT pLinArea-> dbGetAreaDSType(Enum\_GisDS\_Type \*type);

参数说明：

参数	类型	出口/入口	描述
type	Enum_GisDS_Type	[out]	返回工作区数据源类型： PC_GIS/NET_GIS=单机数据源/网络数据源

**dbGetAreaDSName 方法**

描述 取工作区数据源名称。

Visual Basic rtl = LinArea.dbGetAreaDSName(szDSN)

C++ HRESULT pLinArea->dbGetAreaDSName(BSTR \*szDSN, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
szDSN	String	[out]	工作区数据源名称
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbLockAll 方法**

描述 全锁工作区中的数据。

Visual Basic rtl = LinArea.dbLockAll()

C++ HRESULT pLinArea-> dbLockAll(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbUnLockAll 方法**

描述 解除全锁。

Visual Basic rtl = LinArea.dbUnLockAll()

C++ HRESULT pLinArea-> dbUnLockAll(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbLockRect 方法**

描述 锁工作区中矩形内的数据。

Visual Basic rtl = LinArea.dbLockRect(rc)

C++ HRESULT pLinArea-> dbLockRect(ID\_Rect \*rc, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rc	D_Rect	[in]	被封锁的数据范围
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbUnLockRect 方法**

描述 解除矩形锁。

Visual Basic rtl = LinArea.dbUnLockRect()

C++ HRESULT pLinArea-> dbUnLockRect(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbGetLockRange 方法**

描述 取被自己封锁的数据范围。

Visual Basic rtl = LinArea.dbGetLockRange(rc)

C++ HRESULT pLinArea-> dbGetLockRange(ID\_Rect \*\*rc, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
rc	D_Rect	[out]	被自己封锁的数据范围
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dbGetLockNumByOther 方法

描述 取被其他用户封锁的矩形块数。

Visual Basic num = LinArea.dbGetLockNumByOther()

C++ HRESULT pLinArea-> dbGetLockNumByOther(long \*num);

参数说明：

参数	类型	出口/入口	描述
num	Long	[out]	返回值：-1-失败，0-其他用户未封锁，>0-其他用户封锁的矩形块数

#### dbGetLockRangeByOther 方法

描述 取被其他用户封锁的矩形范围。

Visual Basic num = LinArea.dbGetLockRangeByOther(rc)

C++ HRESULT pLinArea-> dbGetLockRangeByOther(  
ID\_RectSet \*\*rc, long \*num);

参数说明：

参数	类型	出口/入口	描述
rc	D_RectSet	[out]	被其他用户封锁的矩形范围
num	Long	[out]	返回值：-1-失败，0-其他用户未封锁，>0-其他用户封锁的矩形块数

#### dbLockSingle 方法

描述 封锁单个图元。

Visual Basic rtl = LinArea.dbLockSingle(i)

C++ HRESULT pLinArea-> dbLockSingle(long i, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
i	Long	[in]	封锁的图元号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dbUnLockSingle 方法

描述 取消单个图元的封锁。

Visual Basic rtl = LinArea.dbUnLockSingle(li)

C++ HRESULT pLinArea-> dbUnLockSingle(long li, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	取消封锁的图元号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dbBeginTransaction 方法

描述 事务开始。

Visual Basic rtl = LinArea.dbBeginTransaction([newTrans])

C++ HRESULT pLinArea-> dbBeginTransaction(short newTrans=0, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
newTrans	Integer	[in]	newTrans=0 时只增加嵌套次数，=1 时则强制提交以前开始的事务，并开始一个新的事务。 可选；缺省值：0
rtl	Integer	[out]	成功返回事务嵌套次数，失败返回 0

#### dbRollbackTransaction 方法

描述 事务回滚。忽略事务嵌套次数，回滚到最外层事务开始的地方，并将嵌套次数设 0。

Visual Basic rtl = LinArea.dbRollbackTransaction()

C++ HRESULT pLinArea-> dbRollbackTransaction(short \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Integer	[out]	成功返回 TRUE，失败返回 FALSE

#### dbCommitTransaction 方法

描述 事务提交。

Visual Basic rtl = LinArea.dbCommitTransaction([commitNow])

C++ HRESULT pLinArea-> dbCommitTransaction(  
short commitNow=0, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

commitNow	Integer	[in]	commitNow!=0 则马上提交并将嵌套次数置 0，否则只有嵌套次数减到 0 时才提交。 可选；缺省值：0
rtl	Integer	[out]	成功返回 TRUE，失败返回 FALSE

**dbCreateFieldIndex 方法**

描述 创建属性字段索引（只有拥有者才能创建索引）

Visual Basic rtl = LinArea.dbCreateFieldIndex(stru,  
ndxFldFlag, IsUnique, IsClustered, IsPadIndex)

C++ HRESULT pLinArea->dbCreateFieldIndex(  
IRecord\_Head \*stru, IRepFldFlag \*ndxFldFlag, short IsUnique,  
short IsClustered, short IsPadIndex, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
stru	Record_Head	[in]	属性结构（主要用其中的字段名称）
ndxFldFlag	RepFldFlag	[in]	创建字段索引标识（用来对应于创建索引的字段，ndxFldFlag[i]=1：表示字段 i 创建索引；=0：表示字段 i 不创建索引）
IsUnique	Integer	[in]	创建唯一索引标识：1/0-是/否
IsClustered	Integer	[in]	创建聚簇索引标识：1/0-是/否
IsPadIndex	Integer	[in]	1/0-是/否
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**PartToList 方法**

描述 拷贝部分线到列表当中。

Visual Basic Set lst = LinArea.PartToList(pad, inf, [layerOnOffPad])

C++ HRESULT pLinArea->PartToList (ILin\_Pad \*pad, ILin\_Info  
\*inf, IlayerOnOffPad \*layerOnOffPad=NULL, IIDList \*\*lst);

参数说明：

参数	类型	出口/入口	描述
pad	Lin_Pad	[in]	线实体参数开关
inf	Lin_Info	[in]	线实体图形信息
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空，则只拷贝处于打开状态的图层。 可选；缺省值：Nothing



lst	IDList	[out]	目标列表
-----	--------	-------	------

**AppendFile 方法**

描述 添加一个文件到工作区中。

Visual Basic rtl = LinArea.AppendFile([fName], [szDSN], [szLOG], [szPSW])

C++ HRESULT pLinArea->AppendFile(BSTR fname= NULL,BSTR szDSN= NULL,  
BSTR szLOG= NULL, BSTR szPSW= NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fName	String	[in]	文件名字符串。 可选；如果缺省则弹出对话框选择文件。
szDSN	String	[in]	ODBC 数据源名称。 可选；缺省值：NULL
szLOG	String	[in]	登录号（登录名）。 可选；缺省值：NULL
szPSW	String	[out]	登录口令。 可选；缺省值：NULL
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetFileName 方法**

描述 取工作区对应的文件名（不带路径）。

Visual Basic rtl = LinArea.GetFileName(fName)

C++ HRESULT pLinArea->GetFileName(BSTR \*fname, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fName	String	[out]	工作区对应的文件名
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**3.98、RegArea 对象**

描述 RegArea 对象封装了对区工作区的操作。

**3.98.1、属性：**

属性	描述
----	----

count	区物理数
Long	
只读	
logCount	区逻辑数
Long	
只读	
lin	线数据
LinData	
只读	
nod	结点数据
NodData	
只读	
regAtt	区属性数据
RegAtt	
只读	
linAtt	线属性数据
LinAtt	
只读	
nodAtt	结点属性数据
NodAtt	
只读	
extDB	区工作区外挂数据库
ExtDbTool	
只读	
layer	区工作区图层
Layer	
只读	
Type	工作区类型
Enum_Area_Type	
只读	
Name	工作区名称
String	
可读写	
Changed	工作区已经改变标志
Boolean	
可读写	

ShowErrMsg	显示错误信息标志
Boolean	
可读写	
Warning	对不能后悔的操作是否给出警告提示
Boolean	
可读写	
Compact	保存工作区时压缩数据标志
Boolean	
可读写	
Empty	工作区是否为空标志
Boolean	
只读	
LastErrCode	最后错误码
Long	
只读	
rect	图形范围
D_Rect	
只读	

### 3.98.2、方法：

#### GetRect 方法

描述 取区实体图形范围。

Visual Basic rtl = RegArea.GetRect(ri, rc)

C++ HRESULT pRegArea->GetRect(long ri, ID\_Rect \*\*rc, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	区实体号
rc	D_Rect	[out]	图元的图形范围
rtl	Integer	[out]	返回：1/0/-1=存在/不存在/被删除

#### GetRectLayer 方法

描述 取区实体图形范围和所在图层号。

Visual Basic rtl = RegArea.GetRectLayer(ri, rc, layer)

C++ HRESULT pRegArea->GetRectLayer(  
long ri, ID\_Rect \*\*rc, short \*layer, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

ri	Long	[in]	区实体号
rc	D_Rect	[out]	区图元的图形范围
layer	Integer	[out]	区图元所在的图层号
rtl	Integer	[out]	返回：1/0/-1=存在/不存在/被删除

**GetLayer 方法**

描述 取区实体所在图层号。

Visual Basic rtl = RegArea.GetLayer(ri, layer)

C++ HRESULT pRegArea->GetLayer(long ri, short \*layer, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	区实体号
layer	Integer	[out]	区图元所在的图层号
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**GetExistFlag 方法**

描述 取区存在标志。

Visual Basic rtl = RegArea.GetExistFlag(ri)

C++ HRESULT pRegArea-> GetExistFlag(long ri, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	区实体号
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**GetAllExistNo 方法**

描述 取所有没有被删除的区号。

Visual Basic Set lst = RegArea.GetAllExistNo()

C++ HRESULT pRegArea-> GetAllExistNo (IIDList \*\*lst);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[out]	没有被删除的区列表

**GetInfo 方法**

描述 取第 ri 号区图形信息。

Visual Basic rtl = RegArea.GetInfo(ri, inf)

C++ HRESULT pRegArea-> GetInfo(long ri, IReg\_Info \*\*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	区实体号
inf	Reg_Info	[out]	区图形信息
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**UpdateInfo 方法**

描述 更新第 ri 号区图形信息。

Visual Basic rtl = RegArea.UpdateInfo(ri, inf)

C++ HRESULT pRegArea->UpdateInfo(long ri, IReg\_Info \*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	区实体号
inf	Reg_Info	[in]	区图形信息
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

**GetDatLen 方法**

描述 取 rdat 长度。

Visual Basic datLen = RegArea.GetDatLen(ri)

C++ HRESULT pRegArea->GetDatLen(long ri, long \*datLen);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	区实体号
datLen	Long	[out]	返回 rdat 长度

**SavePart 方法**

描述 保存部分区实体。

Visual Basic rtl = RegArea.SavePart(pad, inf, [layerOnOffPad])

C++ HRESULT pRegArea->SavePart(  
IReg\_Pad \*pad, IReg\_Info \*inf,  
ILayerOnOffPad \*layerOnOffPad=NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Reg_Pad	[in]	区实体参数开关
inf	Reg_Info	[in]	区实体图形信息
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志,若此实体不为空, 则只保存处于打开状态的图层。 可选；缺省值：Nothing

rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

**CopyPart 方法**

描述 拷贝部分区到 ai1 工作区。

Visual Basic regNum = RegArea.CopyPart(pad, inf, ai1, [layerOnOffPad])

C++ HRESULT pRegArea->CopyPart (  
IReg\_Pad \*pad, IReg\_Info \*inf, IRegArea \*ai1,  
ILayerOnOffPad \*layerOnOffPad=NULL, long \*regNum);

参数说明：

参数	类型	出口/入口	描述
pad	Reg_Pad	[in]	区实体参数开关
inf	Reg_Info	[in]	区实体图形信息
ai1	RegArea	[in]	目的工作区
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空， 则只拷贝处于打开状态的图层。 可选；缺省值：Nothing
regNum	Long	[out]	成功返回拷贝的区数，失败返回 0

**CopyList 方法**

描述 拷贝列表中的区到工作区 ai1。

Visual Basic regNum = RegArea.CopyList(lst, ai1, [rtnNo])

C++ HRESULT pRegArea->CopyList(  
IIDList \*lst, IRegArea \*ai1, short rtnNo, long \*regNum);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	要拷贝的区图元列表
ai1	RegArea	[in]	目的工作区
rtnNo	Integer	[in]	是否通过 lst 返回新的实体号
regnum	Long	[out]	成功返回拷贝的区数，失败返回 0

**CopyAll 方法**

描述 拷贝未删除的区,包括这些区所用弧段（网段）及其所连节点。

Visual Basic regNum = RegArea.CopyAll(ai1)

C++ HRESULT pRegArea->CopyAll (IRegArea \*ai1, long \*regNum);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

ai1	RegArea	[in]	目的工作区
regNum	Long	[out]	成功返回拷贝的区数，失败返回 0

### CopyArea 方法

描述 拷贝 ai 工作区中所有未删除的信息,包括独立弧段,独立节点。

Visual Basic rtl = RegArea.CopyArea(ai1)

C++ HRESULT pRegArea->CopyArea(IRegArea \*ai1, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai1	RegArea	[in]	目的工作区
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### Get 方法

描述 取区域。

Visual Basic rtl = RegArea.Get(ri, rdat, inf)

C++ HRESULT pRegArea->Get(  
long ri, ILONGList \*\*rdat, IReg\_Info \*\*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	区实体号
rdat	LONGList	[out]	区弧段号集合
inf	Reg_Info	[out]	区图形信息
rtl	Integer	[out]	返回值：1/0/-1 = 成功/失败/被删除

### Append 方法

描述 添加区域。

Visual Basic regNo = RegArea.Append(rdat, inf)

C++ HRESULT pRegArea->Append(  
ILONGList \*rdat, IReg\_Info \*inf, long \*regNo);

参数说明：

参数	类型	出口/入口	描述
rdat	LONGList	[in]	区弧段号集合
inf	Lin_Info	[in]	区图形信息
regNo	Long	[out]	成功返回区实体号 (>0)，失败返回 0

### Update 方法

描述 更新第 ri 号区实体。

Visual Basic rtl = RegArea.Update(ri, rdat, inf)

C++ HRESULT pRegArea->Update(  
long ri, ILONGList \*rdat, IReg\_Info \*inf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	区实体号
rdat	LONGList	[in]	区弧段号集合
inf	Reg_Info	[in]	区图形信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## Del 方法

描述 从逻辑上删除第 ri 号区实体。

Visual Basic rtl = RegArea.Del(ri)

C++ HRESULT pRegArea->Del(long ri, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	删除的区实体号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## UnDel 方法

描述 恢复被删除的区。

Visual Basic rtl = RegArea.UnDel(ri)

C++ HRESULT pRegArea->UnDel(long ri, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	恢复的区实体号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## GetCircleNum 方法

描述 取区圈数。

Visual Basic cirNum = RegArea.GetCircleNum(rdat)

C++ HRESULT pRegArea->GetCircleNum(ILONGList \*rdat, short \*cirNum);

参数说明：

参数	类型	出口/入口	描述
rdat	LONGList	[in]	区弧段号集合
cirNum	Integer	[out]	区圈数



**GetEdge 方法**

描述 取区边界坐标。

Visual Basic rtl = RegArea.GetEdge(rdat, xy, ne);

C++ HRESULT pRegArea->GetEdge(ILONGList \*rdat,  
ID\_DotSet \*\*xy, ILONGList \*\*ne, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
rdat	LONGList	[in]	区弧段号集合
xy	D_DotSet	[out]	区边界坐标
ne	LONGList	[ou]	区域边界封闭圈点数数组
rtl	Integer	[out]	成功返回 1，失败返回 0

**GetLongEdge 方法**

描述 取区边界坐标。

Visual Basic rtl = RegArea.GetLongEdge(rdat, xy, ne, xy0, rate);

C++ HRESULT pRegArea->GetLongEdge(ILONGList \*rdat,  
IL\_DotSet \*\*xy, ILONGList \*\*ne, ID\_Dot xy0, double rate, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
rdat	LONGList	[in]	区弧段号集合
xy	L_DotSet	[out]	区边界坐标
ne	LONGList	[ou]	区域边界封闭圈点数数组
xy0	D_Dot	[in]	原点坐标
rate	Double	[in]	缩放比例
rtl	Integer	[out]	成功返回 1，失败返回 0

**ClearTopDat 方法**

描述 清除拓扑数据。

Visual Basic rtl = RegArea.ClearTopDat()

C++ HRESULT pRegArea->ClearTopDat(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UnionArc 方法**

描述 合并子区弧段。

Visual Basic rtl = RegArea.UnionArc(rdat)

C++ HRESULT pRegArea->UnionArc(ILONGList \*rdat, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rdat	LONGList	[in]	区弧段号集合
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### UnionAtoB 方法

描述 把区域 ra 合并到区域 rb。

Visual Basic rtl = RegArea.UnionAtoB(ra, rb, [delCommArc])

C++ HRESULT pRegArea->UnionAtoB(long ra, long rb, short delCommArc=0, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ra	Long	[in]	区实体号
rb	Long	[in]	区实体号
delCommArc	Integer	[in]	删除公共弧段标志 (0 : 保留, 1 : 删除), 可选; 缺省值: 0
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### Split 方法

描述 分裂区。

Visual Basic rtl = RegArea.Split(ri, clpArc)

C++ HRESULT pRegArea->Split(long ri, ILONGList \*clpArc, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	分裂的区号
clpArc	LONGList	[in]	分裂区的弧段
rtl	Integer	[out]	成功返回 1，失败返回 0

### GetPerimeter 方法

描述 取区周长。

Visual Basic rtl = RegArea.GetPerimeter(ri, perimeter)

C++ HRESULT pRegArea->GetPerimeter(long ri, double \*perimeter, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	区号

perimeter	Double	[out]	区周长
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

**GetArea 方法**

描述 取区面积。

Visual Basic rtl = RegArea.GetArea(ri, area)

C++ HRESULT pRegArea->GetArea(  
long ri, double \*area, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	区号
area	Double	[out]	区面积
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

**DistOfPntToReg 方法**

描述 计算点到区之间的距离。

Visual Basic rtl = RegArea.DistOfPntToReg(pnt, ri, minDis, maxDis)

C++ HRESULT pRegArea->DistOfPntToReg(  
ID\_Dot \*pnt, long ri, double \*minDis,  
double \*maxDis, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pnt	ID_Dot	[in]	点坐标
ri	Long	[in]	区号
minDis	Double	[out]	最小距离
maxDis	Double	[out]	最大距离
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

**MinDistOfPntToReg 方法**

描述 计算点到区之间的最小距离。

Visual Basic minDis = RegArea.MinDistOfPntToReg(pnt, ri)

C++ HRESULT pRegArea->MinDistOfPntToReg (  
ID\_Dot \*pnt, long ri, double \*minDis);

参数说明：

参数	类型	出口/入口	描述
pnt	ID_Dot	[in]	点坐标
ri	Long	[in]	区号

minDis	Double	[out]	返回最小距离
--------	--------	-------	--------

**MaxDistOfPntToReg 方法**

描述 计算点到区之间的最大距离。

Visual Basic minDis = RegArea.MaxDistOfPntToReg(pnt, ri)

C++ HRESULT pRegArea->MaxDistOfPntToReg (  
ID\_Dot \*pnt, long ri, double \*maxDis);

参数说明：

参数	类型	出口/入口	描述
pnt	D_Dot	[in]	点坐标
ri	Long	[in]	区号
maxDis	Double	[out]	返回最小距离

**Seek 方法**

描述 查找包含点的区。

Visual Basic regNo = RegArea.Seek(xy, [layerOnOffPad])

C++ HRESULT pRegArea->Seek(  
ID\_Dot \*xy, ILayerOnOffPad \*layerOnOffPad=NULL, long \*regNo);

参数说明：

参数	类型	出口/入口	描述
xy	D_Dot	[in]	点位置，xy 为图形实际坐标
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此项参数不为 NULL， 则只在处于打开状态的图层中查找。 可选；缺省值：Nothing
regNo	Long	[out]	成功返回区号 (>0)，失败返回 0

**MathAllSubreg 方法**

描述 匹配所有的子区。

Visual Basic rtl = RegArea.MathAllSubreg()

C++ HRESULT pRegArea->MathAllSubreg(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**MathSubreg 方法**

描述 匹配子区。

Visual Basic rtl = RegArea.MathSubreg(ri)

C++ HRESULT pRegArea->MathSubreg(long ri, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	区号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### ResetLRPolyofArcTop 方法

描述 根据区数据重新设置弧段拓扑的左右多边形，该函数根据各区域的弧段之间的连接关系，重新确定各条弧段的左右区号。

Visual Basic rtl = RegArea.ResetLRPolyofArcTop()

C++ HRESULT pRegArea->ResetLRPolyofArcTop(long \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Long	[out]	返回值 1/0/<0 - 成功/失败/失败（返回值的绝对值=拓扑错误数）

### ResetLinTopLRPolyNo 方法

描述 重新设置弧段左右多边形号。

Visual Basic rtl = RegArea.ResetLinTopLRPolyNo()

C++ HRESULT pRegArea->ResetLinTopLRPolyNo(long \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Long	[out]	返回值 1/0/<0 - 成功/失败/失败（返回值的绝对值=拓扑错误数）

### IsDotIn 方法

描述 判断点是否在区域内。

Visual Basic rtl = RegArea.IsDotIn(dot, ri, flg)

C++ HRESULT pRegArea->IsDotIn(ID\_Dot \*dot, long ri, short flg, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
dot	D_Dot	[in]	坐标点
ri	Long	[in]	区号
flg	Integer	[in]	是否判断内圈标志，1/0：判断/不判断； 即 flg=1 时，要严格判断点是否落在区域内，若落在区域的空洞内，也认为是落在区域外边

rtl	Integer	[out]	dot 点在区域内返回 2，在边线上返回 1，在区域外返回 0
-----	---------	-------	---------------------------------

**CalculateArea 方法**

**描述** 计算区面积；该函数要求 xy 构成的多边形封闭，返回面积<0 时表示该区逆时针旋转，>0 时表示该区顺时针旋转。

**Visual Basic** rtl = RegArea.CalculateArea(rdat, area, par)

**C++** HRESULT pRegArea->CalculateArea(  
ILONGList \*rdat, double \*area, double \*par, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
rdat	LONGList	[in]	区弧段数据
area	Double	[out]	区域面积，若包含子区，则扣除子区面积
par	Double	[out]	返回母区边界周长
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetAutoGetBinDatFlag 方法**

**描述** 取自动取二进制数据字段标志。

**Visual Basic** flg = RegArea.GetAutoGetBinDatFlag()

**C++** HRESULT pRegArea-> GetAutoGetBinDatFlag(short \*flg);

**参数说明：**

参数	类型	出口/入口	描述
flg	Integer	[out]	返回值：TRUE/FALSE=自动/不自动

**SetAutoGetBinDatFlag 方法**

**描述** 设自动取二进制数据字段标志。

**Visual Basic** preFlg = RegArea.SetAutoGetBinDatFlag(flgl)

**C++** HRESULT pRegArea-> SetAutoGetBinDatFlag(short flgl, short \*preFlgl);

**参数说明：**

参数	类型	出口/入口	描述
flgl	Integer	[in]	自动取二进制数据字段标志
preFlgl	Integer	[out]	返回值：TRUE/FALSE=自动/不自动

**ModParam 方法**

**描述** 统改区参数。

**Visual Basic** rtl = RegArea.ModParam(pad0, inf0, pad1, inf1, [layerOnOffPad])

**C++** HRESULT pRegArea-> ModParam(

IReg\_Pad \*pad0, IReg\_Info \*inf0, IReg\_Pad \*pad1, IReg\_Info \*inf1,  
ILayerOnOffPad \*layerOnOffPad=NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad0	Reg_Pad	[in]	查找区实体参数开关
inf0	Reg_Info	[in]	源区实体参数
pad1	Reg_Pad	[in]	修改区实体参数开关
inf1	Reg_Info	[in]	新的区实体参数
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空， 则只修改处于打开状态的图层。 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetMapParam 方法

描述 取地图参数。

Visual Basic rtl = RegArea.GetMapParam(mpar)

C++ HRESULT pRegArea-> GetMapParam(  
IMap\_Para \*\*mpar, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
mpar	Map_Para	[out]	地图参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### SetMapParam 方法

描述 设地图参数。

Visual Basic rtl = RegArea.SetMapParam(mpar)

C++ HRESULT pRegArea-> SetMapParam(  
IMap\_Para \*mpar, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
mpar	Map_Para	[in]	地图参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### ModMapParam 方法

描述 修改地图参数。

Visual Basic rtl = RegArea.ModMapParam()

C++ HRESULT pRegArea-> ModMapParam(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### CopyMapParam 方法

描述 拷贝地图参数到工作区 ai1。

Visual Basic rtl = RegArea.CopyMapParam(ai1)

C++ HRESULT pRegArea-> CopyMapParam(  
IWorkArea \*ai1, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai1	WorkArea	[in]	目的工作区
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### Transform 方法

描述 图形变换。

Visual Basic rtl = RegArea.Transform(tp)

C++ HRESULT pRegArea-> Transform(ITran\_Par \*tp, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
tp	Tran_Par	[in]	图形变换参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### dbGetAreaDSType 方法

描述 取工作区数据源类型。

Visual Basic type = RegArea.dbGetAreaDSType()

C++ HRESULT pRegArea-> dbGetAreaDSType(Enum\_GisDS\_Type \*type);

参数说明：

参数	类型	出口/入口	描述
type	Enum_GisDS_Type	[out]	返回工作区数据源类型： PC_GIS/NET_GIS=单机数据源/网络数据源

### dbGetAreaDSName 方法

描述 取工作区数据源名称。

Visual Basic rtl = RegArea.dbGetAreaDSName(szDSN)

C++ HRESULT pRegArea->dbGetAreaDSName(BSTR \*szDSN, short \*rtl);



参数说明：

参数	类型	出口/入口	描述
szDSN	String	[out]	工作区数据源名称
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dbLockAll 方法

描述 全锁工作区中的数据。

Visual Basic rtl = RegArea.dbLockAll()

C++ HRESULT pRegArea-> dbLockAll(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dbUnLockAll 方法

描述 解除全锁。

Visual Basic rtl = RegArea.dbUnLockAll()

C++ HRESULT pRegArea-> dbUnLockAll(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dbLockRect 方法

描述 锁工作区中矩形内的数据。

Visual Basic rtl = RegArea.dbLockRect(rc)

C++ HRESULT pRegArea-> dbLockRect(ID\_Rect \*rc, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rc	D_Rect	[in]	被封锁的数据范围
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dbUnLockRect 方法

描述 解除矩形锁。

Visual Basic rtl = RegArea.dbUnLockRect()

C++ HRESULT pRegArea-> dbUnLockRect(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbGetLockRange 方法**

描述 取被自己封锁的数据范围。

Visual Basic rtl = RegArea.dbGetLockRange(rc)

C++ HRESULT pRegArea-> dbGetLockRange(ID\_Rect \*\*rc,short \*rtl);

参数说明：

参数	类型	出口/入口	描述
rc	D_Rect	[out]	被自己封锁的数据范围
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbGetLockNumByOther 方法**

描述 取被其他用户封锁的矩形块数。

Visual Basic num = RegArea.dbGetLockNumByOther()

C++ HRESULT pRegArea-> dbGetLockNumByOther(long \*num);

参数说明：

参数	类型	出口/入口	描述
num	Long	[out]	返回值：-1-失败，0-其他用户未封锁，>0-其他用户封锁的矩形块数

**dbGetLockRangeByOther 方法**

描述 取被其他用户封锁的矩形范围。

Visual Basic num = RegArea.dbGetLockRangeByOther(rc)

C++ HRESULT pRegArea-> dbGetLockRangeByOther(  
ID\_RectSet \*\*rc, long \*num);

参数说明：

参数	类型	出口/入口	描述
rc	D_RectSet	[out]	被其他用户封锁的矩形范围
num	Long	[out]	返回值：-1-失败，0-其他用户未封锁，>0-其他用户封锁的矩形块数

**dbLockSingle 方法**

描述 封锁单个图元。

Visual Basic rtl = RegArea.dbLockSingle(ri)

C++ HRESULT pRegArea-> dbLockSingle(long ri, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	封锁的图元号

rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

**dbUnLockSingle 方法**

描述 取消单个图元的封锁。

Visual Basic rtl = RegArea.dbUnLockSingle(ri)

C++ HRESULT pRegArea-> dbUnLockSingle(long ri, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	取消封锁的图元号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbBeginTransaction 方法**

描述 事务开始。

Visual Basic rtl = RegArea.dbBeginTransaction([newTrans])

C++ HRESULT pRegArea-> dbBeginTransaction(short newTrans=0, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
newTrans	Integer	[in]	newTrans=0 时只增加嵌套次数，=1 时则强制提交以前开始的事务，并开始一个新的事务。 可选；缺省值：0
rtl	Integer	[out]	成功返回事务嵌套次数，失败返回 0

**dbRollbackTransaction 方法**

描述 事务回滚，忽略事务嵌套次数，回滚到最外层事务开始的地方，并将嵌套次数设 0。

Visual Basic rtl = RegArea.dbRollbackTransaction()

C++ HRESULT pRegArea-> dbRollbackTransaction(short \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Integer	[out]	成功返回 TRUE，失败返回 FALSE

**dbCommitTransaction 方法**

描述 事务提交。

Visual Basic rtl = RegArea.dbCommitTransaction([commitNow])

C++ HRESULT pRegArea-> dbCommitTransaction(  
short commitNow=0, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
commitNow	Integer	[in]	commitNow!=0 则马上提交并将嵌套次数置 0，否则只有嵌套次数减到 0 时才提交。 可选；缺省值：0
rtl	Integer	[out]	成功返回 TRUE，失败返回 FALSE

### dbCreateFieldIndex 方法

描述 创建属性字段索引（只有拥有者才能创建索引）。

Visual Basic rtl = RegArea.dbCreateFieldIndex(stru,  
ndxFldFlag, IsUnique, IsClustered, IsPadIndex)

C++ HRESULT pRegArea->dbCreateFieldIndex(  
IRecord\_Head \*stru, IRepFldFlag \*ndxFldFlag, short IsUnique,  
short IsClustered, short IsPadIndex, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
stru	Record_Head	[in]	属性结构( 主要用其中的字段名称 )
ndxFldFlag	RepFldFlag	[in]	创建字段索引标识（用来对应于创建索引的字段，ndxFldFlag[i]=1：表示字段 i 创建索引；=0:表示字段 i 不创建索引）
IsUnique	Integer	[in]	创建唯一索引标识：1/0-是/否
IsClustered	Integer	[in]	创建聚簇索引标识：1/0-是/否
IsPadIndex	Integer	[in]	1/0-是/否
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### PartToList 方法

描述 拷贝部分区到列表当中。

Visual Basic Set lst = RegArea.PartToList(pad, inf, [layerOnOffPad])

C++ HRESULT pRegArea-> PartToList(IReg\_Pad \*pad, IReg\_Info \*inf,  
ILayerOnOffPad \*layerOnOffPad=NULL, IIDList \*\*lst);

参数说明：

参数	类型	出口/入口	描述
pad	Reg_Pad	[in]	区实体参数开关
inf	Reg_Info	[in]	区实体图形信息

layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空， 则只拷贝处于打开状态的图层。 可选；缺省值：Nothing
lst	IDList	[out]	目标列表

#### AppendFile 方法

描述 添加一个文件到工作区中。

Visual Basic rtl = RegArea.AppendFile([fName], [szDSN], [szLOG], [szPSW])

C++ HRESULT pRegArea->AppendFile(BSTR fname= NULL,BSTR szDSN= NULL,  
BSTR szLOG= NULL, BSTR szPSW= NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fName	String	[in]	文件名字符串。 可选；如果缺省则弹出对话框选择文件。
szDSN	String	[in]	ODBC 数据源名称。 可选；缺省值：NULL
szLOG	String	[in]	登录号（登录名）。 可选；缺省值：NULL
szPSW	String	[out]	登录口令。 可选；缺省值：NULL
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### GetFileName 方法

描述 取工作区对应的文件名（不带路径）。

Visual Basic rtl = RegArea.GetFileName(fName)

C++ HRESULT pRegArea->GetFileName(BSTR \*fname, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fName	String	[out]	工作区对应的文件名
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### 3.99、NetArea 对象

描述 NetArea 对象封装了对网工作区的操作。

#### 3.99.1、属性：

属性	描述
count	网物理数
Long	
只读	
logCount	网逻辑数
Long	
只读	
lin	线数据
LinData	
只读	
nod	结点数据
NodData	
只读	
netAtt	网属性数据
NetAtt	
只读	
linAtt	线属性数据
LinAtt	
只读	
nodAtt	结点属性数据
NodAtt	
只读	
extDB	网工作区外挂数据库
ExtDbTool	
只读	
layer	网工作区图层
Layer	
只读	
path	路径数据
PathData	
只读	
Type	工作区类型
Enum_Area_Type	
只读	
Name	工作区名称
String	
可读写	

Changed	工作区已经改变标志
Boolean	
可读写	
ShowErrMsg	显示错误信息标志
Boolean	
可读写	
Warning	对不能后悔的操作是否给出警告提示
Boolean	
可读写	
Compact	保存工作区时压缩数据标志
Boolean	
可读写	
Empty	工作区是否为空标志
Boolean	
只读	
LastErrCode	最后错误码
Long	
只读	
rect	图形范围
D_Rect	
只读	

### 3.99.2、方法：

#### GetRect 方法

描述 取网实体图形范围。

Visual Basic rtl = NetArea.GetRect(ni, rc)

C++ HRESULT pNetArea->GetRect(long ni, ID\_Rect \*\*rc, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	网实体号
rc	D_Rect	[out]	图元的图形范围
rtl	Integer	[out]	返回：1/0/-1=存在/不存在/被删除

#### GetRectLayer 方法

描述 取网实体图形范围和所在图层号。

Visual Basic rtl = NetArea.GetRectLayer(ni, rc, layer)

C++ HRESULT pNetArea->GetRectLayer(

```
long ni, ID_Rect **rc, short *layer, short *rtl);
```

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	网实体号
rc	D_Rect	[out]	网图元的图形范围
layer	Integer	[out]	网图元所在的图层号
rtl	Integer	[out]	返回：1/0/-1=存在/不存在/被删除

### GetLayer 方法

描述 取网实体所在图层号。

Visual Basic rtl = NetArea.GetLayer(ni, layer)

C++ HRESULT pNetArea->GetLayer(long ni, short \*layer, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	网实体号
layer	Integer	[out]	网图元所在的图层号
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

### GetExistFlag 方法

描述 取网存在标志。

Visual Basic rtl = NetArea.GetExistFlag(ni)

C++ HRESULT pNetArea-> GetExistFlag(long ni, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	网实体号
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

### GetAllExistNo 方法

描述 取所有没有被删除的网号。

Visual Basic lst = NetArea.GetAllExistNo()

C++ HRESULT pNetArea-> GetAllExistNo (IIDList \*\*lst);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[out]	没有被删除的网列表

### GetInfo 方法

描述 取第 ni 号网图形信息。



Visual Basic rtl = NetArea.GetInfo(ni, inf)

C++ HRESULT pNetArea-> GetInfo(long ni, INet\_Info \*\*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	网实体号
inf	Net_Info	[out]	网图形信息
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

### UpdateInfo 方法

描述 更新第 ni 号网图形信息。

Visual Basic rtl=NetArea.UpdateInfo(ni, inf)

C++ HRESULT pNetArea-> UpdateInfo(long ni, INet\_Info \*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	网实体号
inf	Net_Info	[in]	网图形信息
rtl	Integer	[out]	返回值：1/0/-1=存在/不存在/被删除

### GetDatLen 方法

描述 取 ndat 长度。

Visual Basic datLen=NetArea. GetDatLen (ni)

C++ HRESULT pNetArea->GetDatLen(long ni, long \*datLen);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	网实体号
datLen	Long	[out]	返回 ndat 长度

### GetLength 方法

描述 取网长度。

Visual Basic rtl = GetLength(ni, length)

C++ HRESULT pNetArea->GetLength(  
long ni, double \*length, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	网号
length	Double	[in]	网长度
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SavePart 方法**

描述 保存部分网实体。

Visual Basic rtl = NetArea.SavePart(pad, inf, [layerOnOffPad])

C++ HRESULT pNetArea-> SavePart(  
INet\_Pad \*pad, INet\_Info \*inf,  
ILayerOnOffPad \*layerOnOffPad=NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Net_Pad	[in]	网实体参数开关
inf	Net_Info	[in]	网实体图形信息
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空，则只保存处于打开状态的图层。 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CopyPart 方法**

描述 拷贝部分网到 ai1 工作区。

Visual Basic netNum = NetArea.CopyPart(pad, inf, ai1, [layerOnOffPad])

C++ HRESULT pNetArea-> CopyPart (  
INet\_Pad \*pad, INet\_Info \*inf, INetArea \*ai1,  
ILayerOnOffPad \*layerOnOffPad=NULL, long \*netNum);

参数说明：

参数	类型	出口/入口	描述
pad	Net_Pad	[in]	网实体参数开关
inf	Net_Info	[in]	网实体图形信息
ai1	NetArea	[in]	目的工作区
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空，则只拷贝处于打开状态的图层。 可选；缺省值：Nothing
netNum	Long	[out]	成功返回拷贝的网数，失败返回 0

**CopyList 方法**

描述 拷贝列表中的网到工作区 ai1。

Visual Basic netNum = NetArea.CopyList(lst, ai1, [rtnNo])

C++ HRESULT pNetArea->CopyList(  
IIDList \*lst, INetArea \*ai1, short rtnNo, long \*netNum);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	要拷贝的网图元列表
ai1	NetArea	[in]	目的工作区
rtnNo	Integer	[in]	是否通过 lst 返回新的实体号
netnum	Long	[out]	成功返回拷贝的网数，失败返回 0

**CopyAll 方法**

描述 拷贝未删除的网,包括这些网所用弧段（网段）及其所连节点。

Visual Basic netNum = NetArea.CopyAll(ai1)

C++ HRESULT pNetArea-&gt;CopyAll (INetArea \*ai1, long \*netNum);

参数说明：

参数	类型	出口/入口	描述
ai1	NetArea	[in]	目的工作区
netNum	Long	[out]	成功返回拷贝的网数，失败返回 0

**CopyArea 方法**

描述 拷贝 ai 工作区中所有未删除的信息,包括独立弧段,独立节点。

Visual Basic rtl=NetArea.CopyArea(ai1)

C++ HRESULT pNetArea-&gt;CopyArea(INetArea \*ai1, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai1	NetArea	[in]	目的工作区
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Get 方法**

描述 取网。

Visual Basic rtl = NetArea.Get(ni, ndat, inf)

C++ HRESULT pNetArea->Get(  
long ni, ILONGList \*\*ndat, INet\_Info \*\*inf, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	网实体号
ndat	LONGList	[out]	网弧段号集合
inf	Net_Info	[out]	网图形信息
rtl	Integer	[out]	返回值：1/0/-1 = 成功/失败/被删除

**Append 方法**

**描述** 添加网。

**Visual Basic** netNo = NetArea.Append(ndat, inf)

**C++** HRESULT pNetArea-> Append(  
ILONGList \*ndat, INet\_Info \*inf, long \*netNo);

**参数说明：**

参数	类型	出口/入口	描述
ndat	LONGList	[in]	网弧段号集合
inf	Lin_Info	[in]	网图形信息
netNo	Long	[out]	成功返回网实体号 (>0), 失败返回 0

**Update 方法**

**描述** 更新第 ni 号网实体。

**Visual Basic** rtl = NetArea.Update(ni, ndat, inf)

**C++** HRESULT pNetArea-> Update(  
long ni, ILONGList \*ndat, INet\_Info \*inf, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
ni	Long	[in]	网实体号
ndat	LONGList	[in]	网段号集合
inf	Net_Info	[in]	网图形信息
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**Del 方法**

**描述** 从逻辑上删除第 ni 号网实体。

**Visual Basic** rtl = NetArea.Del(ni)

**C++** HRESULT pNetArea-> Del(long ni, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
ni	Long	[in]	删除的网实体号
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**UnDel 方法**

**描述** 恢复被删除的网。

**Visual Basic** rtl = NetArea.UnDel(ni)

**C++** HRESULT pNetArea->UnDel(long ni, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
ni	Long	[in]	恢复的网实体号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ClearTopDat 方法**

描述 清除拓扑数据。

Visual Basic rtl = NetArea.ClearTopDat()

C++ HRESULT pNetArea->ClearTopDat(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetArcAllocInfo 方法**

描述 取弧段分配信息。

Visual Basic rtl = NetArea.GetArcAllocInfo(li, allocInf)

C++ HRESULT pNetArea->GetArcAllocInfo(  
long li, IArc\_Alloc\_Info \*\*allocInf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	弧段号
allocInf	Arc_Alloc_Info	[out]	弧段分配信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SetArcAllocInfo 方法**

描述 设置弧段分配信息。

Visual Basic rtl = NetArea.SetArcAllocInfo(li, allocInf)

C++ HRESULT pNetArea->SetArcAllocInfo(  
long li, IArc\_Alloc\_Info \*allocInf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	弧段号
allocInf	Arc_Alloc_Info	[in]	弧段分配信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SetAllocatedArcs 方法**

描述 设置弧段分配方案。

Visual Basic rtl = NetArea.SetAllocatedArcs(arcLst)

C++ HRESULT pNetArea->SetAllocatedArcs (  
ILONGList \*arcLst, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
arcLst	LONGList	[in]	弧段分配列表
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetAllocatedArcs 方法

描述 设置弧段分配方案。

Visual Basic rtl = NetArea.GetAllocatedArcs(arcLst)

C++ HRESULT pNetArea->GetAllocatedArcs (  
ILONGList \*\*arcLst, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
arcLst	LONGList	[in]	弧段分配列表
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetAllocatedArcsNum 方法

描述 取所有弧段数。

Visual Basic arcNum = NetArea.GetAllocatedArcsNum()

C++ HRESULT pNetArea->GetAllocatedArcsNum (long \*arcNum);

参数说明：

参数	类型	出口/入口	描述
arcNum	Long	[out]	成功返回弧段数，失败返回 0

### GetAutoGetBinDatFlag 方法

描述 取自动取二进制数据字段标志。

Visual Basic flg = NetArea.GetAutoGetBinDatFlag()

C++ HRESULT pNetArea-> GetAutoGetBinDatFlag(short \*flg);

参数说明：

参数	类型	出口/入口	描述
flg	Integer	[out]	返回值：TRUE/FALSE=自动/不自动

### SetAutoGetBinDatFlag 方法

描述 设自动取二进制数据字段标志。

Visual Basic preFlg = NetArea.SetAutoGetBinDatFlag(flgs)

C++ HRESULT pNetArea-> SetAutoGetBinDatFlag(short flg, short \*preFlg);

参数说明：

参数	类型	出口/入口	描述
flg	Integer	[in]	自动取二进制数据字段标志
preFlg	Integer	[out]	返回值：TRUE/FALSE=自动/不自动

**ModParam 方法**

描述 统改网参数。

Visual Basic rtl = NetArea.ModParam(pad0, inf0, pad1, inf1, [layerOnOffPad])

C++ HRESULT pNetArea-> ModParam(  
 INet\_Pad \*pad0, INet\_Info \*inf0, INet\_Pad \*pad1, INet\_Info \*inf1,  
 ILayerOnOffPad \*layerOnOffPad=NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad0	Net_Pad	[in]	查找网实体参数开关
inf0	Net_Info	[in]	源网实体参数
pad1	Net_Pad	[in]	修改网实体参数开关
inf1	Net_Info	[in]	新的网实体参数
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空， 则只修改处于打开状态的图层。 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetMapParam 方法**

描述 取地图参数。

Visual Basic rtl = NetArea.GetMapParam(mpar)

C++ HRESULT pNetArea-> GetMapParam(  
 IMap\_Para \*\*mpar, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
mpar	Map_Para	[out]	地图参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SetMapParam 方法**

描述 设地图参数。

Visual Basic rtl = NetArea.SetMapParam(mpar)

C++ HRESULT pNetArea-> SetMapParam(  
 IMap\_Para \*mpar, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
mpar	Map_Para	[in]	地图参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### ModMapParam 方法

描述 修改地图参数。

Visual Basic rtl = NetArea.ModMapParam()

C++ HRESULT pNetArea-> ModMapParam(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### CopyMapParam 方法

描述 拷贝地图参数到工作区 ai1。

Visual Basic rtl = NetArea.CopyMapParam(ai1)

C++ HRESULT pNetArea-> CopyMapParam(  
IWorkArea \*ai1, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai1	WorkArea	[in]	目的工作区
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### Transform 方法

描述 图形变换。

Visual Basic rtl = NetArea.Transform(tp)

C++ HRESULT pNetArea-> Transform(ITran\_Par \*tp, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
tp	Tran_Par	[in]	图形变换参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### dbGetAreaDSType 方法

描述 取工作区数据源类型。

Visual Basic type = NetArea.dbGetAreaDSType()

C++ HRESULT pNetArea-> dbGetAreaDSType(Enum\_GisDS\_Type \*type);

参数说明：



参数	类型	出口/入口	描述
type	Enum_GisDS_Type	[out]	返回工作区数据源类型： PC_GIS/NET_GIS=单机数据源/网络数据源

**dbGetAreaDSName 方法**

描述 取工作区数据源名称。

Visual Basic rtl = NetArea.dbGetAreaDSName(szDSN)

C++ HRESULT pNetArea->dbGetAreaDSName(BSTR \*szDSN, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
szDSN	String	[out]	工作区数据源名称
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbLockAll 方法**

描述 全锁工作区中的数据。

Visual Basic rtl = NetArea.dbLockAll()

C++ HRESULT pNetArea->dbLockAll(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbUnLockAll 方法**

描述 解除全锁。

Visual Basic rtl = NetArea.dbUnLockAll()

C++ HRESULT pNetArea->dbUnLockAll(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbLockRect 方法**

描述 锁工作区中矩形内的数据。

Visual Basic rtl = NetArea.dbLockRect(rc)

C++ HRESULT pNetArea->dbLockRect(ID\_Rect \*rc, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rc	D_Rect	[in]	被封锁的数据范围

rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

**dbUnLockRect 方法**

描述 解除矩形锁。

Visual Basic rtl = NetArea.dbUnLockRect()

C++ HRESULT pNetArea-> dbUnLockRect(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbGetLockRange 方法**

描述 取被自己封锁的数据范围。

Visual Basic rtl = NetArea.dbGetLockRange(rc)

C++ HRESULT pNetArea-> dbGetLockRange(ID\_Rect \*\*rc, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
rc	D_Rect	[out]	被自己封锁的数据范围
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbGetLockNumByOther 方法**

描述 取被其他用户封锁的矩形块数。

Visual Basic num = NetArea.dbGetLockNumByOther()

C++ HRESULT pNetArea-> dbGetLockNumByOther(long \*num);

参数说明：

参数	类型	出口/入口	描述
num	Long	[out]	返回值：-1-失败，0-其他用户未封锁，>0-其他用户封锁的矩形块数

**dbGetLockRangeByOther 方法**

描述 取被其他用户封锁的矩形范围。

Visual Basic num = NetArea.dbGetLockRangeByOther(rc)

C++ HRESULT pNetArea-> dbGetLockRangeByOther(  
ID\_RectSet \*\*rc, long \*num);

参数说明：

参数	类型	出口/入口	描述
rc	D_RectSet	[out]	被其他用户封锁的矩形范围

num	Long	[out]	返回值：-1-失败，0-其他用户未封锁，>0-其他用户封锁的矩形块数
-----	------	-------	------------------------------------

#### dbLockSingle 方法

描述 封锁单个图元。

Visual Basic rtl = NetArea.dbLockSingle (ni)

C++ HRESULT pNetArea-> dbLockSingle(long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	封锁的图元号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dbUnLockSingle 方法

描述 取消单个图元的封锁。

Visual Basic rtl = NetArea.dbUnLockSingle(ni)

C++ HRESULT pNetArea-> dbUnLockSingle(long ni, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	取消封锁的图元号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dbBeginTransaction 方法

描述 事务开始。

Visual Basic rtl = NetArea.dbBeginTransaction([newTrans])

C++ HRESULT pNetArea-> dbBeginTransaction(short newTrans=0, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
newTrans	Integer	[in]	newTrans=0 时只增加嵌套次数，=1 时则强制提交以前开始的事务，并开始一个新的事务。 可选；缺省值：0
rtl	Integer	[out]	成功返回事务嵌套次数，失败返回 0

#### dbRollbackTransaction 方法

描述 事务回滚，忽略事务嵌套次数，回滚到最外层事务开始的地方，并将嵌套次数设 0。

Visual Basic rtl = NetArea.dbRollbackTransaction()

C++ HRESULT pNetArea-> dbRollbackTransaction(short \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Integer	[out]	成功返回 TRUE，失败返回 FALSE

#### dbCommitTransaction 方法

描述 事务提交。

Visual Basic rtl = NetArea.dbCommitTransaction([commitNow])

C++ HRESULT pNetArea-> dbCommitTransaction(  
short commitNow=0, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
commitNow	Integer	[in]	commitNow!=0 则马上提交并将嵌套次数置 0，否则只有嵌套次数减到 0 时才提交。 可选；缺省值：0
rtl	Integer	[out]	成功返回 TRUE，失败返回 FALSE

#### dbCreateFieldIndex 方法

描述 创建属性字段索引（只有拥有者才能创建索引）

Visual Basic rtl = NetArea.dbCreateFieldIndex(stru,  
ndxFldFlag, IsUnique, IsClustered, IsPadIndex)

C++ HRESULT pNetArea->dbCreateFieldIndex(  
IRecord\_Head \*stru, IRepFldFlag \*ndxFldFlag, short IsUnique,  
short IsClustered, short IsPadIndex, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
stru	Record_Head	[in]	属性结构( 主要用其中的字段名称 )
ndxFldFlag	RepFldFlag	[in]	创建字段索引标识（用来对应于创建索引的字段，ndxFldFlag[i]=1：表示字段 i 创建索引；=0：表示字段 i 不创建索引）
IsUnique	Integer	[in]	创建唯一索引标识：1/0-是/否
IsClustered	Integer	[in]	创建聚簇索引标识：1/0-是/否
IsPadIndex	Integer	[in]	1/0-是/否
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### PartToList 方法

描述 拷贝部分网到列表当中。

Visual Basic Set lst = NetArea.PartToList(pad, inf, [layerOnOffPad])

C++ HRESULT pNetArea->PartToList(INet\_Pad \*pad, INet\_Info \*inf,  
ILayerOnOffPad \*layerOnOffPad=NULL, IIDList \*\*lst);

参数说明：

参数	类型	出口/入口	描述
pad	Net_Pad	[in]	网实体参数开关
inf	Net_Info	[in]	网实体图形信息
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志，若此实体不为空， 则只拷贝处于打开状态的图层。 可选；缺省值：Nothing
lst	IDList	[out]	目标列表

### AppendFile 方法

描述 添加一个文件到工作区中。

Visual Basic rtl = NetArea.AppendFile([fName], [szDSN], [szLOG], [szPSW])

C++ HRESULT pNetArea->AppendFile(BSTR fname= NULL,BSTR szDSN= NULL,  
BSTR szLOG= NULL, BSTR szPSW= NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fName	String	[in]	文件名字符串。 可选；如果缺省则弹出对话框选择文件。
szDSN	String	[in]	ODBC 数据源名称。 可选；缺省值：NULL
szLOG	String	[in]	登录号（登录名）。 可选；缺省值：NULL
szPSW	String	[out]	登录口令。 可选；缺省值：NULL
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetFileName 方法

描述 取工作区对应的文件名（不带路径）。

Visual Basic rtl = NetArea.GetFileName(fName)

C++ HRESULT pNetArea->GetFileName(BSTR \*fname, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fName	String	[out]	工作区对应的文件名
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### 3.100、TblArea 对象

描述 TblArea 对象封装了对表工作区的操作。

#### 3.100.1、属性：

属性	描述
nCount	表工作区中的记录数
Long	
只读	
stru	表属性结构
Record_Head	
可读写	
Type	工作区类型
Enum_Area_Type	
只读	
Name	工作区名称
String	
可读写	
Changed	工作区已经改变标志
Boolean	
可读写	
ShowErrMsg	显示错误信息标志
Boolean	
可读写	
Warning	对不能后悔的操作是否给出警告提示
Boolean	
可读写	
Compact	保存工作区时压缩数据标志
Boolean	
可读写	

Empty	工作区是否为空标志
Boolean	
只读	
LastErrCode	最后错误码
Long	
只读	

### 3.100.2、方法：

#### GetRcdExistFlag 方法

描述 取表格记录存在标志。

Visual Basic rtl = TblArea.GetRcdExistFlag(rcdi)

C++ HRESULT pTblArea->GetRcdExistFlag(long rcdi, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rcdi	Long	[in]	记录号
rtl	Boolean	[out]	存在返回 TRUE，失败返回 FALSE

#### GetAllExistNo 方法

描述 取所有没有被删除的记录号。

Visual Basic nCount = TblArea.GetAllExistNo(lst)

C++ HRESULT pTblArea->GetAllExistNo(IDList \*\*lst, long \*nCount);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[out]	记录号列表
nCount	Long	[out]	返回记录数

#### SetMMDPath 方法

描述 设置多媒体数据路径。

Visual Basic rtl = TblArea.SetMMDPath(mmdPath)

C++ HRESULT pTblArea->SetMMDPath(  
BSTR mmdPath, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
mmdPath	String	[in]	多媒体数据路径
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetMMDPath 方法**

描述 读取多媒体数据路径。

Visual Basic rtl = TblArea.GetMMDPath(mmdPath)

C++ HRESULT pTblArea->GetMMDPath(  
BSTR \*mmdPath, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
mmdPath	String	[out]	多媒体数据路径
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ModMMDPath 方法**

描述 修改多媒体数据路径。

Visual Basic rtl = TblArea.ModMMDPath()

C++ HRESULT pTblArea->ModMMDPath(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Get 方法**

描述 取第 i 条属性记录。

Visual Basic rtl = TblArea.Get(i, att)

C++ HRESULT pTblArea->Get(  
long i, IRecord \*\*att, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
i	Long	[in]	实体号
att	Record	[out]	属性记录值
rtl	Integer	[out]	返回值：1=成功、2=空属性记录、0=失败、-1=该属性对应的记录已经被删除

**GetDef 方法**

描述 取缺省属性记录。

Visual Basic rtl = TblArea.GetDef(IRecord \*\*att)

C++ HRESULT pTblArea->GetDef(IRecord \*\*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
att	Record	[out]	属性记录值



rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE
-----	---------	-------	------------------------

**Write 方法**

描述 写属性。

Visual Basic rtl = TblArea.Write(i, att)

C++ HRESULT pTblArea->Write(  
long i, IRecord \*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
i	Long	[in]	实体号
att	Record	[in]	属性记录值
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

**WriteDef 方法**

描述 设置缺省属性记录值。

Visual Basic rtl = TblArea. WriteDef(att)

C++ HRESULT pTblArea->WriteDef(IRecord \*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
att	Record	[in]	缺省属性记录值
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

**GetID 方法**

描述 取属性 ID。

Visual Basic pid = TblArea.GetID(i)

C++ HRESULT pTblArea->GetID(long i, long \*pid);

参数说明：

参数	类型	出口/入口	描述
i	Long	[in]	实体号
pid	Long	[out]	返回属性 ID

**GetMaxID 方法**

描述 取最大 ID 值。

Visual Basic max\_id\_val = TblArea.GetMaxID()

C++ HRESULT pTblArea->GetMaxID(long \*max\_id\_val);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

max_id_val	Long	[out]	返回最大属性 ID
------------	------	-------	-----------

**CopyAll 方法**

描述 拷贝所有属性记录到表工作区 ai1 中。

Visual Basic attnum = TblArea.CopyAll(ai1)

C++ HRESULT pTblArea->CopyAll(ITblArea \*ai1, long \*attnum);

参数说明：

参数	类型	出口/入口	描述
ai1	TblArea	[in]	目的工作区
attnum	Long	[out]	拷贝的记录数

**CopyList 方法**

描述 拷贝 lst 指定的记录到表工作区 ai1 中。

Visual Basic attnum = TblArea.CopyList(lst, ai1, [changeStru], [rtnNo])

C++ HRESULT pTblArea->CopyList(  
IIDList \*lst, ITblArea \*ai1, short changeStru=1, short rtnNo=0, long \*attnum);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	属性记录列表
ai1	TblArea	[in]	目的工作区
changeStru	Integer	[in]	是否改变 ai1 的属性结构 ,使其和 TblArea 的结构相同。 可选；缺省值：1
rtnNo	Integer	[in]	是否通过 lst 返回新的记录号。 可选；缺省值：0
attnum	Long	[out]	拷贝的记录数

**AppendRcd 方法**

描述 添加一条表记录。

Visual Basic rcdNo = TblArea.AppendRcd(att)

C++ HRESULT pTblArea->AppendRcd(IRecord \*att, long \*rcdNo);

参数说明：

参数	类型	出口/入口	描述
att	Record	[in]	记录值
rcdNo	Long	[out]	成功返回记录号，失败返回 0

**DelRcd 方法**

描述 删除第 rcdi 条记录。

Visual Basic rtl = TblArea.DelRcd(rcdi)

C++ HRESULT pTblArea-> DelRcd(long rcdi, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rcdi	Long	[in]	记录号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UnDelRcd 方法**

描述 撤消对第 rcdi 条记录的删除。

Visual Basic rtl = TblArea.UnDelRcd(rcdi)

C++ HRESULT pTblArea->UnDelRcd(long rcdi, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rcdi	Long	[in]	记录号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**RepLstRcdFld0 方法**

描述 统改属性记录字段值。

Visual Basic nCount = TblArea.RepLstRcdFld0(lst, repFldFlg, attExp, pad, inf, repAtt)

C++ HRESULT pTblArea->RepLstRcdFld0(  
IIDList \*lst, IRepFldFlag \*pRepFldFlg, BSTR attExp,  
IPnt\_Pad \*pad, IPnt\_Info \*inf, IRecord \*repAtt, long \*nCount);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	将进行替换的记录号列表对象，若 lst 为空则统改所有记录
repFldFlg	RepFldFlag	[in]	替换字段开关；repFldFlag[i]对应 i（从 0 开始）号字段，1=替换，0=不替换
attExp	String	[in]	属性条件表达式字符串； 可为空,即无属性条件
pad	Pnt_Pad	[in]	图形参数开关,可为空
inf	Pnt_Info	[in]	图形参数,可为空
repAtt	Record	[in]	替换值记录数据
nCount	Long	[out]	成功返回统改的记录数，失败返回 0

### RepLstRcdFld 方法

描述 字段统改。

Visual Basic rtl = TblArea.RepLstRcdFld(lst, ttlName)

C++ HRESULT pTblArea->RepLstRcdFld(  
IIDList \*lst, BSTR ttlName, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	统改记录号表，若为空则统改所有记录
ttlName	String	[in]	统改属性值条件窗口标题
rtl	Integer	[out]	成功返回 1，失败返回 0

### ModLstRcdFldVal 方法

描述 统改字段域值。

Visual Basic rtl = TblArea.ModLstRcdFldVal(lst, fldI, fldVal)

C++ HRESULT pTblArea->ModLstRcdFldVal(  
IIDList \*lst, short fldI, VARIANT fldVal, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	统改记录号表，若为空则统改所有记录
fldI	Integer	[in]	统改字段号
fldVal	Variant	[in]	字段域值
rtl	Integer	[out]	成功返回 1，失败返回 0

### CalculateFldVal 方法

描述 计算字段值。

Visual Basic rtl = TblArea.CalculateFldVal(fldName, calExp, domainErrVal)

C++ HRESULT pTblArea->CalculateFldVal(  
BSTR fldName,BSTR calExp,  
VARIANT domainErrVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fldName	String	[in]	计算字段名称
calExp	String	[in]	计算表达式

domainErrVal	Variant	[in]	表达式计算过程中定义域错时 ( 如除 0 错 ), 计算结果替换值,若 domainErrVal 不为空,则用 domainErrVal 作为 fldName 的字段值;若为空,则不修改当前记录的 fldName 字段值
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**dbGetAreaDSType 方法**

描述 取工作区数据源类型。

Visual Basic type = TblArea.dbGetAreaDSType()

C++ HRESULT pTblArea-> dbGetAreaDSType(Enum\_GisDS\_Type \*type);

参数说明：

参数	类型	出口/入口	描述
type	Enum_GisDS_Type	[out]	返回工作区数据源类型： PC_GIS/NET_GIS=单机数据源/网络数据源

**dbGetAreaDSName 方法**

描述 取工作区数据源名称。

Visual Basic rtl = TblArea.dbGetAreaDSName(szDSN)

C++ HRESULT pTblArea->dbGetAreaDSName(BSTR \*szDSN, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
szDSN	String	[out]	工作区数据源名称
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**dbLockAll 方法**

描述 全锁工作区中的数据。

Visual Basic rtl=TblArea. dbLockAll()

C++ HRESULT pTblArea-> dbLockAll(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**dbUnLockAll 方法**

描述 解除全锁。

Visual Basic rtl = TblArea.dbUnLockAll()

C++ HRESULT pTblArea-> dbUnLockAll(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dbGetLockNumByOther 方法

描述 取被其他用户封锁的矩形块数。

Visual Basic num = TblArea.dbGetLockNumByOther()

C++ HRESULT pTblArea-> dbGetLockNumByOther(long \*num);

参数说明：

参数	类型	出口/入口	描述
num	Long	[out]	返回值：-1-失败，0-其他用户未封锁，>0-其他用户封锁的矩形块数

#### dbLockSingle 方法

描述 封锁单条记录。

Visual Basic rtl = TblArea.dbLockSingle(pi)

C++ HRESULT pTblArea-> dbLockSingle(long pi, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	封锁的记录号
rtl	Boolean	[out]	返回值：1/0 = 成功/失败

#### dbUnLockSingle 方法

描述 取消单条记录的封锁。

Visual Basic rtl = TblArea.dbUnLockSingle(pi)

C++ HRESULT pTblArea-> dbUnLockSingle(long pi, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pi	Long	[in]	取消封锁的记录号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dbBeginTransaction 方法

描述 事务开始。

Visual Basic rtl = TblArea.dbBeginTransaction([newTrans])

C++ HRESULT pTblArea-> dbBeginTransaction(short newTrans=0, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
newTrans	Integer	[in]	newTrans=0 时只增加嵌套次数，=1 时则强制提交以前开始的事务，并开始一个新的事务。 可选；缺省值：0
rtl	Integer	[out]	成功返回事务嵌套次数，失败返回 0

**dbRollbackTransaction 方法**

**描述** 事务回滚，忽略事务嵌套次数，回滚到最外层事务开始的地方，并将嵌套次数设 0。

Visual Basic rtl = TblArea.dbRollbackTransaction()

C++ HRESULT pTblArea-> dbRollbackTransaction(short \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Integer	[out]	成功返回 TRUE，失败返回 FALSE

**dbCommitTransaction 方法**

**描述** 事务提交。

Visual Basic rtl = TblArea.dbCommitTransaction([commitNow])

C++ HRESULT pTblArea-> dbCommitTransaction(  
short commitNow=0, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
commitNow	Integer	[in]	commitNow!=0 则马上提交并将嵌套次数置 0，否则只有嵌套次数减到 0 时才提交。 可选；缺省值：0
rtl	Integer	[out]	成功返回 TRUE，失败返回 FALSE

**dbCreateFieldIndex 方法**

**描述** 创建属性字段索引（只有拥有者才能创建索引）。

Visual Basic rtl = TblArea.dbCreateFieldIndex(stru,  
ndxFldFlag, IsUnique, IsClustered, IsPadIndex)

C++ HRESULT pTblArea->dbCreateFieldIndex(  
IRecord\_Head \*stru, IRepFldFlag \*ndxFldFlag, short IsUnique,  
short IsClustered, short IsPadIndex, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
stru	Record_Head	[in]	属性结构( 主要用其中的字段名称 )
ndxFldFlag	RepFldFlag	[in]	创建字段索引标识 ( 用来对应于创建索引的字段 , ndxFldFlag[i]=1 : 表示字段 i 创建索引 ; =0 : 表示字段 i 不创建索引 )
IsUnique	Integer	[in]	创建唯一索引标识 : 1/0-是/否
IsClustered	Integer	[in]	创建聚簇索引标识 : 1/0-是/否
IsPadIndex	Integer	[in]	1/0-是/否
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

**GetAutoGetBinDatFlag 方法**

描述 取自动取二进制数据字段标志。

Visual Basic flg = TblArea.GetAutoGetBinDatFlag()

C++ HRESULT pTblArea-> GetAutoGetBinDatFlag(short \*flg);

参数说明 :

参数	类型	出口/入口	描述
flg	Integer	[out]	返回值 : TRUE/FALSE=自动/不自动

**SetAutoGetBinDatFlag 方法**

描述 设自动取二进制数据字段标志。

Visual Basic preFlg = TblArea.SetAutoGetBinDatFlag(flgl)

C++ HRESULT pTblArea-> SetAutoGetBinDatFlag(short flgl, short \*preFlgl);

参数说明 :

参数	类型	出口/入口	描述
flgl	Integer	[in]	自动取二进制数据字段标志
preFlgl	Integer	[out]	返回值 : TRUE/FALSE=自动/不自动

**AppendFile 方法**

描述 添加一个文件到工作区中。

Visual Basic rtl = TblArea.AppendFile([fName], [szDSN], [szLOG], [szPSW])

C++ HRESULT pTblArea->AppendFile(BSTR fName= NULL,BSTR szDSN= NULL,  
BSTR szLOG= NULL, BSTR szPSW= NULL, VARIANT\_BOOL \*rtl);

参数说明 :

参数	类型	出口/入口	描述
----	----	-------	----



fName	String	[in]	文件名字符串。 可选；如果缺省则弹出对话框选择文件。
szDSN	String	[in]	ODBC 数据源名称。 可选；缺省值：NULL
szLOG	String	[in]	登录号（登录名）。 可选；缺省值：NULL
szPSW	String	[out]	登录口令。 可选；缺省值：NULL
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetFileName 方法**

描述 取工作区对应的文件名（不带路径）。

Visual Basic rtl = TblArea.GetFileName(fName)

C++ HRESULT pTblArea->GetFileName(BSTR \*fname, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fName	String	[out]	工作区对应的文件名
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**3.101、ExpObj 对象**

描述 ExpObj 对象是表达式对象。

**3.101.1、属性：**

属性	描述
myObj	表达式对象
Long 只读	

**3.101.2、方法：无****3.102、Query 对象**

描述 Query 对象是查询对象。

### 3.102.1、属性：

属性	描述
sourceArea	源工作区
WorkArea	
可读写	

### 3.102.2、方法：

#### RectAskToList 方法

描述 矩形范围查询,查询结果存入实体号数组，并返回数组地址。

Visual Basic Set ptLst = Query.RectAskToList(atype, frc, [layerOnOffPad])

C++ HRESULT pQuery->RectAskToList(Enum\_Entity\_Type atype, ID\_Rect \*frc, ILayerOnOffPad \*layerOnOffPad=NULL, IIDList \*\*ptLst);

参数说明：

参数	类型	出口/入口	描述
atype	Enum_Entity_Type	[in]	查询类型
frc	D_Rect	[in]	查询范围
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志。 可选；缺省值：Nothing
ptLst	IIDList	[out]	成功返回实体号数组地址， 否则返回 NULL

#### RectAskToList0 方法

描述 矩形范围查询图元，返回图元表。cmpLenArea 和 cmpFlag 两个参数，仅对线或区进行检索时才有效。

Visual Basic Set ptLst = Query.RectAskToList0(atype, frc, [layerOnOffPad], [cmpRectOnly], [mustInside], [cmpLenArea], [cmpFlag])

C++ HRESULT pQuery->RectAskToList0(Enum\_Entity\_Type atype, ID\_Rect \*frc, ILayerOnOffPad \*layerOnOffPad=NULL, short cmpRectOnly=1, short mustInside=0, double cmpLenArea=0, short cmpFlag = 5 , IIDList \*\*ptLst);

参数说明：

参数	类型	出口/入口	描述
atype	Enum_Entity_Type	[in]	查询类型
frc	D_Rect	[in]	查询范围
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志。 可选；缺省值：Nothing

cmpRectOnly	Integer	[in]	是否只比较图元的矩型范围标志 =1 则只要图元的外包络矩形范围与 rc 矩形范围相交，该图元即被选中。 可选；缺省值：1
mustInside	Integer	[in]	是否严格要求图元在矩型范围内的标志， =1 则必须要判断图元是否与 rc 矩形范围相交，若相交则该图元即被选中。 可选；缺省值：0
cmpLenArea	Double	[in]	比较长度或面积
cmpFlag	Integer	[in]	比较类型，其值为：1/2/3/4/5/6= 小于/小于等于/等于/不等/大于等于/大于。缺省值为 5。
ptLst	IDList	[out]	成功返回实体号数组地址， 否则返回 NULL

#### RectAskToArea 方法

描述 矩形范围查询，查询结果存入实体号数组，并返回数组地址。

Visual Basic entityNum = Query.RectAskToArea(atype, frc, ai1, [layerOnOffPad])

C++ HRESULT pQuery->RectAskToArea(  
Enum\_Entity\_Type atype, ID\_Rect \*frc, IWorkArea \*ai1,  
ILayerOnOffPad \*layerOnOffPad=NULL, long \*entityNum);

参数说明：

参数	类型	出口/入口	描述
atype	Enum_Entity_Type	[in]	查询类型
frc	D_Rect	[in]	查询范围
ai1	WorkArea	[in]	目的工作区对象
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志
entityNum	Long	[out]	成功返回实体号数组地址， 否则返回 NULL

#### LayerAskToList 方法

描述 图层查询图元，查询结果存入实体号数组，并返回数组地址。

Visual Basic Set ptLst = Query.LayerAskToList(atype, layer)

C++ HRESULT pQuery->LayerAskToList (

```
Enum_Entity_Type atype, short layer, IIDList **ptLst);
```

参数说明：

参数	类型	出口/入口	描述
atype	Enum_Entity_Type	[in]	查询类型
layer	Integer	[in]	图层号
ptLst	IIDList	[out]	成功返回实体号数组地址， 否则返回 NULL

### LayerAskToArea 方法

描述 图层查询图元，查询结果存入目的工作区。

Visual Basic entityNum = Query.LayerAskToArea(atype, layer, ai1)

C++ HRESULT pQuery->LayerAskToArea (  
Enum\_Entity\_Type atype, short layer, IWorkArea \*ai1, long \*entityNum);

参数说明：

参数	类型	出口/入口	描述
atype	Enum_Entity_Type	[in]	查询类型
layer	Integer	[in]	图层号
ai1	WorkArea	[in]	目的工作区对象
entityNum	Long	[out]	成功返回实体号数组地址， 否则返回 NULL

### AttAskToArea 方法

描述 属性检索，查询结果存入目的工作区。

Visual Basic entityNum = Query.AttAskToArea(atype, ai1)

C++ HRESULT pQuery->AttAskToArea (  
Enum\_Entity\_Type atype, IWorkArea \*ai1, long \*entityNum);

参数说明：

参数	类型	出口/入口	描述
atype	Enum_Entity_Type	[in]	查询类型
ai1	WorkArea	[in]	目的工作区对象
entityNum	Long	[out]	成功返回实体号数组地址， 否则返回 NULL

### AttAskToArea0 方法

描述 属性检索，查询结果存入目的工作区。

Visual Basic linNum = Query.AttAskToArea0(atype, ai1,exp)

C++ HRESULT pQuery->AttAskToArea0 (

---

```
Enum_Entity_Type atype, BSTR exp, IWorkArea *ai1, long * linNum);
```

参数说明：

参数	类型	出口/入口	描述
atype	Enum_Entity_Type	[in]	查询类型
exp	String	[in]	条件表达式字符串
ai1	WorkArea	[in]	目的工作区对象
linNum	Long	[out]	成功返回查询到的实体数 (>0), 否则返回 0

---

**AttAskToList0 方法**

描述 图层查询图元，查询结果存入实体号数组，并返回数组地址。

Visual Basic Set ptLst = Query.AttAskToList0(atype, exp)

```
C++ HRESULT pQuery-> AttAskToList0(
    Enum_Entity_Type atype, BSTR exp, IIDList **ptLst);
```

参数说明：

参数	类型	出口/入口	描述
atype	Enum_Entity_Type	[in]	查询类型
exp	String	[in]	条件表达式字符串
ptLst	IDList	[out]	成功返回实体号数组地址，否则返回 NULL

---

**AttAskToListOnDbNo 方法**

描述 根据外挂数据库号查找属性。

Visual Basic Set ptLst = Query.AttAskToListOnDbNo(atype, keyFldName, dbNo)

```
C++ HRESULT pQuery-> AttAskToListOnDbNo (
    Enum_Entity_Type atype, BSTR keyFldName, short dbNo, IIDList **ptLst);
```

参数说明：

参数	类型	出口/入口	描述
atype	Enum_Entity_Type	[in]	查询类型
keyFldName	String	[in]	关键字段名
dbNo	Integer	[in]	外挂数据库号
ptLst	IDList	[out]	返回属性地址

---

**AttAskToListOnDbExp 方法**

描述 根据外挂数据库及条件表达式查找对应属性。

Visual Basic Set ptLst = Query.AttAskToListOnDbExp(atype, keyFldName, dbNo, exp)

```
C++ HRESULT pQuery-> AttAskToListOnDbExp (Enum_Entity_Type atype,
```

---

BSTR keyFldName, short dbNo, BSTR exp IIDList \*\*ptLst);

参数说明：

参数	类型	出口/入口	描述
atype	Enum_Entity_Type	[in]	查询类型
keyFldName	String	[in]	关键字段名
dbNo	Integer	[in]	外挂数据库号
exp	String	[in]	表达式字符串
ptLst	IDList	[out]	返回属性地址

---

**AttCrossAskToArea 方法**

描述 交叉属性条件查询，查询结果存入目的工作区。

Visual Basic num = Query.AttCrossAskToArea(exp, linkfld, ai1, ai2)

C++ HRESULT pQuery-> AttCrossAskToArea(  
BSTR exp, BSTR linkfld, IWorkArea \*ai1, IWorkArea \*ai2, long \*num);

参数说明：

参数	类型	出口/入口	描述
exp	String	[in]	表达式字符串
linkfld	String	[in]	连接字段字符串
ai1	WorkArea	[in]	查询工作区对象
ai2	WorkArea	[in]	目的工作区对象
num	Long	[out]	成功返回实体数 (>0), 否则返回 0

---

**AttCrossAskToList 方法**

描述 交叉属性条件查询。

Visual Basic Set ptLst = Query.AttCrossAskToList(exp, linkfld, ai1)

C++ HRESULT pQuery-> AttCrossAskToList (  
BSTR exp, BSTR linkfld, IWorkArea \*ai1, IIDList \*\*ptLst);

参数说明：

参数	类型	出口/入口	描述
exp	String	[in]	表达式字符串
linkfld	String	[in]	连接字段字符串
ai1	WorkArea	[in]	查询工作区对象
ptLst	IDList	[out]	成功返回实体号数组地址， 否则返回 NULL

---

**RegAskToArea 方法**

描述 区域查询，查询结果存入目的工作区。

Visual Basic entityNum = Query.RegAskToArea(ai0, atype, ai2, [outFlg])

C++ HRESULT pQuery-> RegAskToArea(  
IRegArea \*ai0, Enum\_Entity\_Type atype, IWorkArea \*ai2,  
short outFlg=0, long \*entityNum);

参数说明：

参数	类型	出口/入口	描述
ai0	RegArea	[in]	区工作区对象
atype	Enum_Entity_Type	[in]	查询类型
ai2	WorkArea	[in]	目的工作区对象
outFlg	Integer	[in]	查询标志，0/1 查询所有与区域相交的图元/查询区域外的图元。 可选；缺省值：0
entityNum	Long	[out]	成功返回实体号数组地址， 否则返回 NULL

**RegAskToList 方法**

描述 区域查询，查询结果存入实体号数组，并返回数组地址。

Visual Basic Set ptLst = Query.RegAskToList(ai0, atype, [outFlg])

C++ HRESULT pQuery-> RegAskToList(  
IRegArea \*ai0, Enum\_Entity\_Type atype, short outFlg=0, IIDList \*\*ptLst);

参数说明：

参数	类型	出口/入口	描述
ai0	RegArea	[in]	区工作区对象
atype	Enum_Entity_Type	[in]	查询类型
outFlg	Integer	[in]	查询标志，0/1-查询所有与区域相交的图元/查询区域外的图元。 可选；缺省值：0
ptLst	IDList	[out]	成功返回实体号数组地址， 否则返回 NULL

**NearNod 方法**

描述 查找最近结点。

Visual Basic nodNo = Query.NearNod(xy, rtnDis)

C++ HRESULT pQuery-> NearNod(ID\_Dot \*xy, double \*rtnDis, long \*nodNo);

参数说明：

参数	类型	出口/入口	描述
xy	D_Dot	[in]	点对象，xy 为图型实际坐标
rtnDis	Double	[in]	距离
nodNo	Long	[out]	成功返回点号 (>0)，失败返回 0

**NearPnt 方法**

描述 查找最近点。

Visual Basic pntNo = Query.NearPnt(xy, [layerOnOffPad], [dis]);

C++ HRESULT pQuery-> NearPnt(  
ID\_Dot \*xy, ILayerOnOffPad \*layerOnOffPad=NULL, double \*dis=NULL, long  
\*pntNo);

参数说明：

参数	类型	出口/入口	描述
xy	D_Dot	[in]	点对象，xy 为图型实际坐标
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志。 可选；缺省值：Nothing
dis	Double	[out]	距离。 可选；缺省值：Nothing
pntNo	Long	[out]	成功返回点号 (>0)，失败返回 0

**NearLin 方法**

描述 查找最近线。

Visual Basic linNo = Query.NearLin(xy, [layerOnOffPad], [dis]);

C++ HRESULT pQuery-> NearLin (  
ID\_Dot \*xy, ILayerOnOffPad \*layerOnOffPad=NULL, double \*dis=NULL, long \*  
linNo);

参数说明：

参数	类型	出口/入口	描述
xy	D_Dot	[in]	点对象，xy 为图型实际坐标
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志。 可选；缺省值：Nothing
dis	Double	[out]	距离。 可选；缺省值：Nothing
linNo	Long	[out]	成功返回线号 (>0)，失败返回 0

**SeekReg 方法**

描述 查找最近区。



Visual Basic regNo = Query.SeekReg(xy, [layerOnOffPad], [dis]);  
 C++ HRESULT pQuery-> SeekReg (  
 ID\_Dot \*xy, ILayerOnOffPad \*layerOnOffPad=NULL, double \*dis=NULL, long \*  
 regNo);

参数说明：

参数	类型	出口/入口	描述
xy	D_Dot	[in]	点对象，xy 为图型实际坐标
layerOnOffPad	LayerOnOffPad	[in]	图层开关标志。 可选；缺省值：Nothing
dis	Double	[out]	距离。 可选；缺省值：Nothing
regNo	Long	[out]	成功返回区号 (>0)，失败返回 0

#### LinInterReg 方法

描述 判断线是否与区域相交。

Visual Basic rtl = Query.LinInterReg(li, ai, ri)

C++ HRESULT pQuery-> LinInterReg(long li, IRegArea \*ai, long ri, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线号
ai	RegArea	[in]	区工作区对象
ri	Long	[in]	区域号
rtl	Integer	[out]	相交返回 1，否则返回 0

#### RegInterReg 方法

描述 判断两个区域是否相交。

Visual Basic rtl = Query.RegInterReg(ri0, ai, ri1)

C++ HRESULT pQuery-> RegInterReg(long ri0, IRegArea \*ai, long ri1, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri0	Long	[in]	区域号
ai	RegArea	[in]	区工作区对象
ri1	Long	[in]	区域号
rtl	Integer	[out]	相交返回 1，否则返回 0

#### SelDotOfLin 方法

描述 查找线上最近的点。

Visual Basic dotNo = Query.SelDotOfLin(xy, dot)  
C++ HRESULT pQuery-> SelDotOfLin(  
ID\_DotSet \*xy, ID\_Dot \*dot, long \*dotNo);

参数说明：

参数	类型	出口/入口	描述
xy	D_DotSe	[in]	线坐标点集对象
dot	D_Dot	[in]	查找位置点对象
dotNo	Long	[out]	成功返回点序号 (>=0), 失败返回-1

### LinInterReg1 方法

描述 判断线是否与区域相交。

Visual Basic rtl = Query.LinInterReg1(xy, rxy, ne)  
C++ HRESULT pQuery-> LinInterReg1(  
ID\_DotSet \*xy, ID\_DotSet \*rxy, ILONGList \*ne, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
xy	D_DotSet	[in]	线坐标点集对象
rxy	D_DotSet	[in]	区域边界坐标点集对象
ne	LONGList	[in]	区域边界每一圈点数
rtl	Integer	[out]	相交返回 1, 否则返回 0

### RegInterReg1 方法

描述 判断两个区域是否相交。

Visual Basic rtl = Query.RegInterReg1(rxy0, ne0, rxy1, ne1)  
C++ HRESULT pQuery-> RegInterReg1 (  
ID\_DotSet \*rxy0, ILONGList \*ne0, ID\_DotSet \*rxy1, ILONGList \*ne1, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
rxy0	D_DotSet	[in]	区域边界坐标点集对象
ne0	LONGList	[in]	区域边界每一圈点数
rxy1	D_DotSet	[in]	区域边界坐标点集对象
ne1	LONGList	[in]	区域边界每一圈点数
rtl	Integer	[out]	相交返回 1, 否则返回 0

### NetInterReg 方法

描述 判断网络是否与区域相交。

Visual Basic rtl = Query.NetInterReg(ni, ai, ri)

C++ HRESULT pQuery-> NetInterReg(long ni, IRegArea \*ai, long ri, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	网络号
ai	RegArea	[in]	区工作区对象
ri	Long	[in]	区域号
rtl	Integer	[out]	相交返回 1，否则返回 0

### NetInterReg1 方法

描述 判断网络是否与区域相交。

Visual Basic rtl = Query.NetInterReg1(ni, rxy1, ne1)

C++ HRESULT pQuery-> NetInterReg1(  
long ni, ID\_DotSet \*rxy1, ILONGList \*ne1, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	网络号
rxy1	D_DotSet	[in]	区域边界坐标点集对象
ne1	LONGList	[in]	区域边界每一圈点数
rtl	Integer	[out]	相交返回 1，否则返回 0

### InputExpresion 方法

描述 输入条件表达式。

Visual Basic rtl = Query.InputExpresion(stru, ttl, expStr)

C++ HRESULT pQuery-> InputExpresion(  
IRecord\_Head \*stru, BSTR ttl, BSTR \*expStr, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
stru	Record_Head	[in]	属性结构
ttl	String	[in]	窗口标题
expStr	String	[out]	条件表达式字符串
rtl	Integer	[out]	成功返回 1，失败 0

### ExpresionValid 方法

描述 表达式合法性检查。

Visual Basic rtl = Query.ExpresionValid(stru, expStr)

C++ HRESULT pQuery-> ExpresionValid (  
IRecord\_Head \*stru, BSTR \*expStr, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
stru	Record_Head	[in]	属性结构
expStr	String	[out]	条件表达式字符串
rtl	Integer	[out]	成功返回 1，失败 0

### InputExpOBJ 方法

描述 输入条件表达式,返回表达式对象。

Visual Basic Set myObj = Query.InputExpOBJ(stru, [ttl])

C++ HRESULT pQuery-> InputExpOBJ (  
IRecord\_Head \*stru, BSTR ttl=NULL, IExpObj \*\*myObj);

参数说明：

参数	类型	出口/入口	描述
stru	Record_Head	[in]	属性结构
ttl	String	[in]	窗口标题。 可选；缺省值：Nothing
myObj	ExpObj	[out]	返回表达式对象

### ExpStrValid 方法

描述 表达式合法性检查。

Visual Basic Set myObj = Query.ExpStrValid(stru, expStr)

C++ HRESULT pQuery-> ExpStrValid (  
IRecord\_Head \*stru, BSTR \*expStr, IExpObj \*\*myObj);

参数说明：

参数	类型	出口/入口	描述
stru	Record_Head	[in]	属性结构
expStr	String	[out]	条件表达式字符串
myObj	ExpObj	[out]	返回表达式对象

### ExpValue 方法

描述 计算表达式值。

Visual Basic val = Query.ExpValue(myObj, att, [domainErrNo])

C++ HRESULT pQuery-> ExpValue(  
IExpObj \*myObj, IRecord \*att, short \*domainErrNo=NULL0, double \*val);

参数说明：

参数	类型	出口/入口	描述
myObj	ExpObj	[in]	返回表达式对象

att	Record	[in]	属性记录
domainErrNo	Integer	[out]	表达式计算过程中的错误信息变量。 可选；缺省值：Nothing
val	Double	[out]	返回表达式计算结果

**dbAttAskToArea0 方法**

描述 属性检索，查询结果存入目的工作区。

Visual Basic num = Query.dbAttAskToArea0(atype, ai1, exp, expIsSQLWhereClause)

C++ HRESULT pQuery-> dbAttAskToArea0 (  
Enum\_Entity\_Type atype, BSTR exp, IWorkArea \*ai1,  
short expIsSQLWhereClause, long \* num);

参数说明：

参数	类型	出口/入口	描述
atype	Enum_Entity_Type	[in]	查询类型
exp	String	[in]	条件表达式字符串
ai1	WorkArea	[in]	目的工作区对象
expIsSQLWhereClause	Integer	[in]	=1 则 exp 是 SQL 查询语言的 WHERE 子句
num	Long	[out]	成功返回查询到的实体数 (>0)，否则返回 0

**dbAttAskToList0 方法**

描述 属性检索，查询结果存入实体号数组，并返回数组地址。

Visual Basic Set ptLst = Query.dbAttAskToList0(atype, exp, expIsSQLWhereClause)

C++ HRESULT pQuery-> dbAttAskToList0(  
Enum\_Entity\_Type atype, BSTR exp, short expIsSQLWhereClause,  
IIDList \*\*ptLst);

参数说明：

参数	类型	出口/入口	描述
atype	Enum_Entity_Type	[in]	查询类型
exp	String	[in]	条件表达式字符串
expIsSQLWhereClause	Integer	[in]	=1 则 exp 是 SQL 查询语言的 WHERE 子句
ptLst	IDList	[out]	成功返回实体号数组地址，否则返回 NULL

**dbLDateAskToList 方法**

**描述** 根据最后修改时间查找图元。

**Visual Basic** Set ptLst = Query.dbLDateAskToList(  
atype, beginDt, endDt, includeBeginDt, includeEndDt)

**C++** HRESULT pQuery->dbLDateAskToList(  
Enum\_Entity\_Type atype, ITimeStamp\_Stru \*beginDt,  
ITimeStamp\_Stru \*endDt, short includeBeginDt, short includeEndDt,  
IIDList \*\*ptLst);

**参数说明：**

参数	类型	出口/入口	描述
atype	Enum_Entity_Type	[in]	查询类型
beginDt	TimeStamp_Stru	[in]	开始时间
endDt	TimeStamp_Stru	[in]	结束时间
includeBeginDt	Integer	[in]	对应开始时间的点
includeEndDt	Integer	[in]	对应结束时间的点
ptLst	IDList	[out]	成功返回实体号数组地址，否则返回 NULL

**dbSetltime 方法**

**描述** 重新设置图元最后修改时间。

**Visual Basic** rtl = Query.dbSetltime(atype, i)

**C++** HRESULT pQuery->dbSetltime(  
Enum\_Entity\_Type atype, long i, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
atype	Enum_Entity_Type	[in]	查询类型
i	Long	[in]	实体号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbGetltime 方法**

**描述** 取图元最后修改时间。

**Visual Basic** rtl = Query.dbGetltime(atype, i, dt)

**C++** HRESULT pQuery->dbGetltime (  
Enum\_Entity\_Type atype, long i, ITimeStamp\_Stru \*\*dt,  
VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
----	----	-------	----

atype	Enum_Entity_Type	[in]	查询类型
i	Long	[in]	实体号
dt	TimeStamp_Stru	[out]	修改时间
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**AttAskToList 方法**

描述 属性检索，查询结果存入实体号数组，并返回数组地址。

Visual Basic Set ptLst = Query.AttAskToList(atype)

C++ HRESULT pQuery-> AttAskToList (  
Enum\_Entity\_Type atype, IIDList \*\*ptLst);

参数说明：

参数	类型	出口/入口	描述
atype	Enum_Entity_Type	[in]	查询类型
ptLst	IIDList	[out]	成功返回实体号数组地址， 否则返回 NULL

**PntInRect 方法**

描述 判断点是否在矩形框内。

Visual Basic rtl = Query.PntInRect(pnt, frc)

C++ HRESULT pQuery-> PntInRect(ID\_Dot \*pnt, ID\_Rect \*frc, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pnt	ID_Dot	[in]	判断点对象
frc	ID_Rect	[in]	矩形范围
rtl	Integer	[out]	在框内返回 2，在矩形框边界上返回 1， 在框外返回 0

**RegInterRect 方法**

描述 判断已知区域是否与矩形框相交。

Visual Basic rtl = Query.RegInterRect(rdat, rc, [rrc])

C++ HRESULT pQuery-> RegInterRect(  
ILongList \*rdat, ID\_Rect \*rc, ID\_Rect \*rrc=NULL, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
rdat	ILongList	[in]	区数据
rc	ID_Rect	[in]	矩形范围

rrc	D_Rect	[in]	区域范围。 可选；缺省值：Nothing
rtl	Integer	[out]	相交则返回 1，否则返回 0

**NetInterRect 方法**

描述 判断已知网络是否与矩形框相交。

Visual Basic rtl = Query.NetInterRect(ndat, rc, [nrc])

C++ HRESULT pQuery->NetInterRect (  
ILONGList \*ndat, ID\_Rect \*rc, ID\_Rect \*nrc=NULL, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ndat	LONGList	[in]	网数据
rc	D_Rect	[in]	矩形范围
nrc	D_Rect	[in]	网络范围。 可选；缺省值：Nothing
rtl	Integer	[out]	相交则返回 1，否则返回 0

**LinCrossRect 方法**

描述 判断某条线是否与矩形框相交。

Visual Basic rtl = Query.LinCrossRect(li, rc)

C++ HRESULT pQuery->LinCrossRect (  
long li, ID\_Rect \*rc, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	线号
rc	D_Rect	[in]	矩形范围
rtl	Integer	[out]	相交则返回 1，否则返回 0

**RegCrossRect 方法**

描述 判断某个区是否与矩形框相交。

Visual Basic rtl = Query.RegCrossRect(ri, rc)

C++ HRESULT pQuery->RegCrossRect (  
long ri, ID\_Rect \*rc, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ri	Long	[in]	区号
rc	D_Rect	[in]	矩形范围



rtl	Integer	[out]	相交则返回 1，否则返回 0
-----	---------	-------	----------------

**NetCrossRect 方法**

描述 判断某个网是否与矩形框相交。

Visual Basic rtl = Query.NetCrossRect(ni, rc)

C++ HRESULT pQuery-> NetCrossRect (  
long ni, ID\_Rect \*rc, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ni	Long	[in]	区号
rc	ID_Rect	[in]	矩形范围
rtl	Integer	[out]	相交则返回 1，否则返回 0

**dbSQLAskToArea 方法**

描述 SQL 图形和属性综合查询，结果存入目的工作区。

Visual Basic entityNum = Query.dbSQLAskToArea(  
type, ai1, attSQLWhereClause, [attBindParam], [attBindLen],  
grpSQLWhereClause, [grpBindParam], [grpBindLen], [and\_orFlag])

C++ HRESULT pQuery-> dbSQLAskToArea(  
Enum\_Entity\_Type type, IWorkArea \*ai1, BSTR attSQLWhereClause,  
double \*attBindParam=NULL, short attBindLen=0,  
BSTR grpSQLWhereClause,  
double \*grpBindParam=NULL, short grpBindLen=0,  
short and\_orFlag, long \*entityNum);

参数说明：

参数	类型	出口/入口	描述
type	Enum_Entity_Type	[in]	查询类型
ai1	WorkArea	[in]	目的工作区
attSQLWhereClause	String	[in]	属性条件 where 子句
attBindParam	Double	[in]	属性条件参数数组
attBindLen	Integer	[in]	属性条件参数长度
grpSQLWhereClause	String	[in]	图形参数条件 where 子句
grpBindParam	Double	[in]	图形条件参数数组
grpBindLen	Integer	[in]	图形条件参数长度
and_orFlag	Integer	[in]	属性 where 子句和图形 where 子句逻辑运算标志， 1=AND，2=OR

entityNum	Long	[out]	成功,则返回实体号数组地址, 否则返回 NULL
-----------	------	-------	--------------------------

### dbSQLAskToList 方法

描述 SQL 图形和属性综合查询, 结果存入列表。

Visual Basic Set ptLst = Query.dbSQLAskToList(  
type, [attSQLWhereClause], [attBindParam], [attBindLen],  
[grpSQLWhereClause], [grpBindParam], [grpBindLen], [and\_orFlag])

C++ HRESULT pQuery-> dbSQLAskToList(  
Enum\_Entity\_Type type, BSTR attSQLWhereClause,  
double \*attBindParam=NULL,short attBindLen=0,  
BSTR grpSQLWhereClause,  
double \*grpBindParam=NULL,short grpBindLen=0,  
short and\_orFlag, IIDList \*\*ptLst);

参数说明：

参数	类型	出口/入口	描述
type	Enum_Entity_Type	[in]	查询类型
attSQLWhereClause	String	[in]	属性条件 where 子句
attBindParam	Double	[in]	属性条件参数数组
attBindLen	Integer	[in]	属性条件参数长度
grpSQLWhereClause	String	[in]	图形参数条件 where 子句
grpBindParam	Double	[in]	图形条件参数数组
grpBindLen	Integer	[in]	图形条件参数长度
and_orFlag	Integer	[in]	属性 where 子句和图形 where 子句逻辑运算标志, 1=AND, 2=OR
ptLst	IDList	[out]	成功返回实体号数组地址, 否则返回 NULL

### LinInterRect 方法

描述 判断已知曲线是否与矩形框相交。

Visual Basic rtl = Query.LinInterRect(xy, rc,[lrc])

C++ HRESULT pQuery-> LinInterRect(  
ID\_DotSet \*xy, ID\_Rect \*rc, ID\_Rect \*lrc=NULL, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
xy	D_DotSet	[in]	线坐标点集对象

rc	D_Rect	[in]	矩形范围
lrc	D_Rect	[in]	线范围。
			可选；缺省值：Nothing
rtl	Integer	[out]	相交则返回 1，否则返回 0

### 3.103、Analysis 对象

描述 Analysis 对象是分析对象。

#### 3.103.1、属性：无

#### 3.103.2、方法：

##### OverlayRegReg 方法

描述 区对区的叠加。

Visual Basic rtl = Analysis.OverlayRegReg(ai0, ai1, ai2, radiu, op)

C++ HRESULT pAnalysis-> OverlayRegReg(  
IRegArea \*ai0, IRegArea \*ai1, IRegArea \*ai2,  
double radiu, Enum\_Overlay\_Type op, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai0	RegArea	[in]	叠加工作区对象
ai1	RegArea	[in]	被叠加工作区对象
ai2	RegArea	[in]	结果工作区对象
radiu	Double	[in]	模糊结点半径
op	Enum_Overlay_Type	[in]	叠加操作类型
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

##### OverlayLinReg 方法

描述 线对区的叠加。

Visual Basic rtl = Analysis.OverlayLinReg(ai0, ai1, ai2, radiu, op)

C++ HRESULT pAnalysis-> OverlayLinReg(  
ILinArea \*ai0, IRegArea \*ai1, ILinArea \*ai2,  
double radiu, Enum\_Overlay\_Type op, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai0	LinArea	[in]	叠加工作区对象

ai1	RegArea	[in]	被叠加工作区对象
ai2	LinArea	[in]	结果工作区对象
radiu	Double	[in]	模糊结点半径
op	Enum_Overlay_Type	[in]	叠加操作类型
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**OverlayPntReg 方法**

描述 点对区的叠加。

Visual Basic rtl = Analysis.OverlayPntReg(ai0, ai1, ai2, radiu, op)

C++ HRESULT pAnalysis-> OverlayPntReg (  
IPntArea \*ai0, IRegArea \*ai1, IPntArea \*ai2,  
double radiu, Enum\_Overlay\_Type op, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai0	PntArea	[in]	叠加工作区对象
ai1	RegArea	[in]	被叠加工作区对象
ai2	PntArea	[in]	结果工作区对象
radiu	Double	[in]	模糊结点半径
op	Enum_Overlay_Type	[in]	叠加操作类型
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**OverlayPntLin 方法**

描述 点对线的叠加。

Visual Basic rtl = Analysis.OverlayPntLin(ai0, ai1, ai2, radiu, op)

C++ HRESULT pAnalysis-> OverlayPntLin (  
IPntArea \*ai0, ILinArea\*ai1, IPntArea \*ai2,  
double radiu, Enum\_Overlay\_Type op, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai0	PntArea	[in]	叠加工作区对象
ai1	LinArea	[in]	被叠加工作区对象
ai2	PntArea	[in]	结果工作区对象
radiu	Double	[in]	模糊结点半径
op	Enum_Overlay_Type	[in]	叠加操作类型
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**OverlayRegPnt 方法**

描述 区对点的叠加。

Visual Basic rtl = Analysis.OverlayRegPnt(ai0, ai1, ai2, radiu, op)

C++ HRESULT pAnalysis-> OverlayRegPnt (  
IRegAre \*ai0, IPntArea\*ai1, IRegAre \*ai2,  
double radiu, Enum\_Overlay\_Type op, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai0	RegAre	[in]	叠加工作区对象
ai1	PntArea	[in]	被叠加工作区对象
ai2	RegArea	[in]	结果工作区对象
radiu	Double	[in]	模糊结点半径
op	Enum_Overlay_Type	[in]	叠加操作类型
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ClipPnt 方法**

描述 裁剪点。

Visual Basic rtl = Analysis.ClipPnt(ai0, ai1, ai2, clipop)

C++ HRESULT pAnalysis-> ClipPnt (  
ILinArea \*ai0, IPntArea\*ai1, IPntArea \*ai2,  
Enum\_Overlay\_Type clipop, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai0	LinArea	[in]	裁剪框工作区对象
ai1	PntArea	[in]	被裁工作区对象
ai2	PntArea	[in]	结果工作区对象
clipop	Enum_Clip_Type	[in]	裁剪类型
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ClipLin 方法**

描述 裁剪线。

Visual Basic rtl = Analysis.ClipLin(ai0, ai1, ai2, radiu, clipop)

C++ HRESULT pAnalysis-> ClipLin (  
ILinArea \*ai0, ILinArea\*ai1, ILinArea \*ai2,  
double radiu, Enum\_Overlay\_Type clipop, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

ai0	LinArea	[in]	裁剪框工作区对象
ai1	LinArea	[in]	被裁工作区对象
ai2	LinArea	[in]	结果工作区对象
radiu	Double	[in]	模糊结点半径
clipop	Enum_Clip_Type	[in]	裁剪类型
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ClipReg 方法**

描述 裁剪区。

Visual Basic rtl = Analysis.ClipReg(ai0, ai1, ai2, radiu, clipop)

C++ HRESULT pAnalysis-> ClipReg (  
ILinArea \*ai0, IRegArea\*ai1, IRegArea \*ai2,  
double radiu, Enum\_Overlay\_Type clipop, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai0	LinArea	[in]	裁剪框工作区对象
ai1	RegArea	[in]	被裁工作区对象
ai2	RegArea	[in]	结果工作区对象
radiu	Double	[in]	模糊结点半径
clipop	Enum_Clip_Type	[in]	裁剪类型
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ClipRegGraph 方法**

描述 裁剪图形。

Visual Basic rtl = Analysis.ClipRegGraph(ai0, ai1, ai2, radiu,clipop)

C++ HRESULT pAnalysis-> ClipRegGraph (  
ILinArea \*ai0, IRegArea\*ai1, IRegArea \*ai2,  
double radiu ,Enum\_Overlay\_Type clipop, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai0	LinArea	[in]	裁剪框工作区对象
ai1	RegArea	[in]	被裁工作区对象
ai2	RegArea	[in]	结果工作区对象
clipop	Enum_Clip_Type	[in]	裁剪类型
radiu	Double	[in]	模糊结点半径
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**OneLinBuffer 方法**

描述 求一条线 BUFFER 区。

Visual Basic rtl = Analysis.OneLinBuffer(xy, frc, r, rai, useMode, knobFlg)

C++ HRESULT pAnalysis-> OneLinBuffer(  
ID\_DotSet \*xy, ID\_Rect \*frc, double r,  
IRegArea \*rai, Enum\_UseMode\_Type useMode,  
Enum\_KnobFlg\_Type knobFlg, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
xy	D_DotSet	[in]	线坐标点集对象
frc	D_Rect	[in]	线范围
r	Double	[in]	BUFFER 半径
rai	RegArea	[in]	结果工作区对象
useMode	Enum_UseMode_Type	[in]	求缓冲区方法类型（矢量/栅格）
knobFlg	Enum_KnobFlg_Type	[in]	缓冲区结点头类型（圆头/平头）
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**OneRegBuffer 方法**

描述 求一个区 BUFFER 区。

Visual Basic rtl = Analysis.OneRegBuffer(ai0, ri, r, rai, useMode)

C++ HRESULT pAnalysis-> OneRegBuffer(  
IRegArea \*ai0, long ri, double r, IRegArea \*rai,  
Enum\_UseMode\_Type useMode, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai0	RegArea	[in]	源工作区对象
ri	Long	[in]	区号
r	Double	[in]	BUFFER 半径
rai	RegArea	[in]	结果工作区对象
useMode	Enum_UseMode_Type	[in]	求缓冲区方法类型（矢量/栅格）
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**OnePntBuffer 方法**

描述 求一个点的 BUFFER 区。

Visual Basic rtl = Analysis.OnePntBufferr(xy, r, rai)

C++ HRESULT pAnalysis-> OnePntBuffer (  
ID\_Dot \*xy, double r, IRegArea \*rai, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
xy	D_Dot	[in]	点坐标点对象
r	Double	[in]	BUFFER 半径
rai	RegArea	[in]	结果工作区对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### TotalLinBuffer 方法

描述 求全部线 BUFFER 区。

Visual Basic rtl = Analysis.TotalLinBuffer(ai0, r, rai, useMode, knobFlg)

C++ HRESULT pAnalysis->TotalLinBuffer (ILinArea \*ai0, double r, IRegArea \*rai, Enum\_UseMode\_Type useMode, Enum\_KnobFlg\_Type knobFlg, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai0	LinArea	[in]	源工作区对象
r	Double	[in]	BUFFER 半径
rai	RegArea	[in]	结果工作区对象
useMode	Enum_UseMode_Type	[in]	求缓冲区方法类型（矢量/栅格）
knobFlg	Enum_KnobFlg_Type	[in]	缓冲区结点头类型（圆头/平头）
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### TotalRegBuffer 方法

描述 求全部区 BUFFER 区。

Visual Basic rtl = Analysis.TotalRegBuffer(ai0, r, rai, useMode)

C++ HRESULT pAnalysis->TotalRegBuffer (IRegArea \*ai0, double r, IRegArea \*rai, Enum\_UseMode\_Type useMode, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai0	RegArea	[in]	源工作区对象
r	Double	[in]	BUFFER 半径
rai	RegArea	[in]	结果工作区对象
useMode	Enum_UseMode_Type	[in]	求缓冲区方法类型（矢量/栅格）
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE



**TotalPntBuffer 方法**

描述 求全部点 BUFFER 区。

Visual Basic rtl = Analysis.TotalPntBuffer(ai0, r, rai, useMode)

C++ HRESULT pAnalysis->TotalPntBuffer (  
IPntArea \*ai0, double r, IRegArea \*rai  
Enum\_UseMode\_Type useMode, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai0	PntArea	[in]	源工作区对象
r	Double	[in]	BUFFER 半径
rai	RegArea	[in]	结果工作区对象
useMode	Enum_UseMode_Type	[in]	求缓冲区方法类型（矢量/栅格）
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ListLinBuffer 方法**

描述 求指定线的可变半径 BUFFER 区。

Visual Basic rtl = Analysis.ListLinBuffer(ai0, rBuf, rai, useMode, knobFlg)

C++ HRESULT pAnalysis->ListLinBuffer (  
ILinArea \*ai0, IRadSet \*rBuf,  
IRegArea \*rai, Enum\_UseMode\_Type useMode,  
Enum\_KnobFlg\_Type, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai0	LinArea	[in]	源工作区对象
rBuf	RadSet	[in]	BUFFER 可变半径对象
rai	RegArea	[in]	结果工作区对象
useMode	Enum_UseMode_Type	[in]	求缓冲区方法类型（矢量/栅格）
knobFlg	Enum_KnobFlg_Type	[in]	缓冲区结点头类型（圆头/平头）
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ListRegBuffer 方法**

描述 求指定区的可变半径 BUFFER 区。

Visual Basic rtl = Analysis.ListRegBuffer(ai0, rBuf, rai, useMode)

C++ HRESULT pAnalysis->ListRegBuffer (  
IRegArea\*ai0, IRadSet \*rBuf, IRegArea \*rai,  
Enum\_UseMode\_Type useMode, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai0	RegArea	[in]	源工作区对象
rBuf	RadSet	[in]	BUFFER 可变半径对象
rai	RegArea	[in]	结果工作区对象
useMode	Enum_UseMode_Type	[in]	求缓冲区方法类型（矢量/栅格）
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ListPntBuffer 方法**

描述 求指定点的可变半径 BUFFER 区。

Visual Basic rtl = Analysis.ListPntBuffer(ai0, rBuf, rai, useMode)

C++ HRESULT pAnalysis->ListPntBuffer (IPntArea\*ai0, IRadSet \*rBuf, IRegArea \*rai, Enum\_UseMode\_Type useMode, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ai0	PntArea	[in]	源工作区对象
rBuf	RadSet	[in]	BUFFER 可变半径对象
rai	RegArea	[in]	结果工作区对象
useMode	Enum_UseMode_Type	[in]	求缓冲区方法类型（矢量/栅格）
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**3.104、TraceDEM 对象**

描述 TraceDEM 对象是数字地形模型追踪对象，主要用来追踪网格化数字地形模型和等值线并形成区或线。

**3.103.1、属性：无****3.103.2、方法：****TraceGridDEM 方法**

描述 追踪网格化数字地形模型并形成区。

Visual Basic rtl = TraceDEM.TraceGridDEM(  
xn, yn, xl, yl, datN, dat, scln, scale, eps, linf, rinf, colscl, note, xs, rai, lai, pai)

C++ HRESULT pTraceDEM->TraceGridDEM(  
long xn, long yn, double xl, double yl, long datN, double \*dat,  
short scln, double \*scale, double eps, ILin\_Info \*linf,  
IReg\_Info \*rinf, short colscl, short note, short xs, IRegArea \*rai,

---

```
ILinArea *lai, IPntArea *pai, short *rtl);
```

参数说明：

参数	类型	出口/入口	描述
xn	Long	[in]	X 向点数
yn	Long	[in]	Y 向点数
xl	Double	[in]	X 向长度
yl	Double	[in]	Y 向长度
datN	Long	[in]	数据长度
dat	Double	[in]	原始数据
scln	Integer	[in]	标尺级数
scale	Double	[in]	标尺刻度 ( 共 scln 个 )
eps	Double	[in]	抽稀系数 ( 0-1 之间 , 0 表示不进行抽稀 )
linf	Lin_Info	[in]	等值线图形参数
rinf	Reg_Info	[in]	填充图形参数
colscl	Integer	[in]	是否画色阶 ; 1/0 : 要/不要
note	Integer	[in]	是否加注记 ; 1/0 : 要/不要
xs	Integer	[in]	注记或色阶注记小数位数 ( colscl 或 note 为 1 时有效 )
rai	RegArea	[in]	打开的区工作区对象
lai	LinArea	[in]	打开的线工作区对象
pai	PntArea	[in]	打开的点工作区对象
rtl	Integer	[out]	成功返回 1 , 失败返回 0

---

**TraceBinDEMToReg 方法**

描述 追踪一层等值线并形成区。

Visual Basic    rtl = TraceDEM.TraceBinDEMToReg(  
                   xn, yn, xl, yl, datN, dat, eps, linf, rinf, rai)

C++            HRESULT pTraceDEM-> TraceBinDEMToReg (   
                   long xn, long yn, double xl, double yl, long datN, short \*dat,   
                   double eps, ILin\_Info \*linf, IReg\_Info \*rinf, IRegArea \*rai, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
xn	Long	[in]	X 向点数
yn	Long	[in]	Y 向点数
xl	Double	[in]	X 向长度
yl	Double	[in]	Y 向长度
datN	Long	[in]	数据长度

dat	Integer	[in]	原始数据
eps	Double	[in]	抽稀系数 ( 0-1 之间 , 0 表示不进行抽稀 )
linf	Lin_Info	[in]	线参数对象
rinf	Reg_Info	[in]	区参数对象
rai	RegArea	[in]	打开的区工作区对象
rtl	Integer	[out]	成功返回 1 , 失败返回 0

**TraceBinDEMToLin 方法**

描述 追踪一层等值线并形成线。

Visual Basic rtl = TraceDEM.TraceBinDEMToLin(

xn, yn, xl, yl, datN, dat, eps, linf, lai)

C++ HRESULT pTraceDEM->TraceBinDEMToLin (

long xn, long yn, double xl, double yl, long datN, short \*dat,

double eps, ILin\_Info \*linf, ILineArea \*lai, short \*rtl);

参数说明 :

参数	类型	出口/入口	描述
xn	Long	[in]	X 向点数
yn	Long	[in]	Y 向点数
xl	Double	[in]	X 向长度
yl	Double	[in]	Y 向长度
datN	Long	[in]	数据长度
dat	Integer	[in]	原始数据
eps	Double	[in]	抽稀系数 ( 0-1 之间 , 0 表示不进行抽稀 )
linf	Lin_Info	[in]	线参数对象
lai	Line Area	[in]	打开的线工作区对象
rtl	Integer	[out]	成功返回 1 , 失败返回 0

**TraceBinDEMForBuf 方法**

描述 追踪一层等值线并形成区。

Visual Basic rtl = TraceDEM.TraceBinDEMForBuf(

xn, yn, xl, yl, datN, dat, eps, linf, rinf, rai)

C++ HRESULT pTraceDEM->TraceBinDEMForBuf (

long xn, long yn, double xl, double yl, long datN, short \*dat,

double eps, ILin\_Info \*linf, IReg\_Info \*rinf, IRegArea \*rai, short \*rtl);

参数说明 :

参数	类型	出口/入口	描述
xn	Long	[in]	X 向点数

yn	Long	[in]	Y 向点数
xl	Double	[in]	X 向长度
yl	Double	[in]	Y 向长度
datN	Long	[in]	数据长度
dat	Integer	[in]	原始数据
eps	Double	[in]	抽稀系数 ( 0-1 之间 , 0 表示不进行抽稀 )
linf	Lin_Info	[in]	线参数对象
rinf	Reg_Info	[in]	区参数对象
rai	RegArea	[in]	打开的区工作区对象
rtl	Integer	[out]	成功返回 1 , 失败返回 0

**SetNoteParam 方法**

描述 设置注记参数 ( 字符串高、最大角度、最小间隔 )。

Visual Basic rtl = TraceDEM.SetNoteParam([txtH], [maxAng], [minSpace])

C++ HRESULT pTraceDEM-> SetNoteParam(  
double txtH=4.0, double maxAng=70.0, double minSpace=80.0,  
VARIANT\_BOOL \*rtl);

参数说明 :

参数	类型	出口/入口	描述
txtH	Double	[in]	字符串高度 可选 ; 缺省值 : 4.0
maxAng	Double	[in]	最大角度 可选 ; 缺省值 : 70.0
minSpace	Double	[in]	最小间隔 可选 ; 缺省值 : 80.0
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

**GetNoteParam 方法**

描述 获取注记参数 ( 字符串高、最大角度、最小间隔 )。

Visual Basic rtl = TraceDEM.GetNoteParam(txtH, maxAng, minSpace)

C++ HRESULT pTraceDEM-> GetNoteParam (  
double txtH, double maxAng, double minSpace, VARIANT\_BOOL \*rtl);

参数说明 :

参数	类型	出口/入口	描述
txtH	Double	[out]	字符串高度
maxAng	Double	[out]	最大角度
minSpace	Double	[out]	最小间隔

rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

### 3.105、MapGis 对象

**描述** MapGis 全局对象。包含了一些全局通用的方法，主要针对基本数据类型对象中定义的数据进行一些计算和简单的判断。

#### 3.105.1、属性：

属性	描述
MapGisVersion	系统版本号
Long	
只读	

#### 3.105.2、方法：

##### GetLinDefParam 方法

**描述** 获取缺省的线实体图形信息参数。

Visual Basic Set inf = MapGis.GetLinDefParam()

C++ HRESULT pMapGis->GetLinDefParam (ILin\_Info \*\* inf);

**参数说明：**

参数	类型	出口/入口	描述
inf	Lin_Info	[out]	返回的 Lin_Info 对象

##### GetRegDefParam 方法

**描述** 获取缺省的区实体图形信息参数。

Visual Basic Set inf = MapGis.GetRegDefParam()

C++ HRESULT pMapGis->GetRegDefParam(IReg\_Info \*\* inf);

**参数说明：**

参数	类型	出口/入口	描述
inf	Reg_Info	[out]	返回的 Reg_Info 对象

##### GetNetDefParam 方法

**描述** 获取缺省的网实体图形信息参数。

Visual Basic Set inf = MapGis.GetNetDefParam()

C++ HRESULT pMapGis->GetNetDefParam(INet\_Info \*\* inf);

**参数说明：**

参数	类型	出口/入口	描述
----	----	-------	----

inf	Net_Info	[out]	返回的 Net_Info 对象
-----	----------	-------	-----------------

**GetPntDefParam 方法**

描述 获取缺省的点实体图形信息参数。

Visual Basic Set inf = MapGis.GetPntDefParam(pntType)

C++ HRESULT pMapGis->GetPntDefParam(  
Enum\_Pnt\_Type pntType, IPnt\_Info \*\* inf);

参数说明：

参数	类型	出口/入口	描述
pntType	Enum_Pnt_Type	[in]	点类型
inf	Pnt_Info	[out]	返回的 Pnt_Info 对象

**SetLinDefParam 方法**

描述 设缺省的线实体图形信息参数。

Visual Basic rtl = MapGis.SetLinDefParam(inf)

C++ HRESULT pMapGis->SetLinDefParam(  
ILin\_Info \* inf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf	Lin_Info	[in]	线实体图形信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SetRegDefParam 方法**

描述 设缺省的区实体图形信息参数。

Visual Basic rtl = MapGis.SetRegDefParam(inf)

C++ HRESULT pMapGis-> SetRegDefParam (  
IReg\_Info \* inf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf	Reg_Info	[in]	区实体图形信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SetNetDefParam 方法**

描述 设缺省的网实体图形信息参数。

Visual Basic rtl = MapGis.SetNetDefParam(inf)

C++ HRESULT pMapGis-> SetNetDefParam (  
INet\_Info \* inf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf	Net_Info	[in]	网实体图形信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### SetPntDefParam 方法

描述 设缺省的点实体图形信息参数。

Visual Basic rtl = MapGis.SetPntDefParam(inf)

C++ HRESULT pMapGis-> SetPntDefParam (  
IPnt\_Info \* inf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf	Pnt_Info	[in]	点实体图形信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### ModLinDefParam 方法

描述 修改缺省的线实体图形信息参数。

Visual Basic rtl = MapGis.ModLinDefParam()

C++ HRESULT pMapGis-> ModLinDefParam (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### ModRegDefParam 方法

描述 修改缺省的区实体图形信息参数。

Visual Basic rtl = MapGis.ModRegDefParam()

C++ HRESULT pMapGis-> ModRegDefParam (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### ModNetDefParam 方法

描述 修改缺省的网实体图形信息参数。

Visual Basic rtl = MapGis.ModNetDefParam()

C++ HRESULT pMapGis-> ModNetDefParam (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----



rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

**ModPntDefParam 方法**

描述 修改缺省的点实体图形信息参数。

Visual Basic rtl = MapGis.ModPntDefParam()

C++ HRESULT pMapGis-> ModPntDefParam (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CompLinParam 方法**

描述 比较图形参数。

Visual Basic rtl = MapGis.CompLinParam(pad, inf0, inf1 )

C++ HRESULT pMapGis->CompLinParam(  
ILin\_Pad \*pad, ILin\_Info \*inf0, ILin\_Info \*inf1, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Lin_Pad	[in]	图形参数开关
inf0	Lin_Info	[in]	图形参数 1
inf1	Lin_Info	[in]	图形参数 2
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CompRegParam 方法**

描述 比较图形参数。

Visual Basic rtl = MapGis.CompRegParam(pad, inf0, inf1)

C++ HRESULT pMapGis-> CompRegParam (  
IReg\_Pad \*pad, IReg\_Info \*inf0, IReg\_Info \*inf1, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Reg_Pad	[in]	图形参数开关
inf0	Reg_Info	[in]	图形参数 1
inf1	Reg_Info	[in]	图形参数 2
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CompNetParam 方法**

描述 比较图形参数。

Visual Basic rtl = MapGis.CompNetParam(pad, inf0, inf1)

C++ HRESULT pMapGis-> CompNetParam (  
INet\_Pad \*pad, INet\_Info \*inf0, INet\_Info \*inf1, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Net_Pad	[in]	图形参数开关
inf0	Net_Info	[in]	图形参数 1
inf1	Net_Info	[in]	图形参数 2
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### CompPntParam 方法

描述 比较图形参数。

Visual Basic rtl = MapGis.CompPntParam(pad, inf0, inf1)

C++ HRESULT pMapGis-> CompPntParam (  
IPnt\_Pad \*pad, IPnt\_Info \*inf0, IPnt\_Info \*inf1, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	图形参数开关
inf0	Pnt_Info	[in]	图形参数 1
inf1	Pnt_Info	[in]	图形参数 2
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### CalDirection 方法

描述 判断封闭曲线走向。

Visual Basic dir = MapGis.CalDirection(xy)

C++ HRESULT pMapGis-> CalDirection(ID\_DotSet \*xy, short \*dir);

参数说明：

参数	类型	出口/入口	描述
xy	D_DotSet	[in]	曲线坐标点集对象
dir	Integer	[out]	封闭曲线顺时针旋转返回 1， 逆时针旋转返回-1，否则返回 0。

### PntInRect 方法

描述 判断点是否在矩形区内。

Visual Basic rtl = MapGis.PntInRect(pnt, frc)

C++ HRESULT pMapGis-> PntInRect(ID\_Dot \*pnt, ID\_Rect \*frc, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

pnt	D_Dot	[in]	点参数对象
frc	D_Rect	[in]	矩形范围对象
rtl	Integer	[out]	pnt 点在 frc 矩形框内返回 2，在矩形框边界上返回 1，在框外返回 0

#### IsDotInPolygon0 方法

描述 判断点是否在多边形内。

Visual Basic rtl = MapGis.IsDotInPolygon0(xy, dot, esp)

C++ HRESULT pMapGis-> IsDotInPolygon0(  
ID\_DotSet \*xy, ID\_Dot \*dot,double esp, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
xy	D_DotSet	[in]	多边形坐标点集对象，不要求封闭
dot	D_Dot	[in]	点参数对象
esp	Double	[in]	允许落在多边形上的限度
rtl	Integer	[out]	dot 点在多边形内返回 2， 在边线上返回 1，在多边形外返回 0

#### IsDotInPolygon 方法

描述 判断点是否在多边形内。

Visual Basic rtl = MapGis.IsDotInPolygon(xy, dot)

C++ HRESULT pMapGis-> IsDotInPolygon(  
ID\_DotSet \*xy, ID\_Dot \*dot,short \*rtl);

参数说明：

参数	类型	出口/入口	描述
xy	D_DotSet	[in]	多边形坐标点集对象，不要求封闭
dot	D_Dot	[in]	点参数对象
rtl	Integer	[out]	dot 点在多边形内返回 2， 在边线上返回 1，在多边形外返回 0

#### LinInterRect 方法

描述 判断已知曲线是否与矩形框相交。

Visual Basic rtl = MapGis.LinInterRect(xy, rc, [lrc])

C++ HRESULT pMapGis-> LinInterRect(  
ID\_DotSet \*xy, ID\_Rect \*rc, ID\_Rect \*lrc= NULL, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

xy	D_DotSet	[in]	曲线坐标点集对象
rc	D_Rect	[in]	矩形范围对象
lrc	D_Rect	[in]	曲线范围对象
rtl	Integer	[out]	若曲线与矩形框相交或被矩形框包含则返回 1，否则返回 0

**DistOfPntToPnt 方法**

描述 计算点到点之间的距离。

Visual Basic dist = MapGis.DistOfPntToPnt(pnt0,pnt1)

C++ HRESULT pMapGis-> DistOfPntToPnt(  
ID\_Dot \*pnt0, ID\_Dot \*pnt1, double \*dist);

参数说明：

参数	类型	出口/入口	描述
pnt0	D_Dot	[in]	点对象
pnt1	D_Dot	[in]	点对象
dist	Double	[out]	成功返回距离，<=0 时失败

**DistOfPntToLin 方法**

描述 计算点到线之间的距离。

Visual Basic dist = MapGis.DistOfPntToLin(pnt, xy)

C++ HRESULT pMapGis-> DistOfPntToLin (  
ID\_Dot \*pnt, ID\_DotSet \*xy, double \*dist);

参数说明：

参数	类型	出口/入口	描述
pnt	D_Dot	[in]	点对象
xy	D_DotSet	[in]	线坐标点集对象
dist	Double	[out]	成功返回距离，<=0 时失败

**DistOfPntToReg0 方法**

描述 计算点到区之间的距离。

Visual Basic rtl = MapGis.DistOfPntToReg0(pnt, polxy, minDis, maxDis)

C++ HRESULT pMapGis-> DistOfPntToReg0(ID\_Dot \*pnt, ID\_DotSet \*polxy,  
double \*minDis, double \*maxDis, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pnt	D_Dot	[in]	点对象
polxy	D_DotSet	[in]	区边界坐标点集对象

minDis	Double	[out]	点到区域的最小距离, 若不需要, 可=NULL
maxDis	Double	[out]	点到区域的最大距离 若不需要, 可=NULL
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**MinDistOfPntToReg0 方法**

描述 计算点到区之间的最小距离。

Visual Basic minDis = MapGis.MinDistOfPntToReg0(pnt, polxy)

C++ HRESULT pMapGis->MinDistOfPntToReg0 (  
ID\_Dot \*pnt, ID\_DotSet \*polxy);

参数说明：

参数	类型	出口/入口	描述
pnt	ID_Dot	[in]	点对象
polxy	ID_DotSet	[in]	区边界坐标点集对象
minDis	Double	[out]	点到区域的最小距离

**MaxDistOfPntToReg0 方法**

描述 计算点到区之间的最大距离。

Visual Basic maxDis = MapGis.MaxDistOfPntToReg0(pnt, polxy)

C++ HRESULT pMapGis->MaxDistOfPntToReg0 (  
ID\_Dot \*pnt, ID\_DotSet \*polxy);

参数说明：

参数	类型	出口/入口	描述
pnt	ID_Dot	[in]	点对象
polxy	ID_DotSet	[in]	区边界坐标点集对象
maxDis	Double	[out]	点到区域的最大距离

**LinInterReg1 方法**

描述 判断线是否与区域相交。

Visual Basic rtl = MapGis.LinInterReg1(xy, rxy, ne)

C++ HRESULT pMapGis->LinInterReg1(  
ID\_DotSet \*xy, ID\_DotSet \*rxy, ILONGList \*ne, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
xy	ID_DotSet	[in]	线坐标点集对象
rxy	ID_DotSet	[in]	区边界坐标点集对象

ne	LONGList	[in]	区边界每一圈点数
rtl	Integer	[out]	相交返回 1，否则返回 0

**RegInterReg1 方法**

描述 判断两个区是否相交。

Visual Basic rtl = MapGis.RegInterReg1(rxy0, ne0, rxy1, ne1)

C++ HRESULT pMapGis->RegInterReg1(ID\_DotSet \*rxy0, ILONGList \*ne0, ID\_DotSet \*rxy1, ILONGList \*ne1, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
rxy0	D_DotSet	[in]	区边界坐标点集对象
ne0	LONGList	[in]	区边界每一圈点数
rxy1	D_DotSet	[in]	区边界坐标点集对象
ne1	LONGList	[in]	区边界每一圈点数
rtl	Integer	[out]	在区域外返回 0， 与区域相交（包括重叠）返回 1

**StartClockCursor 方法**

描述 准备显示闹钟光标。

Visual Basic MapGis.StartClockCursor

C++ HRESULT pMapGis->StartClockCursor();

参数说明： 无

**GoingClockCursor 方法**

描述 开始显示闹钟光标。

Visual Basic MapGis.GoingClockCursor

C++ HRESULT pMapGis->GoingClockCursor ();

参数说明： 无

**EndClockCursor 方法**

描述 结束显示闹钟光标。

Visual Basic MapGis.EndClockCursor

C++ HRESULT pMapGis->EndClockCursor ();

参数说明： 无

**StartWait 方法**

描述 开始显示等待光标。

Visual Basic MapGis.StartWait  
 C++ HRESULT pMapGis-> StartWait ();  
 参数说明： 无

**EndWait 方法**

描述 结束显示等待光标。  
 Visual Basic MapGis.EndWait  
 C++ HRESULT pMapGis-> EndWait ();  
 参数说明： 无

**ShowMapGisBmp 方法**

描述 显示 MAPGIS 点位图。  
 Visual Basic rtl = MapGis.ShowMapGisBmp()  
 C++ HRESULT pMapGis-> ShowMapGisBmp (VARIANT\_BOOL \*rtl);  
 参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CloseMapGisBmp 方法**

描述 关闭 MAPGIS 点位图。  
 Visual Basic rtl = MapGis.CloseMapGisBmp()  
 C++ HRESULT pMapGis->CloseMapGisBmp (VARIANT\_BOOL \*rtl);  
 参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetDiskSpace 方法**

描述 取磁盘剩余空间。  
 Visual Basic DiskSpace = MapGis.GetDiskSpace(dsk)  
 C++ HRESULT pMapGis-> GetDiskSpace(short dsk, long \*DiskSpace);  
 参数说明：

参数	类型	出口/入口	描述
dsk	Integer	[in]	磁盘号：1=A,2=B...
DiskSpace	Long	[out]	返回磁盘剩余空间的字节数

**Reduce 方法**

描述 抽稀函数。

Visual Basic    rtl = MapGis.Reduce(dot, eps)  
 C++            HRESULT pMapGis->Reduce(  
                  ID\_DotSet \*dot, double eps, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dot	D_DotSet	[in]	点集对象
eps	Double	[in]	抽稀系数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### IsDotInReg0 方法

描述            判断点是否在区域内。  
 Visual Basic    rtl = MapGis.IsDotInReg0(regNum, xy, ne, dot, flg, esp)

C++            HRESULT pMapGis->IsDotInReg0(  
                  long regNum, ID\_DotSet \*xy, ILONGList \*ne, ID\_Dot \*dot,  
                  short flg, double esp, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
regNum	Long	[in]	区号
xy	D_DotSet	[in]	区边界坐标点集对象
ne	LONGList	[in]	区边界每一圈点数列表对象
dot	ID_Dot	[in]	点对象
flg	Integer	[in]	内圈判断标志，1/0：判断/不判断
eps	Double	[in]	允许在区域上的限度
rtl	Integer	[out]	dot 点在区域内返回 2， 在边线上返回 1，在区域外返回 0

### IsDotInReg 方法

描述            判断点是否在区域内。  
 Visual Basic    rtl = MapGis.IsDotInReg(regNum, xy, ne, dot, flg)

C++            HRESULT pMapGis->IsDotInReg(  
                  long regNum, ID\_DotSet \*xy, ILONGList \*ne, ID\_Dot \*dot,  
                  short flg, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
regnum	Long	[in]	区号
xy	D_DotSet	[in]	区边界坐标点集对象
ne	LONGList	[in]	区边界每一圈点数列表对象



dot	D_Dot	[in]	点对象
flg	Integer	[in]	内圈判断标志，1/0：判断/不判断
rtl	Integer	[out]	dot 点在区域内返回 2， 在边线上返回 1，在区域外返回 0

**Cal\_PIAngle 方法**

描述 计算直线角度。

Visual Basic angle = MapGis.Cal\_PIAngle(xy0, xy1)

C++ HRESULT pMapGis->Cal\_PIAngle(  
ID\_Dot \*xy0, ID\_Dot \*xy1, double \*angle);

参数说明：

参数	类型	出口/入口	描述
xy0	D_Dot	[in]	线起点坐标点对象
xy1	D_Dot	[in]	线终点坐标点对象
angle	Double	[out]	返回直线角度值，角度值用弧度制， 角度范围从-PI 到+PI

**Cal\_PIAngle1 方法**

描述 计算直线角度。

Visual Basic angle = MapGis.Cal\_PIAngle1(x0, y0, x1, y1)

C++ HRESULT pMapGis->Cal\_PIAngle1(  
double x0, double y0, double x1, double y1, double \*angle);

参数说明：

参数	类型	出口/入口	描述
x0	Double	[in]	直线起点坐标 X
y0	Double	[in]	直线起点坐标 Y
x1	Double	[in]	直线起点坐标 X
y1	Double	[in]	直线起点坐标 Y
angle	Double	[out]	返回直线角度值，角度值用弧度制， 角度范围从-PI 到+PI

**Cal\_2PIAngle 方法**

描述 计算直线角度。

Visual Basic angle = MapGis.Cal\_2PIAngle(xy0, xy1)

C++ HRESULT pMapGis->Cal\_2PIAngle (  
ID\_Dot \*xy0, ID\_Dot \*xy1, double \*angle);

参数说明：

参数	类型	出口/入口	描述
xy0	D_Dot	[in]	线起点坐标点对象
xy1	D_Dot	[in]	线终点坐标点对象
angle	Double	[out]	返回直线角度值，角度值用弧度制， 角度范围从 0 到 2PI

**Cal\_2PIAngle1 方法**

描述 计算直线角度。

Visual Basic angle = MapGis.Cal\_2PIAngle1(x0, y0, x1, y1)

C++ HRESULT pMapGis->Cal\_2PIAngle1(  
double x0, double y0, double x1, double y1, double \*angle);

参数说明：

参数	类型	出口/入口	描述
x0	Double	[in]	直线起点坐标 X
y0	Double	[in]	直线起点坐标 Y
x1	Double	[in]	直线终点坐标 X
y1	Double	[in]	直线终点坐标 Y
angle	Double	[out]	返回直线角度值，角度值用弧度制， 角度范围从 0 到 2PI

**Cal\_PIAngleOfArr1 方法**

描述 计算向量角度。

Visual Basic angle = MapGis.Cal\_PIAngleOfArr1(dx, dy)

C++ HRESULT pMapGis->Cal\_PIAngleOfArr1(  
double dx, double dy, double \*angle);

参数说明：

参数	类型	出口/入口	描述
dx	Double	[in]	向量坐标 X
dy	Double	[in]	向量坐标 Y
angle	Double	[out]	返回向量角度值，角度值用弧度制， 角度范围从 -PI 到 +PI

**Cal\_PIAngleOfArr 方法**

描述 计算向量角度。

Visual Basic angle = MapGis.Cal\_PIAngleOfArr(xy)

C++ HRESULT pMapGis->Cal\_PIAngleOfArr (ID\_Dot \*xy, double \*angle);

参数说明：

参数	类型	出口/入口	描述
xy	D_Dot	[in]	向量坐标点对象
angle	Double	[out]	返回向量角度值，角度值用弧度制， 角度范围从-PI 到+PI

**Cal\_2PIAngleOfArr1 方法**

描述 计算向量角度。

Visual Basic angle = MapGis.Cal\_2PIAngleOfArr1(dx, dy)

C++ HRESULT pMapGis->Cal\_2PIAngleOfArr1(  
double dx, double dy, double \*angle);

参数说明：

参数	类型	出口/入口	描述
dx	Double	[in]	向量坐标 X
dy	Double	[in]	向量坐标 Y
angle	Double	[out]	返回向量角度值，角度值用弧度制， 角度范围从 0 到 2PI

**Cal\_2PIAngleOfArr 方法**

描述 计算向量角度。

Visual Basic angle = MapGis.Cal\_2PIAngleOfArr(xy)

C++ HRESULT pMapGis->Cal\_2PIAngleOfArr(ID\_Dot \*xy, double \*angle);

参数说明：

参数	类型	出口/入口	描述
xy	D_Dot	[in]	向量坐标点对象
angle	Double	[out]	返回向量角度值，角度值用弧度制， 角度范围从 0 到 2PI

**SegmentLength 方法**

描述 计算线段长度。

Visual Basic length = MapGis.SegmentLength(xy0, xy1)

C++ HRESULT pMapGis->SegmentLength(  
ID\_Dot \*xy0, ID\_Dot \*xy1, double \*length);

参数说明：

参数	类型	出口/入口	描述
xy0	D_Dot	[in]	直线起点坐标对象
xy1	D_Dot	[in]	直线终点坐标对象
length	Double	[out]	返回直线段长度

**SegmentLength1 方法**

描述 计算线段长度。

Visual Basic length = MapGis.SegmentLength1(x0 , y0, x1 , y1)

C++ HRESULT pMapGis->SegmentLength1(  
double x0, double y0, double x1, double y1, double \*length);

参数说明：

参数	类型	出口/入口	描述
x0	Double	[in]	直线起点坐标 X
y0	Double	[in]	直线起点坐标 Y
x1	Double	[in]	直线终点坐标 X
y1	Double	[in]	直线终点坐标 Y
length	Double	[out]	返回直线段长度

**CalculateLength 方法**

描述 计算线长度。

Visual Basic length = MapGis.CalculateLength(xy)

C++ HRESULT pMapGis->CalculateLength(ID\_DotSet \*xy, double \*length);

参数说明：

参数	类型	出口/入口	描述
xy	D_DotSet	[in]	线坐标点集对象
length	Double	[out]	返回直线长度

**CalculateOneArea 方法**

描述 计算单个区域面积。

Visual Basic area = MapGis.CalculateOneArea(xy)

C++ HRESULT pMapGis->CalculateOneArea(ID\_DotSet \*xy, double \*area);

参数说明：

参数	类型	出口/入口	描述
xy	D_DotSet	[in]	区边界坐标点集对象
area	Double	[out]	返回区域面积 ,<0 时表示该区域边界弧段逆时针旋转 ,>0 时表示该区边界弧段顺时针旋转。

**ThreeDotToArc 方法**

描述 已知弧上开始点、中间点、结束点,求弧的圆心、半径、开始角、结束角。

Visual Basic rtl = MapGis.ThreeDotToArc(

```

C++      cirArcFlg, stax, stay, cenx, ceny, endx, endy, x, y, r, ang1, ang2)
HRESULT pMapGis->ThreeDotToArc(
int cirArcFlg, double stax, double stay, double cenx, double ceny,
double endx, double endy, double *x, double *y, double *r, double *ang1,
double *ang2, VARIANT_BOOL*rtl);
    
```

参数说明：

参数	类型	出口/入口	描述
cirArcFlg	Integer	[in]	判断三点求圆还是弧，0/1=圆/弧
stax	Double	[in]	起始点 X 坐标
stay	Double	[in]	起始点 Y 坐标
cenx	Double	[in]	中间点 X 坐标
ceny	Double	[in]	中间点 Y 坐标
endx	Double	[in]	结束点 X 坐标
endy	Double	[in]	结束点 Y 坐标
x	Double	[out]	返回圆心 X 坐标
y	Double	[out]	返回圆心 Y 坐标
r	Double	[out]	返回半径
ang1	Double	[out]	返回起始角
ang2	Double	[out]	返回终止角
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### Cal\_Arc\_xyab 方法

描述 已知拐角线上的开始点、顶点、结束点和半径，  
求倒角弧上的开始点、中间点、结束点。

```
Visual Basic  rtl = MapGis.Cal_Arc_xyabc(pa, pb, pc, radiu, poa, pob, poc)
```

```

C++      HRESULT pMapGis->Cal_Arc_xyabc(
ID_Dot *pa, ID_Dot *pb, ID_Dot *pc, double radiu,
ID_Dot **poa, ID_Dot **pob, ID_Dot **poc, VARIANT_BOOL *rtl);
    
```

参数说明：

参数	类型	出口/入口	描述
pa	ID_Dot	[in]	起始点对象
pb	ID_Dot	[in]	顶点对象
pc	ID_Dot	[in]	结束点对象
radiu	Double	[in]	倒角半径
poa	ID_Dot	[out]	返回倒角弧上起始点对象
pob	ID_Dot	[out]	返回倒角弧上顶点对象
poc	ID_Dot	[out]	返回倒角弧上结束点对象

rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

**CalCrossDot 方法**

描述 计算两直线段的交点。

Visual Basic rtl = MapGis.CalCrossDot(la0, la1, lb0, lb1, xy)

C++ HRESULT pMapGis->CalCrossDot(ID\_Dot \*la0, ID\_Dot \*la1,  
ID\_Dot \*lb0, ID\_Dot \*lb1, ID\_Dot \*\*xy, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
la0	D_Dot	[in]	线段 1 的起点对象
la1	D_Dot	[in]	线段 1 的终点对象
lb0	D_Dot	[in]	线段 2 的起点对象
lb1	D_Dot	[in]	线段 2 的终点对象
xy	D_Dot	[out]	两直线的交点对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SelDotOfLin 方法**

描述 查找线上最近点。

Visual Basic dotNo = MapGis.SelDotOfLin(xy, dot)

C++ HRESULT pMapGis->SelDotOfLin(  
ID\_DotSet \*xy, ID\_Dot \*dot, long \*dotNo);

参数说明：

参数	类型	出口/入口	描述
xy	D_DotSet	[in]	线坐标点对象
dot	D_Dot	[in]	查找位置点对象
dotNo	Long	[out]	成功返回点序号 (>=0)，失败返回-1

**DispErrorMsg 方法**

描述 显示错误信息。

Visual Basic rtl = MapGis.DispErrorMsg(errNo)

C++ HRESULT pMapGis->DispErrorMsg(short errNo, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
errNo	Integer	[in]	错误号
rtl	Integer	[out]	返回 IDOK 或 IDCANCEL

ErrNo：下列错误号宏定义之一：

宏定义	错误类型
NOT_SYS_FILE	非本系统文件
NOT_ENOUGH_MEM	内存不够，请关闭无用进程
ERROR_VER	错误的版本
ERROR_FILE_TYPE	错误的文件类型
ERROR_OPENFILE	打开文件错
NOT_ENOUGH_FHANDLE	文件句柄不够
NOT_ENOUGH_DISK	没有足够的磁盘空间
LINATT_NOTSAME	线属性结构不相同
PNTATT_NOTSAME	点属性结构不相同
REGATT_NOTSAME	区属性结构不相同
NODATT_NOTSAME	结点属性结构不相同
NETATT_NOTSAME	网络属性结构不相同
CANNOT_OPEN_SUBGR	不能打开子图库
CANNOT_OPEN_FILGR	不能打开图案库
CANNOT_OPEN_LINSTY	不能打开线型库
CANNOT_OPEN_COLOR	不能打开颜色库
CANNOT_OPEN_CCLIB	不能打开汉字库

**EditLinParam 方法**

描述 编辑线图形参数。

Visual Basic rtl = MapGis.EditLinParam(inf, ttl)

C++ HRESULT pMapGis->EditLinParam(ILin\_Info \*inf, BSTR ttl, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf	Lin_Info	[in]	线图形参数对象
ttl	String	[in]	参数编辑窗口标题
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

**EditRegParam 方法**

描述 编辑区图形参数。

Visual Basic rtl = MapGis.EditRegParam(inf, ttl)

C++ HRESULT pMapGis->EditRegParam (IReg\_Info \*inf, BSTR ttl, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf	Reg_Info	[in]	区图形参数对象

ttl	String	[in]	参数编辑窗口标题
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

**EditNetParam 方法**

描述 编辑网图形参数。

Visual Basic rtl = MapGis.EditNetParam(inf, ttl)

C++ HRESULT pMapGis->EditNetParam (INet\_Info \*inf, BSTR ttl, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf	Net_Info	[in]	网图形参数对象
ttl	String	[in]	参数编辑窗口标题
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

**EditPntParam 方法**

描述 编辑点图形参数。

Visual Basic rtl = MapGis.EditPntParam(inf, ttl, [enableSelType])

C++ HRESULT pMapGis-> EditPntParam(  
IPnt\_Info \*inf, BSTR ttl, short enableSelType=0, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf	Pnt_Info	[in]	点图形参数对象
ttl	String	[in]	参数编辑窗口标题
enableSelType	Integer	[in]	点类型选择标志 ,0=不可选 ,1=可选。 可选；缺省值：0
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

**EditLinParamPad 方法**

描述 编辑线图形参数及开关板。

Visual Basic rtl = MapGis.EditLinParamPad(inf, pad, ttl)

C++ HRESULT pMapGis-> EditLinParamPad(  
ILin\_Info \*inf, ILin\_Pad \*pad, BSTR ttl, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf	Lin_Info	[in]	线图形参数对象



pad	Lin_Pad	[in]	线图形开关板对象
ttl	String	[in]	参数编辑窗口标题
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

#### EditRegParamPad 方法

描述 编辑区图形参数及开关板。

Visual Basic rtl = MapGis.EditRegParamPad(inf, pad, ttl)

C++ HRESULT pMapGis-> EditRegParamPad (  
IReg\_Info \*inf, IReg\_Pad \*pad, BSTR ttl, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf	Reg_Info	[in]	区图形参数对象
pad	Reg_Pad	[in]	区图形开关板对象
ttl	String	[in]	参数编辑窗口标题
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

#### EditNetParamPad 方法

描述 编辑网图形参数及开关板。

Visual Basic rtl = MapGis.EditNetParamPad(inf, pad, ttl)

C++ HRESULT pMapGis-> EditNetParamPad (  
INet\_Info \*inf, INet\_Pad \*pad, BSTR ttl, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf	Net_Info	[in]	网图形参数对象
pad	Net_Pad	[in]	网图形开关板对象
ttl	String	[in]	参数编辑窗口标题
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

#### EditPntParamPad 方法

描述 编辑点图形参数及开关板。

Visual Basic rtl = MapGis.EditPntParamPad(inf, pad, ttl, [enableSelType])

C++ HRESULT pMapGis-> EditPntParamPad(  
IPnt\_Info \*inf, IPnt\_Pad \*pad, BSTR ttl, short enableSelType=0, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf	Pnt_Info	[in]	点图形参数对象
pad	Pnt_Pad	[in]	点图形开关板对象
ttl	String	[in]	参数编辑窗口标题
enableSelType	Integer	[in]	点类型选择标志 ,0=不可选 ,1=可选。 可选；缺省值：0
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

**ReplaceLinParam 方法**

描述 修改线参数替换条件和替换结果。

Visual Basic rtl = MapGis.ReplaceLinParam(inf0, pad0, inf1, pad1)

C++ HRESULT pMapGis-> ReplaceLinParam(ILin\_Info \*inf0,  
ILin\_Pad \*pad0, ILin\_Info \*inf1, ILin\_Pad \*pad1, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf0	Lin_Info	[in]	线图形参数对象
pad0	Lin_Pad	[in]	线图形开关板对象
inf1	Lin_Info	[in]	线图形参数对象
pad1	Lin_Pad	[in]	线图形开关板对象
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

**ReplaceRegParam 方法**

描述 修改区参数替换条件和替换结果。

Visual Basic rtl = MapGis.ReplaceRegParam(inf0, pad0, inf1, pad1)

C++ HRESULT pMapGis-> ReplaceRegParam (IReg\_Info \*inf0,  
IReg\_Pad \*pad0, IReg\_Info \*inf1, IReg\_Pad \*pad1, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf0	Reg_Info	[in]	区图形参数对象
pad0	Reg_Pad	[in]	区图形开关板对象
inf1	Reg_Info	[in]	区图形参数对象
pad1	Reg_Pad	[in]	区图形开关板对象
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

**ReplaceNetParam 方法**

描述 修改网参数替换条件和替换结果。

Visual Basic rtl = MapGis.ReplaceNetParam(inf0, pad0, inf1, pad1)

C++ HRESULT pMapGis-> ReplaceNetParam (INet\_Info \*inf0,  
INet\_Pad \*pad0, INet\_Info \*inf1, INet\_Pad \*pad1, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf0	Net_Info	[in]	网图形参数对象
pad0	Net_Pad	[in]	网图形开关板对象
inf1	Net_Info	[in]	网图形参数对象
pad1	Net_Pad	[in]	网图形开关板对象
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

**ReplacePntParam 方法**

描述 修改点参数替换条件和替换结果。

Visual Basic rtl = MapGis.ReplacePntParam(inf0, pad0, inf1, pad1, [enableSelType])

C++ HRESULT pMapGis-> ReplacePntParam (  
IPnt\_Info \*inf0, IPnt\_Pad \*pad0, IPnt\_Info \*inf1, IPnt\_Pad \*pad1,  
short enableSelType=0, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
inf0	Pnt_Info	[in]	点图形参数对象
pad0	Pnt_Pad	[in]	点图形开关板对象
inf1	Pnt_Info	[in]	点图形参数对象
pad1	Pnt_Pad	[in]	点图形开关板对象
enableSelType	Integer	[in]	点类型选择标志 ,0=不可选 ,1=可选。 可选；缺省值：0
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

**SelectColor 方法**

描述 选择颜色。

Visual Basic rtl = MapGis.SelectColor(colNo)

C++ HRESULT pMapGis-> SelectColor(long \*colNo, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

colNo	Long	[out]	颜色号
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

**SelectSubGraph 方法**

描述 选择子图。

Visual Basic rtl = MapGis.SelectSubGraph(subNo)

C++ HRESULT pMapGis-> SelectSubGraph(long \*subNo, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
subNo	Long	[out]	子图号
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

**SelectPattern 方法**

描述 选择填充图案。

Visual Basic rtl = MapGis.SelectPattern(patNo)

C++ HRESULT pMapGis-> SelectPattern(short \*patNo, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
patNo	Integer	[out]	填充图案号
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

**SelectLineType 方法**

描述 选择线型。

Visual Basic rtl = MapGis.SelectLineType(ltp, fltp)

C++ HRESULT pMapGis-> SelectLineType(short \*ltp, short \*fltp, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ltp	Integer	[out]	线型号
fltp	Integer	[out]	辅助线型号
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

**SelectFontType 方法**

描述 选择字体。

Visual Basic    rtl = MapGis.SelectFontType(ztNo)  
C++             HRESULT pMapGis-> SelectFontType(short \*ztNo, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
ztNo	Integer	[out]	字体号
rtl	Integer	[out]	返回 0=失败， IDOK=成功，IDCANCEL=取消

### GetFileMapParam 方法

描述            从本地文件中取地图参数。

Visual Basic    rtl = MapGis.GetFileMapParam(fname, mpar, [off0])  
C++             HRESULT pMapGis-> GetFileMapParam(  
                  BSTR fname, IMap\_Para \*\*mpar, long off0=0, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名字符串
mpar	Map_Para	[out]	地图参数结构对象
off0	Long	[in]	MAPGIS 文件偏移。 可选；缺省值：0
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetFileMapParam1 方法

描述            从打开的文件中取地图参数。

Visual Basic    rtl = MapGis.GetFileMapParam1(fname, ftype, mpar,  
                  [off0], [szDSN], [szLOG], [szPSW])  
C++             HRESULT pMapGis-> GetFileMapParam1(BSTR fname,  
                  Enum\_Area\_Type ftype, IMap\_Para \*\*mpar, long off0=0,  
                  BSTR szDSN=NULL, BSTR szLOG=NULL, BSTR szPSW=NULL,  
                  VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名字符串
ftype	Enum_Area_Type	[in]	文件类型
mpar	Map_Para	[out]	地图参数结构对象
off0	Long	[in]	MAPGIS 文件偏移。 可选；缺省值：0

szDSN	String	[in]	ODBC 数据源名称。 可选；缺省值：Nothing
szLOG	String	[in]	登录号（登录名）。 可选；缺省值：Nothing
szPSW	String	[in]	登录口令。 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SetFileMapParam 方法**

描述 设地图参数到文件中。

Visual Basic rtl = MapGis.SetFileMapParam(fname, mpar, [off0])

C++ HRESULT pMapGis-> SetFileMapParam(  
BSTR fname, IMap\_Para \*mpar, long off0=0, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名字符串
mpar	Map_Para	[in]	地图参数结构对象
off0	Long	[in]	MAPGIS 文件偏移。 可选；缺省值：0
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SetFileMapParam1 方法**

描述 设地图参数到文件中。

Visual Basic rtl = MapGis.SetFileMapParam1(fname, ftype, mpar,  
[off0], [szDSN], [szLOG], [szPSW])

C++ HRESULT pMapGis-> SetFileMapParam1 (BSTR fname,  
Enum\_Area\_Type ftype, IMap\_Para \*\*mpar, long off0=0,  
BSTR szDSN=NULL, BSTR szLOG=NULL, BSTR szPSW=NULL,  
VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名字符串
ftype	Enum_Area_Type	[in]	文件类型
mpar	Map_Para	[in]	地图参数结构对象
off0	Long	[in]	MAPGIS 文件偏移。 可选；缺省值：0

szDSN	String	[in]	ODBC 数据源名称。 可选；缺省值：Nothing
szLOG	String	[in]	登录号（登录名）。 可选；缺省值：Nothing
szPSW	String	[in]	登录口令。 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbGetGisDSType 方法**

描述 取 MAPGIS 数据源类型。

Visual Basic type = MapGis.DbGetGisDSType()

C++ HRESULT pMapGis->dbGetGisDSType(Enum\_GisDS\_Type \*type);

参数说明：

参数	类型	出口/入口	描述
type	Enum_GisDS_Type	[out]	返回 PC_GIS=单机数据源； NET_GIS=网络数据源

**dbGetGisDSNumb 方法**

描述 取 MAPGIS 网络数据源个数。

Visual Basic num = MapGis.dbGetGisDSNumb()

C++ HRESULT pMapGis->dbGetGisDSNumb(short \*num);

参数说明：

参数	类型	出口/入口	描述
num	Integer	[out]	返回 MAPGIS 网络数据源个数

**dbGetGisDSName 方法**

描述 取第 i 项 MAPGIS 网络数据源名称。

Visual Basic rtl = MapGis.dbGetGisDSName(i, dsn)

C++ HRESULT pMapGis->dbGetGisDSName(  
short i, BSTR \*dsn, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
i	Integer	[in]	MAPGIS 网络数据源项数
dsn	String	[out]	第 i 项 MAPGIS 网络数据源名称， 长度必须大于等于 128 字节
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbGetDefGisDSName 方法**

描述 取 MAPGIS 缺省的网络数据源名称。

Visual Basic rtl = MapGis.dbGetDefGisDSName(dsn)

C++ HRESULT pMapGis-> dbGetDefGisDSName (  
BSTR \*dsn, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dsn	String	[out]	MAPGIS 缺省网络数据源名称， 长度必须大于等于 128 字节
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbGetFirstGisDSName 方法**

描述 取第一个 MAPGIS 数据源名称。

Visual Basic rtl = MapGis.dbGetFirstGisDSName(dsn)

C++ HRESULT pMapGis-> dbGetFirstGisDSName (  
BSTR \*dsn, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dsn	String	[out]	第一个 MAPGIS 数据源名称， 长度必须大于等于 128 字节
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbGetNextGisDSName 方法**

描述 取下一个 MAPGIS 数据源名称。

Visual Basic rtl = MapGis.dbGetNextGisDSName(dsn)

C++ HRESULT pMapGis-> dbGetNextGisDSName (  
BSTR \*dsn, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dsn	String	[out]	下一个 MAPGIS 数据源名称， 长度必须大于等于 128 字节
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbInputLogPsw 方法**

描述 输入登录号和登录口令。

Visual Basic rtl = MapGis.dbInputLogPsw(dsn, log, psw)

C++ HRESULT pMapGis-> dbInputLogPsw(



---

 BSTR dsn, BSTR log, BSTR psw, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dsn	String	[in]	MAPGIS 数据源名称
log	String	[in]	登录号（长度>=31 字节）
psw	String	[in]	登录口令（长度>=31 字节）
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbSetLogPsw 方法**

描述 设置登录号和登录口令。

Visual Basic rtl = MapGis.dbSetLogPsw (dsn, log, psw)

 C++ HRESULT pMapGis-> dbSetLogPsw (  
 BSTR dsn, BSTR log, BSTR psw, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dsn	String	[in]	MAPGIS 数据源名称
log	String	[in]	登录号（长度>=31 字节）
psw	String	[in]	登录口令（长度>=31 字节）
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dbGetLogPsw 方法**

描述 设置登录号和登录口令。

Visual Basic rtl = MapGis.dbGetLogPsw (dsn, log, psw)

 C++ HRESULT pMapGis-> dbGetLogPsw (  
 BSTR dsn, BSTR log, BSTR psw, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dsn	String	[in]	MAPGIS 数据源名称
log	String	[out]	登录号（长度>=31 字节）
psw	String	[out]	登录口令（长度>=31 字节）
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**PntTypeParamSame 方法**

描述 比较点图形参数。

Visual Basic rtl = MapGis.PntTypeParamSame(pad, inf, pinf)

 C++ HRESULT pMapGis-> PntTypeParamSame(  
 IPnt\_Pad \*pad, IPnt\_Info \*inf, IPnt\_Info \*pinf, VARIANT\_BOOL\*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点图形参数开关板对象
inf	Pnt_Info	[in]	点图形参数对象
pinf	Pnt_Info	[in]	点图形参数对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### dbTestDataSource 方法

描述 测试数据源。

Visual Basic rtl = MapGis.dbTestDataSource ( szDSN, szLOG, szPSW)

C++ HRESULT pMapGis-> dbTestDataSource ( BSTR szDSN, BSTR szLOG, BSTR szPSW, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
szDSN	String	[in]	ODBC 数据源名称
szLOG	String	[in]	登录号（登录名）
szPSW	String	[in]	登录口令
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetFileVersion 方法

描述 取文件版本号。

Visual Basic ver = MapGis.GetFileVersion(fname, [off0])

C++ HRESULT pMapGis-> GetFileVersion( BSTR fname, long off0=0, Enum\_Version \*ver);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	MAPGIS 文件名
off0	Long	[in]	MAPGIS 文件偏移。 可选；缺省值：0
ver	Enum_Version	[out]	成功返回文件版本号，否则返回 0

### MapGisAbout 方法

描述 显示 About 窗口,返回系统版本号。

Visual Basic ver = MapGis.MapGisAbout(title)

C++ HRESULT pMapGis-> MapGisAbout(BSTR title,Enum\_Version \*ver);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

title	String	[in]	窗口标题
ver	Enum_Version	[out]	成功返回文件版本号，否则返回 0

**InputFileName 方法**

描述 输入文件名。

Visual Basic rtl = MapGis.InputFileName(  
szName, dsn, owner, ftype, openflag, filter,[ttlStr])

C++ HRESULT pMapGis->InputFileName(  
BSTR \*szName, BSTR \*dsn, BSTR \*owner, Enum\_Area\_Type \*ftype,  
short openflag, BSTR filter, BSTR ttlStr=NULL, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
szName	String	[in,out]	文件名，长度至少要 128 字节
dsn	String	[out]	数据源名称 ( >=31 字节,若只取本地数据文件名可为 NULL )
owner	String	[out]	数据拥有者名称 ( >=31 字节,若只取本地数据文件名可为 NULL )
ftype	Enum_File_Type	[out]	文件类型 ( PNT/LIN/REG/NET ) ( 若只取本地数据文件名可为 NULL )
openflag	Integer	[in]	0/1：打开/存文件
filter	String	[in]	文件过滤字符串
ttlStr	String	[in]	输入窗标题字符串。 可选；缺省值：Nothing
rtl	Integer	[out]	成功返回 1，失败返回 0

**SelDirectory 方法**

描述 选择路径。

Visual Basic rtl = MapGis.SelDirectory(szPath, ttlStr)

C++ HRESULT pMapGis->SelDirectory(  
BSTR \*szPath, BSTR ttlStr, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
szPath	String	[in,out]	路径字符串
ttlStr	String	[in]	窗口标题

rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

**GetFileType 方法**

描述 取文件类型。

Visual Basic ftype = MapGis.GetFileType(fname, [off0], [ver])

C++ HRESULT pMapGis-> GetFileType(BSTR fname,  
long off0=0, Enum\_Version \*ver=NULL, Enum\_Area\_Type \*ftype);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in,out]	文件名，要求带路径
off0	Long	[in]	MAPGIS 文件偏移。 可选；缺省值：0
ver	Enum_Version	[out]	文件版本号。 可选；缺省值：Nothing
ftype	Enum_Area_Type	[out]	文件类型

**GetFileRange 方法**

描述 取文件范围。

Visual Basic rtl = MapGis.GetFileRange(fname, frc, [off0])

C++ HRESULT pMapGis-> GetFileRange(  
BSTR fname, ID\_Rect \*\*frc, long off0=0, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名，要求带路径
frc	D_Rect	[out]	文件的图形范围
off0	Long	[in]	MAPGIS 文件偏移。 可选；缺省值：0
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetFileRange1 方法**

描述 取文件范围。

Visual Basic rtl = MapGis.GetFileRange1(  
fname, ftype, frc, [off0], [szDSN], [szLOG], [szPSW])

C++ HRESULT pMapGis-> GetFileRange1(BSTR fname,  
Enum\_Area\_Type ftype, ID\_Rect \*\*frc, long off0=0, BSTR szDSN=NULL, BSTR  
szLOG=NULL, BSTR szPSW=NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名，要求带路径
ftype	Enum_Area_Type	[in]	文件类型
frc	D_Rect	[out]	文件的图形范围
off0	Long	[in]	MAPGIS 文件偏移。 可选；缺省值：0
szDSN	String	[in]	ODBC 数据源名称。 可选；缺省值：Nothing
szLOG	String	[in]	登录号（登录名）。 可选；缺省值：Nothing
szPSW	String	[in]	登录口令。 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetFileClass 方法**

描述 取文件分类码。

Visual Basic fcls = MapGis.GetFileClass(fname, [off0])

C++ HRESULT pMapGis-> GetFileClass(BSTR fname, long off0=0, short \*fcls);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名，要求带路径
off0	Long	[in]	MAPGIS 文件偏移。 可选；缺省值：0
fcls	Integer	[out]	成功返回文件分类码 (>=0)， 失败返回-1

**GetFileLayerPad 方法**

描述 取文件图层信息板。

Visual Basic rtl = MapGis.GetFileLayerPad(fname, layerPad, [off0])

C++ HRESULT pMapGis-> GetFileLayerPad(  
BSTR fname, ILayerOnOffPad \*\*layerPad, long off0=0, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名，要求带路径
layerPad	LayerOnOffPad	[out]	图层信息对象
off0	Long	[in]	MAPGIS 文件偏移。 可选；缺省值：0

rtl	Integer	[out]	成功返回读取的信息长度（以字节为单位）， 文件打不开则返回-1， 其他情况返回 0
-----	---------	-------	---

### GetFileAlias 方法

描述 取文件别名。

Visual Basic rtl = MapGis.GetFileAlias(fname, fAlias, [off0])

C++ HRESULT pMapGis-> GetFileAlias(  
BSTR fname, BSTR \*fAlias, long off0=0, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名，要求带路径
fAlias	String	[out]	文件别名空间
off0	Long	[in]	MAPGIS 文件偏移。 可选；缺省值：0
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### InputGraphFile 方法

描述 输入图形文件名称。

Visual Basic rtl = MapGis.InputGraphFile(  
szName, dsn, owner, ftype, openflag)

C++ HRESULT pMapGis-> InputGraphFile(  
BSTR \*szName, BSTR \*dsn, BSTR \*owner, Enum\_File\_Type ftype,  
short openflag, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
szName	String	[in,out]	文件名，长度至少要 128 字节
dsn	String	[out]	数据源名称 ( >=31 字节 ,若只取本地数据文件名可为 NULL )
owner	String	[out]	数据源名称 ( >=31 字节 ,若只取本地数据文件名可为 NULL )
ftype	Enum_File_Type	[in]	文件类型 ( PNT/LIN/REG/NET ) ( 若只取本地数据文件名可为 NULL )

### 第三章 MAPGIS 对象说明

---

openflag	Integer	[in]	0/1：打开/存文件
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

---

## 第四章 MAPGIS 控件说明

本章的示例代码包含在 D:\MAPGIS61\VBDemo\vbW60Area 目录下。

### 4.1、Arc\_Alloc\_InfoEdit 控件

**描述** Arc\_Alloc\_InfoEdit 控件用于可视化编辑 Arc\_Alloc\_Info 弧段分配信息。

#### 4.1.1、控件界面：如图 4-1 所示

图 4-1 Arc\_Alloc\_InfoEdit 控件界面

图 4-2 Lin\_InfoEdit 控件界面

#### 4.1.2、属性：

属性	描述
allocCenter	分配中心
Long	
可读写	
allocDirect	分配反向
Integer	
可读写	
cumuImped	累积阻碍
Double	
可读写	



demand	需求
Double	
可读写	
demandMeet	被满足的需求
Double	
可读写	
preArc	前一弧段
Long	
可读写	

### 4.1.3、方法：

#### SetValue 方法

描述 设置当前弧段分配信息参数值。

Visual Basic rtl = Arc\_Alloc\_InfoEdit1.SetValue(info)

C++ HRESULT pArc\_Alloc\_InfoEdit -> SetValue(  
IArc\_Alloc\_Info \*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Arc_Alloc_Info	[in]	弧段分配信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### GetValue 方法

描述 获取当前弧段分配信息参数值。

Visual Basic rtl = Arc\_Alloc\_InfoEdit1.GetValue(info)

C++ HRESULT pArc\_Alloc\_InfoEdit -> GetValue (  
IArc\_Alloc\_Info \*\*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Arc_Alloc_Info	[in]	弧段分配信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## 4.2、Lin\_InfoEdit 控件

描述 Lin\_InfoEdit 控件用于可视化编辑 Lin\_Info 线信息。

### 4.2.1、控件界面：如图 4-2 所示

#### 4.2.2、属性：

属性	描述
fclr Long 可读写	辅助颜色
fltp Integer 可读写	辅助线型
layer Integer 可读写	图层
lclass Integer 可读写	线种类
lclr Long 可读写	线颜色
ltp Integer 可读写	线型
lw Double 可读写	线宽
ovprnt Integer 可读写	透明输出
xscale Double 可读写	X 系数
yscale Double 可读写	Y 系数

#### 4.2.3、方法：

##### SetValue 方法

描述          设置当前线信息参数值。

Visual Basic    rtl = Lin\_InfoEdit1.SetValue(info)  
 C++            HRESULT pLin\_InfoEdit -> SetValue(  
                  ILin\_Info \*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Lin_Info	[in]	线信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### GetValue 方法

描述            获取当前线信息参数值。

Visual Basic    rtl = Lin\_InfoEdit1.GetValue(info)  
 C++            HRESULT pLin\_InfoEdit -> GetValue (  
                  ILin\_Info \*\*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Lin_Info	[in]	线信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### 4.3、Lin\_PadEdit 控件

描述            Lin\_PadEdit 控件用于可视化编辑 Lin\_Pad 线信息开关板。

#### 4.3.1、控件界面：

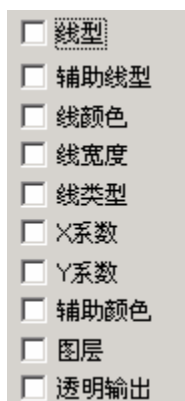


图 4-3 Lin\_PadEdit 控件界面

#### 4.3.2、属性：

属性	描述
fclr Enum_Switch_Type 可读写	辅助颜色
fltp Enum_Switch_Type 可读写	辅助线型
layer Enum_Switch_Type 可读写	图层
lclass Enum_Switch_Type 可读写	线种类
lclr Enum_Switch_Type 可读写	线颜色
ltp Enum_Switch_Type 可读写	线型
lw Enum_Switch_Type 可读写	线宽
ovprnt Enum_Switch_Type 可读写	透明输出
xscale Enum_Switch_Type 可读写	X 系数
yscale Enum_Switch_Type 可读写	Y 系数

### 4.3.3、方法：

#### SetValue 方法

描述	设置当前线信息参数开关板值。
Visual Basic	rtl = Lin_PadEdit1.SetValue(pad)
C++	HRESULT pLin_PadEdit -> SetValue(

---

```
ILin_Pad *pad, VARIANT_BOOL *rtl);
```

参数说明：

参数	类型	出口/入口	描述
pad	Lin_Pad	[in]	线信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### GetValue 方法

描述 获取当前线信息参数开关板值。

Visual Basic rtl = Lin\_PadEdit1.GetValue(pad)

C++ HRESULT pLin\_PadEdit -> GetValue (  
ILin\_Pad \*\*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Lin_Pad	[in]	线信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## 4.4、Lin\_TopEdit 控件

描述 Lin\_TopEdit 控件用于可视化编辑 Lin\_Top 线拓扑信息。

### 4.4.1、控件界面：

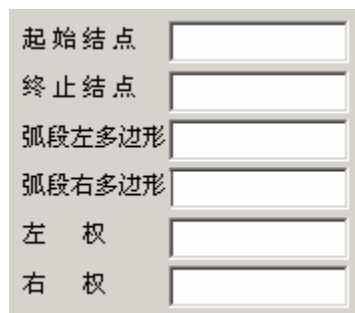


图 4-4 Lin\_TopEdit 控件界面

### 4.4.2、属性：

属性	描述
endNod	终止结点
Long	可读写

lPoly	弧段左多边形
Integer	
可读写	
lWeigh	左权
Double	
可读写	
rPoly	弧段右多边形
Long	
可读写	
rWeigh	右权
Double	
可读写	
stNod	起始结点
Long	
可读写	

#### 4.4.3、方法：

##### SetValue 方法

描述 设置当前线拓扑信息参数值。

Visual Basic rtl = Lin\_TopEdit1.SetValue(top)

C++ HRESULT pLin\_TopEdit -> SetValue(  
ILin\_Top \* top, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
top	Lin_Top	[in]	线拓扑信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

##### GetValue 方法

描述 获取当前线拓扑信息参数值。

Visual Basic rtl = Lin\_TopEdit1.GetValue(top)

C++ HRESULT pLin\_TopEdit -> GetValue (  
ILin\_Top \*\*top, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
top	Lin_Top	[in]	线拓扑信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

4.5、Net\_InfoEdit 控件

描述 Net\_InfoEdit 控件用于可视化编辑 Net\_Info 网信息。

4.5.1、控件界面：



图 4-5 Net\_InfoEdit 控件界面

4.5.2、属性：

属性	描述
fclr	辅助颜色
Long	
可读写	
fltp	辅助线型
Integer	
可读写	
layer	图层
Integer	
可读写	
lclass	线种类
Integer	
可读写	
lclr	颜色
Long	
可读写	

ltp	线型
Integer 可读写	
lw	线宽
Double 可读写	
ovprnt	透明输出
Integer 可读写	
xscale	X 系数
Double 可读写	
yscale	Y 系数
Double 可读写	

### 4.5.3、方法：

#### SetValue 方法

描述 设置当前网信息参数值。

Visual Basic rtl = Net\_InfoEdit1.SetValue(info)

C++ HRESULT pNet\_InfoEdit -> SetValue(  
INet\_Info \*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Net_Info	[in]	网信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### GetValue 方法

描述 获取当前网信息参数值。

Visual Basic rtl = Net\_InfoEdit1.GetValue(info)

C++ HRESULT pNet\_InfoEdit -> GetValue (  
INet\_Info \*\*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Net_Info	[in]	网信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE



4.6、Net\_PadEdit 控件

描述 Net\_PadEdit 控件用于可视化编辑 Net\_Pad 网信息开关板。

4.6.1、控件界面：



图 4-6 Net\_PadEdit 控件界面

4.6.2、属性：

属性	描述
fclr	辅助颜色
Enum_Switch_Type	
可读写	
fltp	辅助线型
Enum_Switch_Type	
可读写	
layer	图层
Enum_Switch_Type	
可读写	
lclass	线种类
Enum_Switch_Type	
可读写	
lclr	线颜色
Enum_Switch_Type	
可读写	

ltp	线型
Enum_Switch_Type	
可读写	
lw	线宽
Enum_Switch_Type	
可读写	
ovprnt	透明输出
Enum_Switch_Type	
可读写	
xscale	X 系数
Enum_Switch_Type	
可读写	
yscale	Y 系数
Enum_Switch_Type	
可读写	

#### 4.6.3、方法：

##### SetValue 方法

描述 设置当前网信息参数开关板值。

Visual Basic rtl = Net\_PadEdit1.SetValue(pad)

C++ HRESULT pNet\_PadEdit -> SetValue(  
INet\_Pad \*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Net_Pad	[in]	网信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

##### GetValue 方法

描述 获取当前网信息参数开关板值。

Visual Basic rtl = Net\_PadEdit1.GetValue(pad)

C++ HRESULT pNet\_PadEdit -> GetValue (  
INet\_Pad \*\*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Net_Pad	[in]	网信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

4.7、Nod\_Center\_InfoEdit 控件

描述 Nod\_Center\_InfoEdit 控件用于可视化编辑 Nod\_Center\_Info 结点中心信息。

4.7.1、控件界面：



图 4-7 Nod\_Center\_InfoEdit 控件界面

4.7.2、属性：

属性	描述
averimped	平均阻抗
Double	
可读写	
capacity	容量
Double	
可读写	
delay	延迟
Double	
可读写	
limit	限度
Double	
可读写	
maximped	最大阻抗
Double	
可读写	

perccap	容量分配百分比
Double	
可读写	
fclr	辅助颜色
Long	
可读写	
fltp	辅助线型
Integer	
可读写	
layer	图层
Integer	
可读写	
lclass	线种类
Integer	
可读写	
lclr	线颜色
Long	
可读写	
ltp	线型
Integer	
可读写	
lw	线宽
Double	
可读写	
ovprnt	透明输出
Integer	
可读写	
xscale	X 系数
Double	
可读写	
yscale	Y 系数
Double	
可读写	

### 4.7.3、方法：

#### SetValue 方法

描述          设置当前结点中心信息参数值。

Visual Basic    rtl = Nod\_Center\_InfoEdit1.SetValue(info)  
 C++            HRESULT pNod\_Center\_InfoEdit -> SetValue(  
                  INode\_Center\_Info \*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Nod_Center_Info	[in]	结点中心信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### GetValue 方法

描述            获取当前结点中心信息参数值。

Visual Basic    rtl = Nod\_Center\_InfoEdit1.GetValue(info)  
 C++            HRESULT pNod\_Center\_InfoEdit -> GetValue (  
                  INode\_Center\_Info \*\*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Nod_Center_Info	[in]	结点中心信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## 4.8、Pnt\_Info\_ArcEdit 控件

描述            Pnt\_Info\_ArcEdit 控件用于可视化编辑弧类型点信息。

### 4.8.1、控件界面：如图 4-8 所示

### 4.8.2、属性：

属性	描述
iclr	弧颜色
Long 可读写	
layer	图层
Integer 可读写	
ovprnt	透明输出
Integer 可读写	

begang	起始角度
Double	
可读写	
endang	终止角度
Double	
可读写	
penw	笔宽
Double	
可读写	
radiu	弧半径
Double	
可读写	

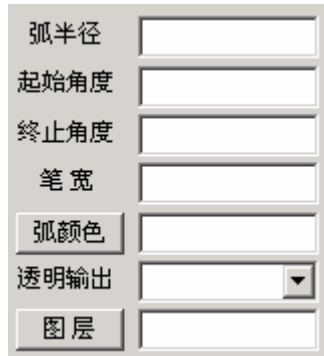


图 4-8 Pnt\_Info\_ArcEdit 控件界面

### 4.8.3、方法：

#### SetValue 方法

描述 设置当前点信息参数值。

Visual Basic rtl = Pnt\_Info\_ArcEdit1.SetValue(info)

C++ HRESULT pPnt\_Info\_ArcEdit -> SetValue(  
IPnt\_Info \*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Pnt_Info	[in]	点信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### GetValue 方法

描述 获取当前点信息参数值。

Visual Basic rtl = Pnt\_Info\_ArcEdit1.GetValue(info)

```
C++      HRESULT pPnt_Info_ArcEdit -> GetValue (  
        IPnt_Info **info, VARIANT_BOOL *rtl);
```

参数说明：

参数	类型	出口/入口	描述
info	Pnt_Info	[in]	点信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

4.9、Pnt\_Info\_CirEdit 控件

描述 Pnt\_Info\_CirEdit 控件用于可视化编辑圆类型点信息。

4.9.1、控件界面：

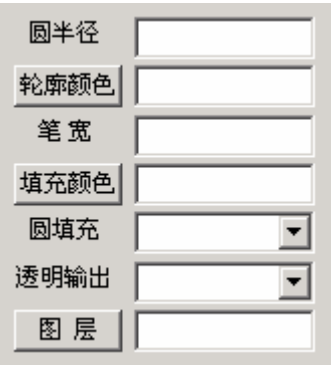


图 4-9 Pnt\_Info\_CirEdit 控件界面

4.9.2、属性：

属性	描述
layer	图层
Integer 可读写	
ovprnt	透明输出
Integer 可读写	
fflag	填充圆
Integer 可读写	

lcol	填充颜色
Long	
可读写	
penw	笔宽
Double	
可读写	
radiu	圆半径
Double	
可读写	
iclr	轮廓颜色
Long	
可读写	

### 4.9.3、方法：

#### SetValue 方法

描述 设置当前点信息参数值。

Visual Basic rtl = Pnt\_Info\_CirEdit1.SetValue(info)

C++ HRESULT pPnt\_Info\_CirEdit -> SetValue(  
IPnt\_Info \*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Pnt_Info	[in]	点信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### GetValue 方法

描述 获取当前点信息参数值。

Visual Basic rtl = Pnt\_Info\_CirEdit1.GetValue(info)

C++ HRESULT pPnt\_Info\_CirEdit -> GetValue (  
IPnt\_Info \*\*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Pnt_Info	[in]	点信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE



4.10、Pnt\_Info\_ImageEdit 控件

描述 Pnt\_Info\_ImageEdit 控件用于可视化编辑图象类型点信息。

4.10.1、控件界面：

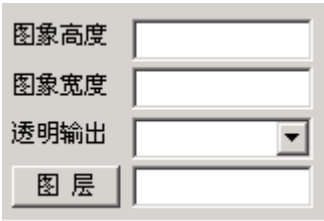


图 4-10 Pnt\_Info\_ImageEdit 控件界面

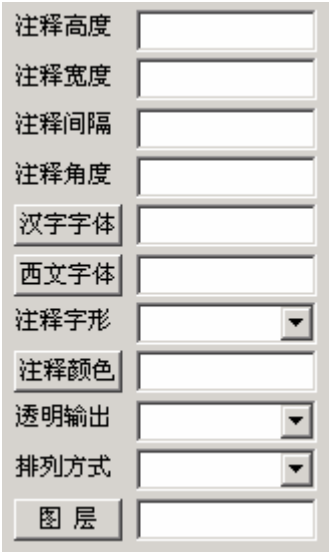


图 4-11 Pnt\_Info\_NoteEdit 控件界面

4.10.2、属性：

属性	描述
layer	图层
Integer 可读写	
ovprnt	透明输出
Integer 可读写	
height	图象高度
Double 可读写	
width	图象宽度
Double 可读写	

### 4.10.3、方法：

#### SetValue 方法

描述 设置当前点信息参数值。

Visual Basic rtl = Pnt\_Info\_ImageEdit1.SetValue(info)

C++ HRESULT pPnt\_Info\_ImageEdit -> SetValue(  
IPnt\_Info \*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Pnt_Info	[in]	点信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### GetValue 方法

描述 获取当前点信息参数值。

Visual Basic rtl = Pnt\_Info\_ImageEdit1.GetValue(info)

C++ HRESULT pPnt\_Info\_ImageEdit -> GetValue (  
IPnt\_Info \*\*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Pnt_Info	[in]	点信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## 4.11、Pnt\_Info\_NoteEdit 控件

描述 Pnt\_Info\_NoteEdit 控件用于可视化编辑注释类型点信息。

### 4.11.1、控件界面：如图 4-11 所示

### 4.11.2、属性：

属性	描述
layer	图层
Integer 可读写	
ovprnt	透明输出
Integer 可读写	

iclr	点颜色
Long	
可读写	
angle	注释角度
Double	
可读写	
chnt	汉字字体
Integer	
可读写	
height	注释高度
Double	
可读写	
hvpl	排列方式
Integer	
可读写	
ifnt	西文字体
Integer	
可读写	
ifnx	汉字字形
Integer	
可读写	
space	注释间隔
Double	
可读写	
width	注释宽度
Double	
可读写	

### 4.11.3、方法：

#### SetValue 方法

描述 设置当前点信息参数值。

Visual Basic rtl = Pnt\_Info\_NoteEdit1.SetValue(info)

C++ HRESULT pPnt\_Info\_NoteEdit -> SetValue(  
IPnt\_Info \*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Pnt_Info	[in]	点信息

rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

GetValue 方法

描述 获取当前点信息参数值。

Visual Basic rtl = Pnt\_Info\_NoteEdit1.GetValue(info)

C++ HRESULT pPnt\_Info\_NoteEdit -> GetValue ( IPnt\_Info \*\*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Pnt_Info	[in]	点信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

4.12、Pnt\_Info\_SubEdit 控件

描述 Pnt\_Info\_SubEdit 控件用于可视化编辑子图类型点信息。

4.12.1、控件界面：

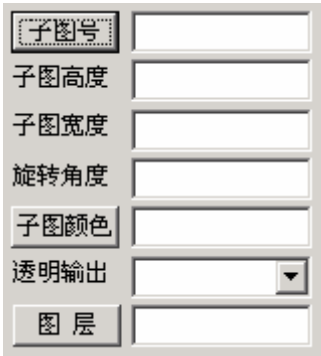


图 4-12 Pnt\_Info\_SubEdit 控件界面

4.12.2、属性：

属性	描述
layer	图层
Integer 可读写	
ovprnt	透明输出
Integer 可读写	

iclr	点颜色
Long	
可读写	
angle	旋转角度
Double	
可读写	
height	子图高度
Double	
可读写	
subno	子图号
Long	
可读写	
width	子图宽度
Double	
可读写	

#### 4.12.3、方法：

##### SetValue 方法

描述 设置当前点信息参数值。

Visual Basic rtl = Pnt\_Info\_SubEdit1.SetValue(info)

C++ HRESULT pPnt\_Info\_SubEdit -> SetValue(  
IPnt\_Info \*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Pnt_Info	[in]	点信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

##### GetValue 方法

描述 获取当前点信息参数值。

Visual Basic rtl = Pnt\_Info\_SubEdit1.GetValue(info)

C++ HRESULT pPnt\_Info\_SubEdit -> GetValue (  
IPnt\_Info \*\*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Pnt_Info	[in]	点信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### 4.13、Pnt\_Info\_TextEdit 控件

描述 Pnt\_Info\_TextEdit 控件用于可视化编辑版图类型点信息。

#### 4.13.1、控件界面：

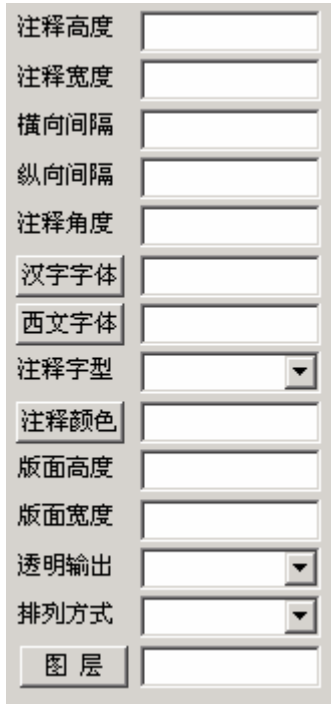


图 4-13 Pnt\_Info\_TextEdit 控件界面

#### 4.13.2、属性：

属性	描述
layer	图层
Integer 可读写	
ovprnt	透明输出
Integer 可读写	
iclr	点颜色
Long 可读写	

angle	版面角度
Double	
可读写	
chnt	汉字字体
Integer	
可读写	
dx	版面高度
Double	
可读写	
dy	版面宽度
Double	
可读写	
height	注释高度
Double	
可读写	
hvpl	排列方式
Integer	
可读写	
ifnt	西文字体
Integer	
可读写	
ifnx	汉字字形
Integer	
可读写	
lspace	纵向间隔
Double	
可读写	
space	横向间隔
Double	
可读写	
width	注释宽度
Double	
可读写	

### 4.13.3、方法：

#### SetValue 方法

描述          设置当前点信息参数值。

Visual Basic    rtl = Pnt\_Info\_TextEdit1.SetValue(info)  
 C++            HRESULT pPnt\_Info\_TextEdit -> SetValue(  
                  IPnt\_Info \*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Pnt_Info	[in]	点信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### GetValue 方法

描述            获取当前点信息参数值。

Visual Basic    rtl = Pnt\_Info\_TextEdit1.GetValue(info)  
 C++            HRESULT pPnt\_Info\_TextEdit -> GetValue (  
                  IPnt\_Info \*\*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Pnt_Info	[in]	点信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### 4.14、Pnt\_Pad\_ArcEdit 控件

描述            Pnt\_Pad\_ArcEdit 控件用于可视化编辑弧类型点信息开关板。

#### 4.14.1、控件界面：如图 4-14 所示

#### 4.14.2、属性：

属性	描述
iclr Enum_Switch_Type 可读写	弧颜色
layer Enum_Switch_Type 可读写	图层
ovprnt Enum_Switch_Type 可读写	透明输出



begang	起始角度
Enum_Switch_Type	
可读写	
endang	终止角度
Enum_Switch_Type	
可读写	
penw	笔宽
Enum_Switch_Type	
可读写	
radiu	弧半径
Enum_Switch_Type	
可读写	

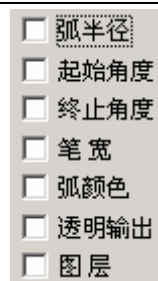


图 4-14 Pnt\_Pad\_ArcEdit 控件界面

### 4.14.3、方法：

#### SetValue 方法

描述 设置当前点信息开关板参数值。

Visual Basic rtl = Pnt\_Pad\_ArcEdit1.SetValue(pad)

C++ HRESULT pPnt\_Pad\_ArcEdit -> SetValue(  
IPnt\_pad \*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### GetValue 方法

描述 获取当前点信息开关板参数值。

Visual Basic rtl = Pnt\_Pad\_ArcEdit1.GetValue(pad)

C++ HRESULT pPnt\_Pad\_ArcEdit -> GetValue (  
IPnt\_Pad \*\*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

4.15、Pnt\_Pad\_CirEdit 控件

描述 Pnt\_Info\_CirEdit 控件用于可视化编辑圆类型点信息开关板。

4.15.1、控件界面：

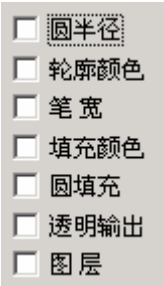


图 4-15 Pnt\_Info\_CirEdit 控件界面

4.15.2、属性：

属性	描述
layer	图层
Enum_Switch_Type 可读写	
ovprnt	透明输出
Enum_Switch_Type 可读写	
fflag	填充圆
Enum_Switch_Type 可读写	
lcol	填充颜色
Enum_Switch_Type 可读写	
penw	笔宽
Enum_Switch_Type 可读写	

radiu	圆半径
Enum_Switch_Type	
可读写	
iclr	轮廓颜色
Enum_Switch_Type	
可读写	

#### 4.15.2、方法：

##### SetValue 方法

描述 设置当前点信息开关板参数值。

Visual Basic rtl = Pnt\_Pad\_CirEdit1.SetValue(pad)

C++ HRESULT pPnt\_Pad\_CirEdit -> SetValue(  
IPnt\_Pad \*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

##### GetValue 方法

描述 获取当前点信息开关板参数值。

Visual Basic rtl = Pnt\_Pad\_CirEdit1.GetValue(pad)

C++ HRESULT pPnt\_Pad\_CirEdit -> GetValue (  
IPnt\_Pad \*\*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### 4.16、Pnt\_Pad\_ImageEdit 控件

描述 Pnt\_Pad\_ImageEdit 控件用于可视化编辑图象类型点信息开关板。

##### 4.16.1、控件界面：

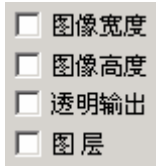


图 4-16 Pnt\_Pad\_ImageEdit 控件界面

4.16.2、属性：

属性	描述
layer	图层
Enum_Switch_Type 可读写	
ovprnt	透明输出
Enum_Switch_Type 可读写	
height	图象高度
Enum_Switch_Type 可读写	
width	图象宽度
Enum_Switch_Type 可读写	

4.16.3、方法：

SetValue 方法

描述 设置当前点信息开关板参数值。

Visual Basic rtl = Pnt\_Pad\_ImageEdit1.SetValue(pad)

C++ HRESULT pPnt\_Pad\_ImageEdit -> SetValue(  
IPnt\_Pad \*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

GetValue 方法

描述 获取当前点信息开关板参数值。

Visual Basic rtl = Pnt\_Pad\_ImageEdit1.GetValue(pad)

C++ HRESULT pPnt\_Pad\_ImageEdit -> GetValue (

---

```
IPnt_Pad **pad, VARIANT_BOOL *rtl);
```

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

---

## 4.17、Pnt\_Pad\_NoteEdit 控件

**描述** Pnt\_Pad\_NoteEdit 控件用于可视化编辑注释类型点信息开关板。

### 4.17.1、控件界面：

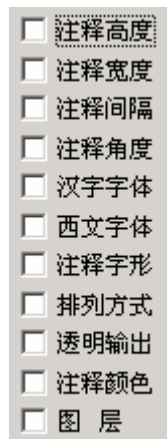


图 4-17 Pnt\_Pad\_NoteEdit 控件界面

### 4.17.2、属性：

属性	描述
layer	图层
Enum_Switch_Type 可读写	
ovprnt	透明输出
Enum_Switch_Type 可读写	
iclr	点颜色
Enum_Switch_Type 可读写	

angle	注释角度
Enum_Switch_Type	
可读写	
chnt	汉字字体
Enum_Switch_Type	
可读写	
height	注释高度
Enum_Switch_Type	
可读写	
hvpl	排列方式
Enum_Switch_Type	
可读写	
ifnt	西文字体
Enum_Switch_Type	
可读写	
ifnx	汉字字形
Enum_Switch_Type	
可读写	
space	注释间隔
Enum_Switch_Type	
可读写	
width	注释宽度
Enum_Switch_Type	
可读写	

### 4.17.3、方法：

#### SetValue 方法

描述 设置当前点信息开关板参数值。

Visual Basic rtl = Pnt\_Pad\_NoteEdit1.SetValue(pad)

C++ HRESULT pPnt\_Pad\_NoteEdit -> SetValue(  
IPnt\_Pad \*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

GetValue 方法

描述 获取当前点信息开关板参数值。

Visual Basic rtl = Pnt\_Pad\_NoteEdit1.GetValue(pad)

C++ HRESULT pPnt\_Pad\_NoteEdit -> GetValue (  
IPnt\_Pad \*\*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

4.18、Pnt\_Pad\_SubEdit 控件

描述 Pnt\_Pad\_SubEdit 控件用于可视化编辑子图类型点信息开关板。

4.18.1、控件界面：

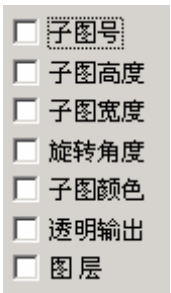


图 4-18 Pnt\_Pad\_SubEdit 控件界面

4.18.2、属性：

属性	描述
layer	图层
Enum_Switch_Type 可读写	
ovprnt	透明输出
Enum_Switch_Type 可读写	
iclr	点颜色
Enum_Switch_Type 可读写	

angle	旋转角度
Enum_Switch_Type	
可读写	
height	子图高度
Enum_Switch_Type	
可读写	
subno	子图号
Enum_Switch_Type	
可读写	
width	子图宽度
Enum_Switch_Type	
可读写	

#### 4.18.3、方法：

##### SetValue 方法

描述 设置当前点信息开关板参数值。

Visual Basic rtl = Pnt\_Pad\_SubEdit1.SetValue(pad)

C++ HRESULT pPnt\_Pad\_SubEdit -> SetValue(  
IPnt\_Pad \*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

##### GetValue 方法

描述 获取当前点信息开关板参数值。

Visual Basic rtl = Pnt\_Pad\_SubEdit1.GetValue(pad)

C++ HRESULT pPnt\_Pad\_SubEdit -> GetValue (  
IPnt\_Pad \*\*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE



4.19、Pnt\_Pad\_TextEdit 控件

描述 Pnt\_Pad\_TextEdit 控件用于可视化编辑版图类型点信息开关板。

4.19.1、控件界面：

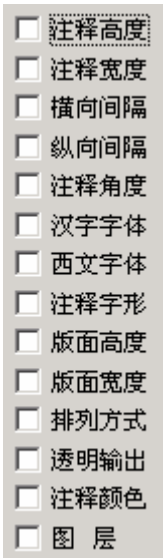


图 4-19 Pnt\_Pad\_TextEdit 控件界面

4.19.2、属性：

属性	描述
layer	图层
Enum_Switch_Type	
可读写	
ovprnt	透明输出
Enum_Switch_Type	
可读写	
iclr	点颜色
Enum_Switch_Type	
可读写	
angle	版面角度
Enum_Switch_Type	
可读写	

chnt	汉字字体
Enum_Switch_Type	
可读写	
dx	版面高度
Enum_Switch_Type	
可读写	
dy	版面宽度
Enum_Switch_Type	
可读写	
height	注释高度
Enum_Switch_Type	
可读写	
hvpl	排列方式
Enum_Switch_Type	
可读写	
ifnt	西文字体
Enum_Switch_Type	
可读写	
ifnx	汉字字形
Enum_Switch_Type	
可读写	
lspace	纵向间隔
Enum_Switch_Type	
可读写	
space	横向间隔
Enum_Switch_Type	
可读写	
width	注释宽度
Enum_Switch_Type	
可读写	

### 4.19.3、方法：

#### SetValue 方法

描述	设置当前点信息开关板参数值。
Visual Basic	rtl = Pnt_Pad_TextEdit1.SetValue(pad)
C++	HRESULT pPnt_Pad_TextEdit -> SetValue(

IPnt\_Pad \*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### GetValue 方法

描述 获取当前点信息开关板参数值。

Visual Basic rtl = Pnt\_Pad\_TextEdit1.GetValue(pad)

C++ HRESULT pPnt\_Pad\_TextEdit -> GetValue (  
IPnt\_Pad \*\*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Pnt_Pad	[in]	点信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## 4.20、Reg\_InfoEdit 控件

描述 Reg\_InfoEdit 控件用于可视化编辑 Reg\_Info 区信息。

### 4.20.1、控件界面：如图 4-20 所示

### 4.20.2、属性：

属性	描述
clr	区域填充色
Long 可读写	
fmode	覆盖模式
Integer 可读写	
layer	图层
Integer 可读写	
patclr	图案颜色
Long 可读写	

pathei	图案高度
Double	
可读写	
patno	填充图案号
Integer	
可读写	
patwid	图案宽度
patwid	
可读写	

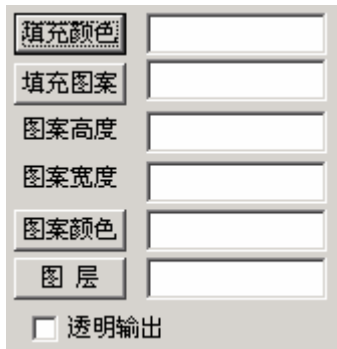


图 4-20 Reg\_InfoEdit 控件界面

4.20.3、方法：

SetValue 方法

描述 设置当前区信息参数值。

Visual Basic rtl = Reg\_InfoEdit1.SetValue(info)

C++ HRESULT pReg\_InfoEdit -> SetValue(  
IReg\_Info \*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Reg_Info	[in]	区信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

GetValue 方法

描述 获取当前区信息参数值。

Visual Basic rtl = Reg\_InfoEdit1.GetValue(info)

C++ HRESULT pReg\_InfoEdit -> GetValue (  
IReg\_Info \*\*info, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
info	Reg_Info	[in]	区信息
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## 4.21、Reg\_PadEdit 控件

**描述** Reg\_PadEdit 控件用于可视化编辑 Reg\_Pad 区信息开关板。

### 4.21.1、控件界面：

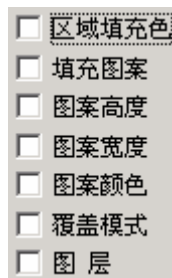


图 4-21 Reg\_PadEdit 控件界面

### 4.21.2、属性：

属性	描述
clr	区域填充色
Enum_Switch_Type 可读写	
fmode	覆盖模式
Enum_Switch_Type 可读写	
layer	图层
Enum_Switch_Type 可读写	
patclr	图案颜色
Enum_Switch_Type 可读写	
pathei	图案高度
Enum_Switch_Type 可读写	

patno	填充图案号
Enum_Switch_Type	
可读写	
patwid	图案宽度
Enum_Switch_Type	
可读写	

### 4.21.3、方法：

#### SetValue 方法

描述 设置当前区信息参数开关板值。

Visual Basic rtl = Reg\_PadEdit1.SetValue(pad)

C++ HRESULT pReg\_PadEdit -> SetValue(  
IReg\_Pad \*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Reg_Pad	[in]	区信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### GetValue 方法

描述 获取当前区信息参数开关板值。

Visual Basic rtl = Reg\_PadEdit1.GetValue(pad)

C++ HRESULT pReg\_PadEdit -> GetValue (  
IReg\_Pad \*\*pad, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pad	Reg_Pad	[in]	区信息开关板
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## 第五章 显示对象说明

显示对象的使用参考 D:\MAPGIS61\VBDemo\vbW60Area 及 D:\MAPGIS61\VBDemo\Controls 目录下的例子。

### 5.1、MapGisDC 对象

**描述** MapGisDC 对象封装了 MapGis 基本的显示操作。

#### 5.1.1、属性：

属性	描述
hMyDC	返回 MapGisDC 的句柄
Long	
只读	
OriginalDisp	还原显示标记 ( 0/1 非还原显示/还原显示 )
Integer	
可读写	

#### 5.1.2、方法：

##### Attach 方法

**描述** 绑定 MyDC 句柄。  
**Visual Basic** MapGisDC.Attach mdc  
**C++** HRESULT pMapGisDC->Attach(long mdc);  
**参数说明：**

参数	类型	出口/入口	描述
mdc	Long	[in]	MyDC 句柄

##### CreateCurvePSDC 方法

**描述** 创建文字变曲线 PS 设备场景。  
**Visual Basic** MapGisDC.CreateCurvePSDC fname  
**C++** HRESULT pMapGisDC->CreateCurvePSDC( BSTR fname );  
**参数说明：**

参数	类型	出口/入口	描述
fname	String	[in]	文件名

**CreateEPSDC 方法**

描述 创建 EPS 设备场景。

Visual Basic MapGisDC.CreateEPSDC fname

C++ HRESULT pMapGisDC->CreateEPSDC(BSTR fname );

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名

**CreateFZPSDC 方法**

描述 创建方正不分色 PS 设备场景。

Visual Basic MapGisDC.CreateFZPSDC fname

C++ HRESULT pMapGisDC->CreateFZPSDC( BSTR fname )

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名

**CreateFZSepPSDC 方法**

描述 创建方正分色 PS 设备场景。

Visual Basic MapGisDC.CreateFZSepPSDC fname

C++ HRESULT pMapGisDC->CreateFZSepPSDC( BSTR fname );

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名

**CreateGIFFileDC 方法**

描述 创建 GIF 图象文件设备场景。

Visual Basic MapGisDC.CreateGIFFileDC fname, wid, hei

C++ HRESULT pMapGisDC->CreateGIFFileDC(  
BSTR fname, short wid, short hei);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名
wid	Integer	[in]	图象文件的宽 ( pixel )
hei	Integer	[in]	图象文件的高 ( pixel )

**CreateMyDCDir 方法**

描述 直接创建 Windows 窗口设备场景。



Visual Basic MapGisDC.CreateMyDCDir hWnd, hdc  
 C++ HRESULT pMapGisDC->CreateMyDCDir( long hWnd, long hdc );

参数说明：

参数	类型	出口/入口	描述
hWnd	Long	[in]	显示对象句柄
hdc	Long	[in]	设备句柄

#### CreateRastDC 方法

描述 创建光栅设备场景。

Visual Basic MapGisDC.CreateRastDC fname  
 C++ HRESULT pMapGisDC->CreateRastDC( BSTR fname );

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名

#### CreateSepEPSDC 方法

描述 创建通用分色 EPS 设备场景。

Visual Basic MapGisDC.CreateSepEPSDC fname  
 C++ HRESULT pMapGisDC->CreateSepEPSDC( BSTR fname );

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名

#### CreateSepPSDC 方法

描述 创建通用分色 PS 设备场景。

Visual Basic MapGisDC.CreateSepPSDC fname  
 C++ HRESULT pMapGisDC->CreateSepPSDC( BSTR fname );

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名

#### CreateSepTextEPSDC 方法

描述 创建分色（文字编号）的 EPS 设备场景。

Visual Basic MapGisDC.CreateSepTextEPSDC fname  
 C++ HRESULT pMapGisDC->CreateSepTextEPSDC( BSTR fname );

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

fname	String	[in]	文件名
-------	--------	------	-----

**CreateTextEPSDC 方法**

描述 创建不分色（文字编号）的 EPS 设备场景。

Visual Basic MapGisDC.CreateTextEPSDC fname

C++ HRESULT pMapGisDC->CreateTextEPSDC( BSTR fname );

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名

**CreateTextPSDC 方法**

描述 创建不分色（文字编号）的 PS 设备场景。

Visual Basic MapGisDC.CreateTextPSDC fname

C++ HRESULT pMapGisDC->CreateTextPSDC( BSTR fname );

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名

**CreateTextSepPSDC 方法**

描述 创建通用不分色（文字编号）的 PS 设备场景。

Visual Basic MapGisDC.CreateTextSepPSDC fname

C++ HRESULT pMapGisDC->CreateTextSepPSDC( BSTR fname );

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名

**CreateWBMPFileDC 方法**

描述 创建 WAP 图象（支持手机图片格式）的设备场景。

Visual Basic MapGisDC.CreateWBMPFileDC fname, wid, hei

C++ HRESULT pMapGisDC->CreateWBMPFileDC(  
BSTR fname,short wid,short hei );

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名
wid	Integer	[in]	图象文件的宽（pixel）
hei	Integer	[in]	图象文件的高（pixel）

**CreateWndDC 方法**

描述 创建 Windows 窗口的设备场景。

Visual Basic MapGisDC.CreateWndDC hWnd

C++ HRESULT pMapGisDC->CreateWndDC( long hWnd );

参数说明：

参数	类型	出口/入口	描述
hWnd	Long	[in]	显示对象句柄

**CreatePrintDC 方法**

描述 创建打印设备场景。

Visual Basic MapGisDC.CreatePrintDC

C++ HRESULT pMapGisDC->CreatePrintDC ();

参数说明： 无

**DispPntArea 方法**

描述 显示点工作区。

Visual Basic sflg = MapGisDC.DispPntArea(area, [layon])

C++ HRESULT pMapGisDC->DispPntArea(  
IWorkArea \*area, short layon=0, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
area	WorkArea	[in]	点工作区
layon	Integer	[in]	图层开关标志 可选；缺省：0
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**DispLinArea 方法**

描述 显示线工作区。

Visual Basic sflg = MapGisDC.DispLinArea(area, [layon])

C++ HRESULT pMapGisDC->DispLinArea(  
IWorkArea \*area, short layon, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
area	WorkArea	[in]	线工作区
layon	Integer	[in]	图层开关标志 可选；缺省：0
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**DispNetArea 方法**

描述 显示网工作区。

Visual Basic sflg = MapGisDC.DispNetArea(area, [layon])

C++ HRESULT pMapGisDC->DispNetArea(  
IWorkArea \*area, short layon=0, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
area	WorkArea	[in]	网工作区
layon	Integer	[in]	图层开关标志 可选；缺省：0
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**DispRegArea 方法**

描述 显示区工作区。

Visual Basic sflg = MapGisDC.DispRegArea(area, [layon])

C++ HRESULT pMapGisDC->DispRegArea(  
IWorkArea \*area, short layon=0, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
area	WorkArea	[in]	区工作区
layon	Integer	[in]	图层开关标志 可选；缺省：0
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**DispArea 方法**

描述 显示点、线、面、网工作区。

Visual Basic sflg = MapGisDC.DispArea(area, [layon])

C++ HRESULT pMapGisDC->DispArea(  
IWorkArea \*area, short layon=0, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
area	WorkArea	[in]	工作区
layon	Integer	[in]	图层开关标志 可选；缺省：0
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**DispElmi 方法**

描述 显示图元。

Visual Basic sflg = MapGisDC.DispElmi(area, elmi, [col])

C++ HRESULT pMapGisDC->DispElmi(  
IWorkArea \*area, long elmi, long col=0, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
area	WorkArea	[in]	工作区
elmi	Long	[in]	图元编号
col	Long	[in]	显示颜色
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**DispImage 方法**

描述 显示图象。

Visual Basic sflg = MapGisDC.DispImage(ImageName, dot, pImageInf)

C++ HRESULT pMapGisDC->DispImage(  
BSTR ImageName, ID\_Dot \*dot, IPnt\_Info \*pImageInf, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
ImageName	String	[in]	图象文件名
dot	D_Dot	[in]	图象显示的设备坐标对象
pImageInf	Pnt_Info	[in]	图象显示参数结构对象
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**DispLinCl 方法**

描述 显示线。

Visual Basic sflg = MapGisDC.DispLinCl(pdotset, [col])

C++ HRESULT pMapGisDC->DispLinCl(  
ID\_DotSet \*pdotset, long col=0, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
pdotset	D_DotSet	[in]	线坐标数组，为图形坐标
col	Long	[in]	显示颜色 可选；缺省：0
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**DispLinElmi 方法**

描述 显示线图元。

Visual Basic sflg = MapGisDC.DispLinElmi(area, elmi, [col])

C++ HRESULT pMapGisDC->DispLinElmi(  
IWorkArea \*area, long elmi, long col=0, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
area	WorkArea	[in]	线工作区
elmi	Long	[in]	图元编号
col	Long	[in]	显示颜色 可选；缺省：0
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**DispLinIn 方法**

描述 显示线。

Visual Basic sflg = MapGisDC.DispLinIn(pdset, plinf, [col])

C++ HRESULT pMapGisDC->DispLinIn(  
ID\_DotSet \*pdset, ILine\_Info \*plinf, long col=0, short \*sflg );

参数说明：

参数	类型	出口/入口	描述
pdset	D_DotSet	[in]	线坐标数组，为图形坐标
plinf	Line_Info	[in]	线图形参数结构地址
col	Long	[in]	显示颜色 可选；缺省：0
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**DispPntElmi 方法**

描述 显示点图元。

Visual Basic sflg = MapGisDC.DispPntElmi(area, elmi, [col])

C++ HRESULT pMapGisDC->DispPntElmi(  
IWorkArea \*area, long elmi, long col=0, short \*sflg );

参数说明：

参数	类型	出口/入口	描述
area	WorkArea	[in]	点工作区
elmi	Long	[in]	图元编号
col	Long	[in]	显示颜色 可选；缺省：0

sflg	Integer	[out,retval]	返回>0/<=0：成功/失败
------	---------	--------------	----------------

**DispPntIn 方法**

描述 显示点。

Visual Basic sflg = MapGisDC.DispPntIn(pdot, dat, ppinf, [col], sflg)

C++ HRESULT pMapGisDC->DispPntIn(  
ID\_Dot \*pdot, BSTR dat, IPnt\_Info \*ppinf, long col=0, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
pdot	D_Dot	[in]	点位置坐标，图形坐标
dat	String	[in]	显示的点数据
ppinf	Pnt_Info	[in]	点图形信息结构地址
col	Long	[in]	显示颜色 可选；缺省：0
sflg	Integer	[out,retval]	返回 1/0：成功/失败

**DispRegCl 方法**

描述 显示区。

Visual Basic sflg = MapGisDC.DispRegCl(area, rdata, [col])

C++ HRESULT pMapGisDC->DispRegCl(  
IWorkArea \*area, VARIANT rdata, long col=0, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
area	WorkArea	[in]	区工作区
rdata	VARIANT	[in]	区域边界弧段数组
col	Long	[in]	显示颜色 可选；缺省：0
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**DispRegElmi 方法**

描述 显示区图元。

Visual Basic sflg = MapGisDC.DispRegElmi(area, elmi, [col])

C++ HRESULT pMapGisDC->DispRegElmi(  
IWorkArea \*area, long elmi, long col, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
area	WorkArea	[in]	区工作区

elmi	Long	[in]	区图元编号
col	Long	[in]	显示颜色
			可选；缺省：0
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**DispRegIn 方法**

描述 显示区。

Visual Basic sflg = MapGisDC.DispRegIn(area, rdata, printf, [col])

C++ HRESULT pMapGisDC->DispRegIn(  
IWorkArea \*area, VARIANT rdata, IReg\_Info \*printf, long col=0,short \*sflg);

参数说明：

参数	类型	出口/入口	描述
area	WorkArea	[in]	区工作区
rdata	VARIANT	[in]	区域边界弧段数组
printf	Reg_Info	[in]	区域图形信息结构变量地址
col	Long	[in]	显示颜色
			可选；缺省：0
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**DispString 方法**

描述 显示串。

Visual Basic sflg = MapGisDC.DispString(str, dot, pNoteInf)

C++ HRESULT pMapGisDC->DispString(  
BSTR str,ID\_Dot \*dot,IPnt\_Info \*pNoteInf, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
str	String	[in]	字符串代码
dot	D_Dot	[in]	字符串坐标数组，为设备坐标
pNoteInf	Pnt_Info	[in]	字符串显示信息结构变量地址
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**DispSubgr 方法**

描述 显示子图。

Visual Basic sflg = MapGisDC.DispSubgr(dot, pSubInf)

C++ HRESULT pMapGisDC->DispSubgr(  
ID\_Dot \*dot, IPnt\_Info \*pSubInf, short \*sflg);

参数说明：



参数	类型	出口/入口	描述
dot	D_Dot	[in]	字符串坐标数组，为设备坐标
pSubInf	Pnt_Info	[in]	字符串显示信息结构变量地址
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**DispText 方法**

描述 显示文本。

Visual Basic sflg = MapGisDC.DispText(str, dot, pTextInf)

C++ HRESULT pMapGisDC->DispText(  
BSTR str,ID\_Dot \*dot,IPnt\_Info \*pTextInf,short \*sflg);

参数说明：

参数	类型	出口/入口	描述
str	String	[in]	字符串代码
dot	D_Dot	[in]	字符串坐标数组，为设备坐标
pTextInf	Pnt_Info	[in]	字符串显示信息结构变量地址
sflg	Integer	[out,retval]	返回>0/<=0：成功/失败

**Arc 方法**

描述 画弧。

Visual Basic MapGisDC.Arc r, ang1, ang2

C++ HRESULT pMapGisDC->Arc(double r, double ang1,double ang2);

参数说明：

参数	类型	出口/入口	描述
r	Double	[in]	弧半径（若 r=0 则该函数把 r 强制为 1）
ang1	Double	[in]	弧线起始角度
ang2	Double	[in]	弧线起始角度

**ArcDev 方法**

描述 画弧。

Visual Basic MapGisDC.ArcDev x, y, r, ang1, ang2, [col]

C++ HRESULT pMapGisDC->ArcDev(  
int x,int y, int r, double ang1, double ang2, long col=0);

参数说明：

参数	类型	出口/入口	描述
x	Long	[in]	弧线所在圆的圆心 x 坐标
y	Long	[in]	弧线所在圆的圆心 y 坐标
r	Long	[in]	弧半径

ang1	Double	[in]	弧线起始角度
ang2	Double	[in]	弧线终止角度
col	Long	[in]	弧线颜色
可选；缺省：0			

**DrawCircle 方法**

描述 画圆。

Visual Basic MapGisDC.DrawCircle rad

C++ HRESULT pMapGisDC->DrawCircle( double rad);

参数说明：

参数	类型	出口/入口	描述
rad	Double	[in]	圆半径（若 r=0 则该函数把 r 强制为 1）

**CircleDev 方法**

描述 画圆。

Visual Basic MapGisDC.CircleDev x, y, r, [col]

C++ HRESULT pMapGisDC->CircleDev(int x,int y,int r,long col=0);

参数说明：

参数	类型	出口/入口	描述
x	Long	[in]	圆心 x 坐标
y	Long	[in]	圆心 y 坐标
r	Long	[in]	圆半径
col	Long	[in]	圆颜色
可选；缺省：0			

**CircleFill 方法**

描述 画填充圆。

Visual Basic MapGisDC.CircleFill rad

C++ HRESULT pMapGisDC->CircleFill(double rad);

参数说明：

参数	类型	出口/入口	描述
rad	Double	[in]	圆半径（若 r=0 则该函数把 r 强制为 1）

**CircleFillDev 方法**

描述 填充圆。

Visual Basic MapGisDC.CircleFillDev x, y, r, [col]

C++ HRESULT pMapGisDC->CircleFillDev(int x, int y, int r, long col=0);

参数说明：

参数	类型	出口/入口	描述
c	Long	[in]	圆心 x 坐标
y	Long	[in]	圆心 y 坐标
r	Long	[in]	圆半径
col	Long	[in]	圆颜色 可选；缺省：0

**DrawLine 方法**

描述 画线到。

Visual Basic MapGisDC.DrawLine pdot1, pdot2

C++ HRESULT pMapGisDC-&gt;DrawLine ( ID\_Dot \*pdot1, ID\_Dot \*pdot2 );

参数说明：

参数	类型	出口/入口	描述
pdot1	D_Dot	[in]	直线起点的设备坐标
pdot2	D_Dot	[in]	直线终点的设备坐标

**LineDev 方法**

描述 画线。

Visual Basic MapGisDC.LineDev x1, y1, x2, y2, [clr]

C++ HRESULT pMapGisDC->LineDev(  
int x1, int y1, int x2, int y2, long clr=0);

参数说明：

参数	类型	出口/入口	描述
x1	Long	[in]	直线起点的设备坐标 x
y1	Long	[in]	直线起点的设备坐标 y
x2	Long	[in]	直线终点的设备坐标 x
y2	Long	[in]	直线终点的设备坐标 y
clr	Long	[in]	线颜色 可选；缺省：0

**LineTo 方法**

描述 画线到。

Visual Basic MapGisDC.LineTo pdot

C++ HRESULT pMapGisDC-&gt;LineTo( ID\_Dot \*pdot)

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

pdot	D_Dot	[in]	目的地点的设备坐标
------	-------	------	-----------

**LineToXY 方法**

描述 画线到。

Visual Basic MapGisDC.LineToXY x, y

C++ HRESULT pMapGisDC->LineToXY(double x,double y);

参数说明：

参数	类型	出口/入口	描述
x	Double	[in]	目的地点的设备坐标 x
y	Double	[in]	目的地点的设备坐标 y

**LineXY 方法**

描述 画线到。

Visual Basic MapGisDC.LineXY x1, y1, x2, y2, [col]

C++ HRESULT pMapGisDC->LineXY(  
double x1, double y1, double x2, double y2, long col=0);

参数说明：

参数	类型	出口/入口	描述
x1	Double	[in]	直线起点的设备坐标 x
y1	Double	[in]	直线起点的设备坐标 y
x2	Double	[in]	直线终点的设备坐标 x
y2	Double	[in]	直线终点的设备坐标 y
col	Long	[in]	直线颜色 可选；缺省：0

**MoveTo 方法**

描述 移动鼠标到。

Visual Basic MapGisDC.MoveTo pdot

C++ HRESULT pMapGisDC->MoveTo(ID\_Dot \*pdot);

参数说明：

参数	类型	出口/入口	描述
pdot	D_Dot	[in]	目的地点的设备坐标

**MoveToXY 方法**

描述 移动鼠标到。

Visual Basic MapGisDC.MoveToXY x, y

C++ HRESULT pMapGisDC->MoveToXY(double x,double y);

参数说明：

参数	类型	出口/入口	描述
x	Double	[in]	目的地点的设备坐标 x
y	Double	[in]	目的地点的设备坐标 y

**PointDev 方法**

描述 画点。

Visual Basic MapGisDC.PointDev x, y, clr

C++ HRESULT pMapGisDC-&gt;PointDev(int x,int y,long clr);

参数说明：

参数	类型	出口/入口	描述
x	Long	[in]	点的设备坐标 x
y	Long	[in]	点的设备坐标 y
clr	Long	[in]	点颜色

**Rect 方法**

描述 画矩形框。

Visual Basic MapGisDC.Rect prect, [clr]

C++ HRESULT pMapGisDC-&gt;Rect(ID\_Rect \*prect,long clr=0);

参数说明：

参数	类型	出口/入口	描述
prect	D_Rect	[in]	矩形对角点坐标
clr	Long	[in]	矩形线颜色

**RectDev 方法**

描述 画矩形框。

Visual Basic MapGisDC.RectDev x1, y1, x2, y2, [clr]

C++ HRESULT pMapGisDC-&gt;RectDev(int x1,int y1,int x2,int y2,long clr=0);

参数说明：

参数	类型	出口/入口	描述
x1	Long	[in]	第一对角点的设备坐标 x
y1	Long	[in]	第一对角点的设备坐标 y
x2	Long	[in]	第二对角点的设备坐标 x
y2	Long	[in]	第二对角点的设备坐标 y
clr	Long	[in]	矩形线颜色 可选；缺省：0

**DrawRect 方法**

描述 画矩形框。

Visual Basic MapGisDC.DrawRect pdot1, pdot2

C++ HRESULT pMapGisDC->DrawRect( ID\_Dot \*pdot1,ID\_Dot \*pdot2);

参数说明：

参数	类型	出口/入口	描述
pdot1	D_Dot	[in]	第一对角点的逻辑坐标
pdot2	D_Dot	[in]	第二对角点的逻辑坐标

**RectXY 方法**

描述 画矩形框。

Visual Basic MapGisDC.RectXY x1, y1, x2, y2, [clr]

C++ HRESULT pMapGisDC->RectXY(  
double x1,double y1,double x2, double y2, long clr=0 );

参数说明：

参数	类型	出口/入口	描述
x1	Double	[in]	第一对角点的设备坐标 x
y1	Double	[in]	第一对角点的设备坐标 y
x2	Double	[in]	第二对角点的设备坐标 x
y2	Double	[in]	第二对角点的设备坐标 y
clr	Long	[in]	矩形线颜色 可选；缺省：0

**ColorBarDev 方法**

描述 填充矩形。

Visual Basic MapGisDC.ColorBarDev x, y, dx, dy, col

C++ HRESULT pMapGisDC->ColorBarDev(int x,int y,int dx,int dy,long col);

参数说明：

参数	类型	出口/入口	描述
x	Long	[in]	第一对角点的设备坐标 x 坐标
y	Long	[in]	第一对角点的设备坐标 y 坐标
dx	Long	[in]	第二对角点的设备坐标 x 坐标
dy	Long	[in]	第二对角点的设备坐标 y 坐标
col	Long	[in]	矩形颜色 可选；缺省：0

**PolyGon 方法**

描述 填充多边形（设备坐标）。

Visual Basic MapGisDC.PolyGon pdotset, ne, na

C++ HRESULT pMapGisDC->PolyGon(ID\_DotSet \*pdotset,VARIANT ne,int na);

参数说明：

参数	类型	出口/入口	描述
pdotset	D_DotSet	[in]	区域边界坐标数组，坐标值为设备坐标
ne	VARIANT	[in]	边界坐标点数组
na	Long	[in]	数组项数

**PolyLine 方法**

描述 画折线（设备坐标）。

Visual Basic MapGisDC.PolyLine pdotset, ne, na

C++ HRESULT pMapGisDC->PolyLine(D\_DotSet \*pdotset,VARIANT ne,int na);

参数说明：

参数	类型	出口/入口	描述
pdotset	D_DotSet	[in]	区域边界坐标数组，坐标值为设备坐标
ne	VARIANT	[in]	边界坐标点数组
na	Long	[in]	数组项数

**LpToDp 方法**

描述 逻辑坐标->设备坐标。

Visual Basic MapGisDC.LpToDp(logx, logy, pdevx, pdevy)

C++ HRESULT pMapGisDC->LpToDp(double logx,double logy,VARIANT \*pdevx, VARIANT \*pdevy);

参数说明：

参数	类型	出口/入口	描述
logx	Double	[in]	点逻辑坐标 x
logy	Double	[in]	点逻辑坐标 y
pdevx	VARIANT	[out]	点设备坐标 x
pdevy	VARIANT	[out]	点设备坐标 y

**DpToLp 方法**

描述 设备坐标->逻辑坐标。

Visual Basic MapGisDC.DpToLp(devx, devy, plogx, plogy)

C++ HRESULT pMapGisDC->DpToLp(int devx,int devy,VARIANT \*plogx, VARIANT \*plogy);

参数说明：

参数	类型	出口/入口	描述
devx	Long	[in]	设备坐标 x
devy	Long	[in]	设备坐标 y
plogx	VARIANT	[out]	逻辑坐标 x
plogy	VARIANT	[out]	逻辑坐标 y

### DpToLpXY 方法

描述 设备坐标->逻辑坐标。

Visual Basic MapGisDC.DpToLpXY(plogx, plogy)

C++ HRESULT pMapGisDC->DpToLpXY( VARIANT \*plogx,VARIANT \*plogy);

参数说明：

参数	类型	出口/入口	描述
plogx	VARIANT	[in,out]	逻辑坐标 x
plogy	VARIANT	[in,out]	逻辑坐标 y

### LpToDpInt 方法

描述 逻辑坐标->设备坐标。

Visual Basic MapGisDC.LpToDpInt(x, y, pdevx, pdevy)

C++ HRESULT pMapGisDC->LpToDpInt(  
double x, double y,VARIANT \*pdevx, VARIANT \*pdevy);

参数说明：

参数	类型	出口/入口	描述
x	Double	[in]	逻辑坐标 x
y	Double	[in]	逻辑坐标 y
pdevx	VARIANT	[out]	设备坐标 x
pdevy	VARIANT	[out]	设备坐标 y

### LpToDpXY 方法

描述 逻辑坐标->设备坐标。

Visual Basic MapGisDC.LpToDpXY(pdevdot)

C++ HRESULT pMapGisDC->LpToDpXY(ID\_Dot \*pdevdot);

参数说明：

参数	类型	出口/入口	描述
pdevdot	D_Dot	[in,out]	设备坐标



**CleanBrush 方法**

描述 清除刷子。

Visual Basic MapGisDC.CleanBrush

C++ HRESULT pMapGisDC->CleanBrush();

参数说明： 无

**CleanPen 方法**

描述 清除笔。

Visual Basic MapGisDC.CleanPen

C++ HRESULT pMapGisDC->CleanPen();

参数说明： 无

**CleanPenColor 方法**

描述 清除笔颜色。

Visual Basic MapGisDC.CleanPenColor

C++ HRESULT pMapGisDC->CleanPenColor();

参数说明： 无

**SetCoordPntDisp 方法**

描述 设置坐标点显示标记。

Visual Basic sflg = MapGisDC.SetCoordPntDisp(sCoord)

C++ HRESULT pMapGisDC->SetCoordPntDisp(short sCoord, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
sCoord	Integer	[in]	坐标点显示状态
sflg	Integer	[out,retval]	返回 1/0：不显示/显示

**GetDispParam 方法**

描述 取显示参数。

Visual Basic sflg = MapGisDC.GetDispParam(pfx0, pfy0, pfScalex, [pfScaley], [pfAngle])

C++ HRESULT pMapGisDC->GetDispParam(VARIANT \*pfx0, VARIANT \*pfy0, VARIANT \*pfScalex, VARIANT \*pfScaley, VARIANT \*pfAngle, short \*sflg );

参数说明：

参数	类型	出口/入口	描述
pfx0	VARIANT	[out]	视口原点坐标 x
pfy0	VARIANT	[out]	视口原点坐标 y

pfScalex	VARIANT	[out]	x 方向缩放比例
pfScaley	VARIANT	[out,optional]	y 方向缩放比例
pfAngle	VARIANT	[out,optional]	显示角度（缺省为 NULL）
sflg	Integer	[out, retval]	返回 1/0：成功/失败

**SetDispParam 方法**

描述 设置显示参数。

Visual Basic sflg = MapGisDC.SetDispParam(fx0, fy0, fScalex, [fScaley], [fAngle])

C++ HRESULT pMapGisDC->SetDispParam(  
double fx0,double fy0,double fScalex, double fScaley=0,  
double fAngle=0, short \*sflg );

参数说明：

参数	类型	出口/入口	描述
fx0	Double	[in]	视口原点横坐标
fy0	Double	[in]	视口原点纵坐标
fScalex	Double	[in]	x 方向的缩放比例
fScaley	Double	[in]	y 方向的缩放比例
fAngle	Double	[in]	显示角度（缺省为 0）
sflg	Integer	[out,retval]	返回 1/0：成功/失败

**GetDispRect 方法**

描述 取显示范围。

Visual Basic sflg = MapGisDC.GetDispRect(pRect)

C++ HRESULT pMapGisDC->GetDispRect(ID\_Rect \*pRect ,short \*sflg);

参数说明：

参数	类型	出口/入口	描述
pRect	D_Rect	[in]	显示范围
sflg	Integer	[out,retval]	返回 1/0：成功/失败

**SetDispRect 方法**

描述 设置显示范围。

Visual Basic sflg = MapGisDC.SetDispRect(pRect)

C++ HRESULT pMapGisDC->SetDispRect(ID\_Rect \*pRect ,short \*sflg);

参数说明：

参数	类型	出口/入口	描述
pRect	D_Rect	[in]	矩形对角点坐标
sflg	Integer	[out,retval]	返回 1/0：不显示/显示

**SetMessageDisp 方法**

描述 设置显示信息标记。

Visual Basic sflg = MapGisDC.SetMessageDisp(Dispif)

C++ HRESULT pMapGisDC->SetMessageDisp( short Dispif,short \*sflg );

参数说明：

参数	类型	出口/入口	描述
Dispif	Integer	[in]	错误提示状态
sflg	Integer	[out,retval]	返回 0/1 :不显示错误信息/显示错误信息

**SetBrush 方法**

描述 设置刷子。

Visual Basic MapGisDC.SetBrush bcol, pno, phe, pcol

C++ HRESULT pMapGisDC->SetBrush(  
long bcol,short pno,double phe, long pcol );

参数说明：

参数	类型	出口/入口	描述
bcol	Long	[in]	显示底色
pno	Integer	[in]	图案编号 ( 为 0 则不填充图案 )
phe	Double	[in]	图案高度
pcol	Long	[in]	图案颜色

**SetPen 方法**

描述 设置笔。

Visual Basic MapGisDC.SetPen penw, color

C++ HRESULT pMapGisDC->SetPen( double penw,int color);

参数说明：

参数	类型	出口/入口	描述
penw	Double	[in]	笔宽的型号
color	Long	[in]	笔的颜色

**SetPenColor 方法**

描述 设置笔颜色。

Visual Basic MapGisDC.SetPenColor clr

C++ HRESULT pMapGisDC->SetPenColor( long clr);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

clr	Long	[in]	笔的颜色
-----	------	------	------

**GetPenMode 方法**

描述 取笔模式。

Visual Basic mode = MapGisDC.GetPenMode()

C++ HRESULT pMapGisDC->GetPenMode(int \*mode);

参数说明：

参数	类型	出口/入口	描述
mode	Long	[out,retval]	绘图模式

**SetPenMode 方法**

描述 设置笔模式。

Visual Basic oldmode = MapGisDC.SetPenMode(nMode)

C++ HRESULT pMapGisDC->SetPenMode( int nMode, int \*oldmode);

参数说明：

参数	类型	出口/入口	描述
nMode	Long	[in]	绘图模式
oldmode	Long	[out,retval]	以前的绘图模式

**LockColor 方法**

描述 锁定颜色。

Visual Basic MapGisDC.LockColor clrno

C++ HRESULT pMapGisDC->LockColor(long clrno);

参数说明：

参数	类型	出口/入口	描述
clrno	Long	[in]	被锁定的颜色

**UnLockColor 方法**

描述 解锁颜色。

Visual Basic MapGisDC.UnLockColor

C++ HRESULT pMapGisDC->UnLockColor();

参数说明： 无

**GetLockState 方法**

描述 取锁颜色状态。

Visual Basic lockstate = MapGisDC.GetLockState()

C++ HRESULT pMapGisDC->GetLockState(short \*lockstate);

参数说明：

参数	类型	出口/入口	描述
lockstate	Integer	[out,retval]	返回 1/0：锁定/未锁定

## 第六章 Layer 和 Map 说明

Layer 和 Map 的使用方法，参考 D:\MAPGIS61\VBDemo\Controls 目录下的例子。

### 6.1、MapLayer 对象

描述 对象为所有 Layer 的基类对象，封装了所有 Layer 的公共操作和实现入口

#### 6.1.1、属性：

属性	描述
Describe	Layer 的描述
String	
可读写	
LayerType	取 Layer 的类型
Enum_Area_Type	
只读	
MaxDispRate	最大显示比例
Double	
可读写	
MinDispRate	最小显示比例
Double	
可读写	
ModifyFlag	修改标记
Integer	
可读写	
Rect	范围
D_Rect	
只读	
Tag	附加数据
Long	
可读写	
Visible	是否可见标记
BOOL	
可读写	

---

 WorkArea 工作区接口

WorkArea

只读

### 6.1.2、方法：

#### Attach 方法

描述 将 WorkArea 与 Layer 对象绑定。

Visual Basic sflg = MapLayer.Attach(pIWorkArea)

 C++ HRESULT pMapLayer->Attach(  
IWorkArea \*pIWorkArea, VARIANT\_BOOL \*sflg);

参数说明：

参数	类型	出口/入口	描述
pIWorkArea	WorkArea	[in]	工作区接口
sflg	Boolean	[out,retval]	成功返回 TRUE，失败返回 FALSE

#### Detach 方法

描述 解除 WorkArea 与 Layer 对象的绑定。

Visual Basic sflg = MapLayer.Detach()

C++ HRESULT pMapLayer-&gt;Detach(VARIANT\_BOOL \*sflg);

参数说明：

参数	类型	出口/入口	描述
sflg	Boolean	[out, retval]	成功返回 TRUE，失败返回 FALSE

#### Display 方法

描述 显示 Layer 的图形内容。

Visual Basic MapLayer.Display pMydc

C++ HRESULT pMapLayer-&gt;Display(IMapGisDC \*pMydc);

参数说明：

参数	类型	出口/入口	描述
pMydc	MapGisDC	[in]	显示设备的 MapGisDC 对象

## 6.2、GroupLayer 对象

描述 GroupLayer 对象主要封装了对多个 Layer 的管理功能。

### 6.2.1、属性：

属性	描述
Count	子 Layer 的数目
Long	
只读	
Describe	描述内容
String	
可读写	
LayerType	取文件类型 (gisGROUP)
Enum_Area_Type	
只读	
MaxDispRate	最大显示比例
Double	
可读写	
MinDispRate	最小显示比例
Double	
可读写	
ModifyFlag	修改标记
Integer	
可读写	
Rect	范围
D_Rect	
只读	
Tag	附加数据
Long	
可读写	
Visible	是否可见的标记
BOOL	
可读写	
WorkArea	工作区接口 (始终为空)
WorkArea	
只读	

## 6.2.2、方法：

### Add 方法

描述 添加一个子 Layer。

Visual Basic sflg = GroupLayer.Add(pVal)

C++ HRESULT pGroupLayer-> Add(IMapLayer \*pVal,short \*sflg);



参数说明：

参数	类型	出口/入口	描述
pVal	MapLayer	[in]	要添加的 Layer 对象
sflg	Integer	[out,retval]	成功返回 1，失败返回 0

#### Attach 方法

描述 将 WorkArea 与 Layer 对象绑定（没有作用）。

Visual Basic sflg = GroupLayer.Attach(pIWorkArea)

C++ HRESULT pGroupLayer->Attach(  
IWorkArea\*pIWorkArea,VARIANT\_BOOL \*sflg);

参数说明：

参数	类型	出口/入口	描述
pIWorkArea	WorkArea	[in]	工作区接口
sflg	Boolean	[out,retval]	成功返回 TRUE，失败返回 FALSE

#### Detach 方法

描述 解除 WorkArea 与 Layer 对象的绑定（没有作用）。

Visual Basic sflg = GroupLayer.Detach()

C++ HRESULT pGroupLayer-> Detach(VARIANT\_BOOL \*sflg);

参数说明：

参数	类型	出口/入口	描述
sflg	Boolean	[out, retval]	成功返回 TRUE，失败返回 FALSE

#### Display 方法

描述 显示组 Layer 的内容。

Visual Basic GroupLayer.Display pMydc

C++ HRESULT pGroupLayer-> Display(IMapGisDC \*pMydc);

参数说明：

参数	类型	出口/入口	描述
pMydc	MapGisDC	[in]	显示 GroupLayer 的 MapGisDC 对象

#### FindFirst 方法

描述 查找所有的子 Layer 接口（第一个）（在所有子孙 Layer 中查找）。

Visual Basic Set ppLayer = GroupLayer.FindFirst(pType)

C++ HRESULT pGroupLayer->FindFirst(  
Enum\_FindLayer\_Type pType, IMapLayer \*\*ppLayer);

参数说明：

参数	类型	出口/入口	描述
pType	Enum_FindLayer_Type	[in]	要查询的类型
ppLayer	MapLayer	[out,retval]	第一个找到的 Layer 对象

**FindNext 方法**

描述 查找所有的子 Layer 接口 ( 下一个 ) ( 和 FindFirst 配合使用 )

Visual Basic Set ppLayer = GroupLayer.FindNext()

C++ HRESULT pGroupLayer->FindNext(IMapLayer \*\*ppLayer);

参数说明：

参数	类型	出口/入口	描述
ppLayer	MapLayer	[out,retval]	下一个找到的 Layer 对象

**FindLayerByName 方法**

描述 通过文件名称查找文件类型的子 Layer ( 在所有子孙 Layer 中查找 )

Visual Basic Set ppMapLayer = GroupLayer.FindLayerByName(szFName, [szDSN])

C++ HRESULT pGroupLayer->FindLayerByName (BSTR szFName, BSTR szDSN, IMapLayer \*\*ppMapLayer);

参数说明：

参数	类型	出口/入口	描述
szFName	String	[in]	要查询 Layer 的文件名称
szDSN	String	[in]	要查询 Layer 的数据源名称 可选；缺省值：“ ”。
ppMapLayer	MapLayer	[out,retval]	返回找到的 Layer 对象

**FindGroupLayer 方法**

描述 通过描述称查找组类型的子 Layer ( 在所有子孙 Layer 中查找 )

Visual Basic Set ppGroupLayer = GroupLayer.FindGroupLayer(szDescribe)

C++ HRESULT pGroupLayer->FindGroupLayer(BSTR szDescribe, IGroupLayer\*\* ppGroupLayer);

参数说明：

参数	类型	出口/入口	描述
szDescribe	String	[in]	要查找 GroupLayer 的描述
ppGroupLayer	GroupLayer	[out,retval]	返回要查找的 GroupLayer 对象

**FindIndexInTopLayer 方法**

描述 查找指定的 Layer 在所有子 Layer 中的序号 ( 仅在所有儿子 Layer 中查找 )

Visual Basic pIndex = GroupLayer.FindIndexInTopLayer(pLayer)

C++ HRESULT pGroupLayer-> FindIndexInTopLayer(  
IMapLayer \*pLayer, long \*pIndex);

参数说明：

参数	类型	出口/入口	描述
pLayer	MapLayer	[in]	要查找的 Layer 对象
pIndex	Long	[out,retval]	查找到子 Layer 的序号

### GetSubGroupLayerCount 方法

描述 取所有的子类型为 Group 的 Layer 的个数（在所有子孙 Layer 中查找）。

Visual Basic pVal = GroupLayer.GetSubGroupLayerCount()

C++ HRESULT pGroupLayer-> GetSubGroupLayerCount(long \*pVal);

参数说明：

参数	类型	出口/入口	描述
pVal	Long	[out,retval]	子类型为 Group 的 Layer 的个数

### GetSubLayerCount 方法

描述 取所有的子类型为非 Group 的 Layer 的个数（在所有子孙 Layer 中查找）。

Visual Basic pVal = GroupLayer.GetSubLayerCount()

C++ HRESULT pGroupLayer-> GetSubLayerCount(long \*pVal);

参数说明：

参数	类型	出口/入口	描述
pVal	Long	[out,retval]	所有子类型为非 Group 的 Layer 的个数

### Insert 方法

描述 在指定的位置插入一个子 Layer。

Visual Basic sflg = GroupLayer.Insert(index, pVal)

C++ HRESULT pGroupLayer->Insert(long index, IMapLayer \*pVal, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	指定的位置
pVal	MapLayer	[in]	要插入的 Layer 对象
sflg	Integer	[out,retval]	成功返回 1，失败返回 0

### IsExistedLayer 方法

描述 判断指定的 Layer 是否为 GroupLayer 的子孙 Layer。

Visual Basic bVal = GroupLayer.IsExistedLayer(pLayer)

C++ HRESULT pGroupLayer->IsExistedLayer(

---

```
IMapLayer*pLayer, VARIANT_BOOL *bVal);
```

参数说明：

参数	类型	出口/入口	描述
pLayer	MapLayer	[in]	要判断的 Layer 对象
bVal	Boolean	[out,retval]	为子孙 Layer 返回 TRUE，否则返回 FALSE

---

### Item 方法

描述 根据序号取指定的子 Layer。

Visual Basic Set ppLayer = GroupLayer.Item(index)

C++ HRESULT pGroupLayer->Item(long index, IMapLayer \*\*ppLayer);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	序号
ppLayer	MapLayer	[out,retval]	得到指定的子 Layer 对象

---

### LayersIsInGroupLayer 方法

描述 判断一个 Layer 以及其子孙是否存在于 GroupLayer 的子孙 Layer 中。

Visual Basic bVal = GroupLayer.LayersIsInGroupLayer(pLayer)

C++ HRESULT pGroupLayer->LayersIsInGroupLayer(  
IMapLayer \*pLayer, VARIANT\_BOOL \*bVal);

参数说明：

参数	类型	出口/入口	描述
pLayer	MapLayer	[in]	要判断的 Layer 对象
bVal	Boolean	[out,retval]	存在返回 TRUE，否则返回 FALSE

---

### MoveLayer 方法

描述 移动指定的子 Layer（根据对象）。

Visual Basic sflg = GroupLayer.MoveLayer(pLayer, toIndex)

C++ HRESULT pGroupLayer->MoveLayer(  
IMapLayer\*pLayer, long toIndex, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
pLayer	MapLayer	[in]	要移动的子 Layer 对象
toIndex	Long	[in]	移动的新位置
sflg	Integer	[out,retval]	成功返回 1，失败返回 0

---

**MoveTo 方法**

描述 移动指定的子 Layer ( 根据序号 )。

Visual Basic sflg = GroupLayer.MoveTo(frmIndex, toIndex)

C++ HRESULT pGroupLayer->MoveTo(  
long frmIndex, long toIndex, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
frmIndex	Long	[in]	要移动子 Layer 的序号
toIndex	Long	[in]	移动到的新位置
sflg	Integer	[out,retval]	成功返回 1，失败返回 0

**Remove 方法**

描述 删除指定的子 Layer ( 根据对象 )。

Visual Basic GroupLayer.Remove pVal

C++ HRESULT pGroupLayer-> Remove(IMapLayer \*pVal);

参数说明：

参数	类型	出口/入口	描述
pVal	MapLayer	[in]	要删除的子 Layer 对象

**RemoveAll 方法**

描述 删除所有的子 Layer。

Visual Basic GroupLayer.RemoveAll

C++ HRESULT pGroupLayer-> RemoveAll();

参数说明： 无

**RemoveAt 方法**

描述 删除指定的子 Layer ( 根据序号 )。

Visual Basic GroupLayer.RemoveAt index

C++ HRESULT pGroupLayer-> RemoveAt(long index);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	要删除子 Layer 的序号

**6.3、VectorLayer 对象**

描述 对象为点/线/区/网等矢量 Layer 的基类对象，并提供其矢量 Layer 的实现入口

### 6.3.1、属性：

属性	描述
Describe	Layer 的描述
String	
可读写	
LayerType	取 Layer 的类型
Enum_Area_Type	
只读	
MaxDispRate	最大显示比例
Double	
可读写	
MinDispRate	最小显示比例
Double	
可读写	
ModifyFlag	修改标记
Integer	
可读写	
GroupCode	组代码
Integer	
可读写	
UserType	用户自定义类型
Integer	
可读写	

### 6.3.2、方法：

**Attach 方法** 参见 MapLayer 的 Attach 方法说明

**Detach 方法** 参见 MapLayer 的 Detach 方法说明

**Display 方法** 参见 MapLayer 的 Display 方法说明

## 6.4、LinLayer 对象

**描述** LinLayer 对象封装了线文件的数据管理和显示功能

### 6.4.1、属性：

属性	描述
----	----

Describe	Layer 的描述
String	
可读写	
LayerType	取 Layer 的类型
Enum_Area_Type	
只读	
MaxDispRate	最大显示比例
Double	
可读写	
MinDispRate	最小显示比例
Double	
可读写	
ModifyFlag	修改标记
Integer	
可读写	
groupCode	组代码
Integer	
可读写	
UserType	用户自定义类型
Integer	
可读写	

### 6.4.2、方法：

#### Attach 方法

描述 将 WorkArea 与 LinLayer 对象绑定

Visual Basic sflg = LinLayer.Attach(pIWorkArea)

C++ HRESULT pLinLayer->Attach(  
IWorkArea\*pIWorkArea, VARIANT\_BOOL \*sflg);

参数说明：

参数	类型	出口/入口	描述
pIWorkArea	WorkArea	[in]	工作区接口
sflg	Boolean	[out,retval]	成功返回 TRUE，失败返回 FALSE

#### Detach 方法

描述 解除 WorkArea 与 LinLayer 对象的绑定

Visual Basic sflg = LinLayer.Detach()

C++ HRESULT pLinLayer->Detach(VARIANT\_BOOL \*sflg);

参数说明：

参数	类型	出口/入口	描述
sflg	Boolean	[out, retval]	成功返回 TRUE，失败返回 FALSE

### Display 方法

描述 显示 LinLayer 的图形内容

Visual Basic LinLayer.Display pMydc

C++ HRESULT pLinLayer->Display(IMapGisDC \*pMydc);

参数说明：

参数	类型	出口/入口	描述
pMydc	MapGisDC	[in]	显示设备的 MapGisDC 对象

## 6.5、NetLayer 对象

描述 对象封装了网文件的数据管理和显示功能

6.5.1、属性：参见 LinLayer 的属性说明

6.5.2、方法：参见 LinLayer 的方法说明

## 6.6、PntLayer 对象

描述 对象封装了点文件的数据管理和显示功能

6.6.1、属性：参见 LinLayer 的属性说明

6.6.2、方法：参见 LinLayer 的方法说明

## 6.7、RegLayer 对象

描述 对象封装了区文件的数据管理和显示功能

6.7.1、属性：参见 LinLayer 的属性说明

6.7.2、方法：参见 LinLayer 的方法说明



## 6.8、Map 对象

**描述** Map 对象主要封装了对 MAPGIS 基本图形的编辑/输出操作和对多个 Layer 的管理功能。

### 6.8.1、属性：

属性	描述
ActiveLayer	当前激活的 Layer
MapLayer	
可读写	
Layer	根据序号取 Layer
MapLayer	参数：Long：序号
只读	
MapLayerCount	Layer 的数目
Long	
只读	
MapName	Map 的文件名称
String	
可读写	
MapRect	范围
D_Rect	
只读	
MapSurround	根据序号取一个环境对象
MapSurround	参数：Long：序号
只读	
MapSurroundCount	环境对象的数目
Long	
只读	
ModifyFlag	修改标记
Integer	
可读写	
Legend	包含的图例对象
MapLegend	
只读	
PrintParam	包含的输出参数对象
MapPrintParam	
只读	

CanDispLegend	是否显示图例内容的标记
Integer	
可读写	
CanDispScaleBar	是否显示比例信息的标记
Integer	
可读写	

## 6.8.2、方法：

### AddLayer 方法

描述 添加一个 Layer。

Visual Basic Map.AddLayer pLayer

C++ HRESULT pMap -> AddLayer(IMapLayer \*pLayer);

参数说明：

参数	类型	出口/入口	描述
pLayer	MapLayer	[in]	要添加的 Layer 对象

### AddMapSurround 方法

描述 增加一个环境对象。

Visual Basic Map.AddMapSurround pMapSurround

C++ HRESULT pMap -> AddMapSurround(IMapSurround \*pMapSurround);

参数说明：

参数	类型	出口/入口	描述
pMapSurround	MapSurround	[in]	要增加环境对象

### ClearLayer 方法

描述 清除所有的 Layer。

Visual Basic Map.ClearLayer

C++ HRESULT pMap -> ClearLayer();

参数说明： 无

### ClearMapSurround 方法

描述 清除所有的环境对象。

Visual Basic Map.ClearMapSurround

C++ HRESULT pMap -> ClearMapSurround();

参数说明： 无

**DeleteMapLayer 方法**

描述 删除指定的 Layer。

Visual Basic Map.DeleteMapLayer pLayer

C++ HRESULT pMap -> DeleteMapLayer(IMapLayer \*pLayer);

参数说明：

参数	类型	出口/入口	描述
pLayer	MapLayer	[in]	要删除的 Layer 对象

**DeleteMapSurround 方法**

描述 删除指定的环境对象。

Visual Basic Map.DeleteMapSurround pMapSurround

C++ HRESULT pMap -> DeleteMapSurround(IMapSurround \*pMapSurround);

参数说明：

参数	类型	出口/入口	描述
pMapSurround	MapSurround	[in]	要删除的环境对象

**DisplayMap 方法**

描述 显示 Map 的图形内容。

Visual Basic Map.DisplayMap pMapDC

C++ HRESULT pMap -> DisplayMap(IMapGisDC \*pMapDC);

参数说明：

参数	类型	出口/入口	描述
pMapDC	MapGisDC	[in]	显示设备的 MapGis 对象

**FindFirst 方法**

描述 查找所有的子 MapLayer 接口 ( 第一个 )。

Visual Basic Set pLayer = Map.FindFirst(pType)

C++ HRESULT pMap->FindFirst(  
Enum\_FindLayer\_Type pType, IMapLayer \*\*ppLayer);

参数说明：

参数	类型	出口/入口	描述
pType	Enum_FindLayer_Type	[in]	要查询的类型
ppLayer	MapLayer	[out,retval]	第一个找到的子 Layer 对象

**FindNext 方法**

描述 查找所有的子 MapLayer 接口 ( 下一个 ) ( 和 FindFirst 配合使用 )。

Visual Basic Set ppLayer = Map.FindNext()

C++ HRESULT pMap->FindNext(IMapLayer \*\*ppLayer);

参数说明：

参数	类型	出口/入口	描述
ppLayer	MapLayer	[out,retval]	下一个找到的子 Layer 对象

### FindLayerByName 方法

描述 通过文件名称查找文件类型的子 Layer。

Visual Basic Set ppMapLayer = Map.FindLayerByName(szFName, [szDSN])

C++ HRESULT pMap-> FindLayerByName (  
BSTR szFName, BSTR szDSN, IMapLayer \*\*ppMapLayer);

参数说明：

参数	类型	出口/入口	描述
szFName	String	[in]	要查询 Layer 的文件名称
szDSN	String	[in]	要查询 Layer 的数据源名称 可选；缺省值：“ ”。
ppMapLayer	MapLayer	[out,retval]	返回找到的子 Layer 对象

### FindGroupLayer 方法

描述 检索函数：通过描述称查找组类型的子 Layer。

Visual Basic Set ppGroupLayer = Map.FindGroupLayer(szDescribe)

C++ HRESULT pMap->FindGroupLayer(  
BSTR szDescribe, IGroupLayer\*\* ppGroupLayer);

参数说明：

参数	类型	出口/入口	描述
szDescribe	String	[in]	要查找 GroupLayer 的描述
ppGroupLayer	GroupLayer	[out,retval]	返回要查找的 GroupLayer 对象

### GetPageSize 方法

描述 取页面的高和宽值。

Visual Basic Map.GetPageSize width, height

C++ HRESULT pMap -> GetPageSize(double \*width , double \*height);

参数说明：

参数	类型	出口/入口	描述
width	Double	[out]	返回页面的宽
height	Double	[out]	返回页面的高

**InsertLayer 方法**

描述 在指定位置插入一个 Layer。

Visual Basic Map.InsertLayer pLayer, index

C++ HRESULT pMap -> InsertLayer(IMapLayer \*pLayer, long index);

参数说明：

参数	类型	出口/入口	描述
pLayer	MapLayer	[in]	要插入的 Layer 对象
index	Long	[in]	插入的位置号

**LoadMapFile 方法**

描述 装载 Map 文件。

Visual Basic Map.LoadMapFile [strFileName]

C++ HRESULT pMap -> LoadMapFile(BSTR strFileName);

参数说明：

参数	类型	出口/入口	描述
StrFileName	String	[in]	Map 的文件名称 可选；缺省值：“ ”。（打开文件选取对话框）

**MapOutput 方法**

描述 输出 Map 的内容。

Visual Basic Map.MapOutput pMapDC, [hasRunTimeDlg]

C++ HRESULT pMap -> MapOutput(IMapGisDC \*pMapDC, short hasRunTimeDlg=1);

参数说明：

参数	类型	出口/入口	描述
pMapDC	MapGisDC	[in]	输出的设备 MapGisDC 对象
hasRunTimeDlg	Integer	[in]	是否显示进程对话框 1—显示 可选；缺省：1

**MoveLayer 方法**

描述 移动指定的 Layer。

Visual Basic Map.MoveLayer pLayer, toindex

C++ HRESULT pMap -> MoveLayer(IMapLayer \*pLayer, long toindex);

参数说明：

参数	类型	出口/入口	描述
pLayer	MapLayer	[in]	要移动的子 Layer 对象
toindex	Long	[in]	移动到的序号

### SaveAsMapFile 方法

描述 另存 Map 文件。

Visual Basic Map.SaveAsMapFile [strFileName]

C++ HRESULT pMap -> SaveAsMapFile(BSTR strFileName);

参数说明：

参数	类型	出口/入口	描述
strFileName	String	[in]	另存 Map 文件的文件名称 可选；缺省：“ ”(弹文件对话框)

### SaveMapFile 方法

描述 保存 Map 文件。

Visual Basic Map.SaveMapFile

C++ HRESULT pMap -> SaveMapFile();

参数说明： 无

### SetPageSize 方法

描述 设置页面的宽和高值。

Visual Basic Map.SetPageSize width, height

C++ HRESULT p Map -> SetPageSize(double width , double height);

参数说明：

参数	类型	出口/入口	描述
width	Double	[in]	页面的宽度
height	Double	[in]	页面的高度

### UpdateActiveLayer 方法

描述 重新获取当前激活的 Layer。

Visual Basic Map.UpdateActiveLayer

C++ HRESULT pMap -> UpdateActiveLayer();

参数说明： 无

## 6.9、LegendItem 对象

描述 对象为所有图例元素的基类对象，封装了所有图例元素的公共操作和实现入口

### 6.9.1、属性：

属性	描述
----	----

Describe	图例元素的描述
String	
可读写	
Type	图例元素的类型
Enum_Area_Type	
只读	
Name	图例元素的名称
String	
可读写	
Code	图例元素的分类码
Double	
可读写	
ModifyFlag	修改标记
Integer	
可读写	
Att	图例元素的属性数据
Record	
可读写	

### 6.9.2、方法：无

## 6.10、PntLegendItem 对象

**描述** PntLegendItem 对象主要封装了点图例元素的数据管理和操作功能。

### 6.10.1、属性：

属性	描述
Describe	点图例元素的描述
String	
可读写	
Type	点图例元素的类型(PNT)
Enum_Area_Type	
只读	
Name	点图例元素的名称
String	
可读写	

Code	点图例元素的分类码
Double	
可读写	
ModifyFlag	修改标记
Integer	
可读写	
Att	点图例元素的属性数据
Record	
可读写	
PntInfo	点图例元素的点参数
Pnt_Info	
可读写	

### 6.10.2、方法：无

## 6.11、LinLegendItem 对象

**描述** LinLegendItem 对象主要封装了线图例元素的数据管理和操作功能。

### 6.11.1、属性：

属性	描述
Describe	线图例元素的描述
String	
可读写	
Type	线图例元素的类型(LIN)
Enum_Area_Type	
只读	
Name	线图例元素的名称
String	
可读写	
Code	线图例元素的分类码
Double	
可读写	
ModifyFlag	修改标记
Integer	
可读写	



Att	线图例元素的属性数据
Record	
可读写	
LinInfo	线图例元素的线参数
Lin_Info	
可读写	

### 6.11.2、方法：无

## 6.12、RegLegendItem 对象

**描述** RegLegendItem 对象主要封装了区图例元素的数据管理和操作功能。

### 6.12.1、属性：

属性	描述
Describe	区图例元素的描述
String	
可读写	
Type	区图例元素的类型(REG)
Enum_Area_Type	
只读	
Name	区图例元素的名称
String	
可读写	
Code	区图例元素的分类码
Double	
可读写	
ModifyFlag	修改标记
Integer	
可读写	
Att	区图例元素的属性数据
Record	
可读写	
RegInfo	区图例元素的区参数
Reg_Info	
可读写	

### 6.12.2、方法：无

## 6.13、MapLegend 对象

**描述** MapLegend 对象主要封装了对图例文件存储和图例元素管理的功能。

### 6.13.1、属性：

属性	描述
MapLegName String 可读写	图例文件名称
Count Long 只读	图例元素总数 参数：Enum_Area_Type：图例类型 可选；缺省 gisGROUP
DispRect D_Rect 可读写	图例显示范围
SelectedLegItem LegendItem 只读	选中的图例元素
ModifyFlag Integer 可读写	修改标记

### 6.13.2、方法：

#### Add 方法

**描述** 添加一个图例元素。

**Visual Basic** MapLegend.Add pLegItem

**C++** HRESULT pMapLegend -> Add(ILegendItem \* pLegItem);

**参数说明：**

参数	类型	出口/入口	描述
pLegItem	LegendItem	[in]	新加的图例元素对象

#### Remove 方法

**描述** 删除一个图例元素。

Visual Basic    sflg = MapLegend.Remove(pLegItem)

C++            HRESULT pMapLegend -> Remove(long no, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
no	Long	[in]	要删除图例元素的序号
sflg	Integer	[out,retval]	成功返回 1，失败返回 0

### RemoveAll 方法

描述            删除所有图例元素。

Visual Basic    MapLegend.RemoveAll

C++            HRESULT pMapLegend -> RemoveAll ();

参数说明：    无

### EditLegItem 方法

描述            编辑图例元素（对话框交互方式）。

Visual Basic    MapLegend.EditLegItem

C++            HRESULT pMapLegend -> EditLegItem();

参数说明：    无

### OpenLegPad 方法

描述            打开图例板。

Visual Basic    MapLegend.OpenLegPad

C++            HRESULT pMapLegend -> OpenLegPad ();

参数说明：    无

### CloseLegPad 方法

描述            关闭图例板。

Visual Basic    MapLegend.CloseLegPad

C++            HRESULT p MapLegend -> CloseLegPad ();

参数说明：    无

### Display 方法

描述            显示图例内容。

Visual Basic    MapLegend.Display pMydc

C++            HRESULT pMapLegend -> Display (IMapGisDC \*pMydc);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

pMydc	MapGisDC	[in]	图例显示的 MapGisDC 对象
-------	----------	------	-------------------

**SetSelectedItem 方法**

描述 设置选中的图例元素。

Visual Basic sflg = MapLegend.SetSelectedItem(no)

C++ HRESULT pMapLegend -> SetSelectedItem (long no, short \*sflg);

参数说明：

参数	类型	出口/入口	描述
no	Long	[in]	图例元素的序号
sflg	Integer	[out,retval]	成功返回 1，失败返回 0

**LoadLegendFile 方法**

描述 装入图例文件。

Visual Basic MapLegend.LoadLegendFile [szFName]

C++ HRESULT pMapLegend -> LoadLegendFile (BSTR szFName);

参数说明：

参数	类型	出口/入口	描述
szFName	String	[in]	图例文件的名称 可选；缺省=" "。(打开文件选择对话框)

**SaveLegendFile 方法**

描述 保存图例文件。

Visual Basic MapLegend.SaveLegendFile

C++ HRESULT pMapLegend -> SaveLegendFile ();

参数说明： 无

**SaveAsLegendFile 方法**

描述 另存图例文件。

Visual Basic MapLegend.SaveAsLegendFile [szNewFName]

C++ HRESULT pMapLegend -> SaveAsLegendFile (BSTR szNewFName);

参数说明：

参数	类型	出口/入口	描述
szNewFName	String	[in]	新图例文件的名称 可选；缺省=" "。(打开文件选择对话框)

**GetItem 方法**

描述 取图例元素的内容。

Visual Basic Set pVal = MapLegend.GetItem(no)

C++ HRESULT pMapLegend -> GetItem (long no, ILegendItem \*\* pVal);

参数说明：

参数	类型	出口/入口	描述
no	Long	[in]	图例元素的序号
pVal	LegendItem	[out,retval]	返回图例元素对象

### UpdateItem 方法

描述 更新图例元素的内容。

Visual Basic MapLegend.UpdateItem no, pVal

C++ HRESULT pMapLegend -> UpdateItem (long pVal, ILegendItem \* pVal);

参数说明：

参数	类型	出口/入口	描述
no	Long	[in]	图例元素的序号
pVal	LegendItem	[in]	更新的图例元素对象

### CloseLegendFile 方法

描述 关闭图例对象。

Visual Basic MapLegend.CloseLegendFile

C++ HRESULT pMapLegend -> CloseLegendFile ();

参数说明： 无

### SetLegParam 方法

描述 设置图例显示参数（对话框交互方式）。

Visual Basic MapLegend.SetLegParam

C++ HRESULT pMapLegend -> SetLegParam ();

参数说明： 无

### WriteLegItemsToFile 方法

描述 把图例元素写到点/线/面文件中。

Visual Basic MapLegend.WriteLegItemsToFile [pntFileName], [linFileName], [regFileName]

C++ HRESULT pMapLegend -> WriteLegItemsToFile (  
BSTR pntFileName, BSTR linFileName, BSTR regFileName);

参数说明：

参数	类型	出口/入口	描述
pntFileName	String	[in]	写入的点文件名称 可选；缺省=" "。（打开文件选择对话框）

linFileName	String	[in]	写入的线文件名称 可选；缺省=" "(打开文件选择对话框)
regFileName	String	[in]	写入的区文件名称 可选；缺省=" "(打开文件选择对话框)

## 6.14、MapPrintParam 对象

**描述** MapPrintParam 对象主要封装了对输出参数的管理功能。

### 6.14.1、属性：

属性	描述
PageWidth	页面宽度
Double	
可读写	
PageHeight	页面高度
Double	
可读写	
TranX	X 方向偏移
Double	
可读写	
TranY	Y 方向偏移
Double	
可读写	
ScaleX	X 方向比例
Double	
可读写	
ScaleY	Y 方向比例
Double	
可读写	
Angle	旋转角度
Double	
可读写	
ModifyFlag	修改标记
Integer	
可读写	

### 6.14.2、方法：

**FitToPage 方法**

描述	按照页面大小设置输出参数。
Visual Basic	MapPrintParam.FitToPage
C++	HRESULT pMapPrintParam->FitToPage();
参数说明：	无

**AutoCheckPage 方法**

描述	系统自动检测幅面。
Visual Basic	MapPrintParam.AutoCheckPage
C++	HRESULT pMapPrintParam->AutoCheckPage ();
参数说明：	无

**6.15、MapNorthArrow 对象**

描述 MapNorthArrow 对象主要封装了 Map 指北针的显示和管理功能。（下一版本发布。）

**6.16、MapScaleBar 对象**

描述 MapScaleBar 对象主要封装了 Map 比例尺的显示和管理功能。（下一版本发布。）

**6.17、MapScaleText 对象**

描述 MapScaleText 对象主要封装了 Map 比例文本的显示和管理功能。（下一版本发布。）

**6.18、MapSurround 对象**

描述 MapSurround 对象主要封装了 Map 外围绑定对象的操作和管理功能。（下一版本发布。）

**6.19、SelectItem 对象**

描述 SelectItem 对象封装了选择实体的数据管理和操作功能。

**6.19.1、属性：**

属性	描述
----	----

ItemNo	选择实体的实体号
Long	
可读写	
Type	选择实体的类型
Enum_Entity_Type	
可读写	
WorkArea	工作区接口
WorkArea	
可读写	

### 6.19.2、方法：无

## 6.20、SelectSet 对象

**描述** SelectSet 对象是多个选择实体的集合，并提供了相应的管理功能

### 6.20.1、属性：

属性	描述
count	选择实体的个数
Long	
只读	
Item	根据序号取一个选择实体
SelectItem	参数：Long：序号
只读	
Name	选择实体集合的名称
String	
可读写	

### 6.20.2、方法：

#### Add 方法

**描述** 添加一个选择实体。

**Visual Basic** SelectSet.Add pItem

**C++** HRESULT pSelectSet-> Add(ISelectItem \*pItem);

**参数说明：**

参数	类型	出口/入口	描述
pItem	SelectItem	[in]	新添加的选择实体对象



**Combine 方法**

描述 合并另外一个选择实体集合。

Visual Basic SelectSet.Combine pSelSet

C++ HRESULT pSelectSet-> Combine(ISelectSet \*pSelSet);

参数说明：

参数	类型	出口/入口	描述
pSelSet	SelectSet	[in]	另外一个选择实体集合对象

**Display 方法**

描述 显示选择实体集合的内容。

Visual Basic SelectSet.Display pDC, [col]

C++ HRESULT pSelectSet-> Display(IMapGisDC \*pDC, long col);

参数说明：

参数	类型	出口/入口	描述
pDC	MapGisDC	[in]	显示集合内容的 MapGisDC 对象
col	Long	[in]	显示的颜色

**Remove 方法**

描述 根据序号从集合中去掉一个选择实体。

Visual Basic SelectSet.Remove no

C++ HRESULT pSelectSet-> Remove(long no);

参数说明：

参数	类型	出口/入口	描述
no	Long	[in]	要去掉选择实体的序号

**RemoveAll 方法**

描述 删除集合中所有的选择实体。

Visual Basic SelectSet.RemoveAll

C++ HRESULT pSelectSet->RemoveAll();

## 第七章 MSI 对象说明

MSI 对象的使用示例代码包含在 D:\MAPGIS61\VBDemo\VBForMsi 目录下。

### 7.1、RasterArea 对象

描述 RasterArea 对象封装了对图象工作区的操作。

#### 7.1.1、属性：

属性	描述
Type	工作区类型
Enum_Area_Type	
只读	
Name	工作区名称
String	
可读写	
ShowErrMsg	显示错误信息标志
Boolean	
可读写	
Warning	对不能后悔的操作是否给出警告提示
Boolean	
可读写	
Compact	保存工作区时压缩数据标志
Boolean	
可读写	
Empty	工作区是否为空标志
Boolean	
只读	
LastErrCode	最后错误码
Long	
只读	
Rect	图形范围
D_Rect	
只读	

IBands	图象波段数
Long	
只读	
ILines	图象行数
Long	
只读	
ILineCells	图象每行的象员数
Long	
只读	
IBandsInfo	图象波段信息
Enum_BandsInfo_Msi	
只读	
ICellType	图象象员类型
Enum_CellType_Msi 只读	
dNullCell	无定义象员的值
Double	
读写	
IHasNullCell	是否有无定义象员的值
Long	
读写	
dXResolution	X 方向的分辨率
Double	
读写	
dYResolution	Y 向的分辨率
Double	
读写	
lResolutionUnit	图象的分辨率单位
Long	
读写	
IsSupportProjection	是否支持投影变换
Enum_ISROJECTION_Msi	
读写	
Rect	图象范围
D_Rect	
读写	

lCoordinateSysType	缺省坐标系统
Enum_COORDTYPE_Msi	
读写	
lCoordinateUnit	缺省坐标单位
Enum_COORDUNIT_Msi	
读写	
lHwToCoForm	坐标缺省正变换方式
Long t	
只读	
GCPFileName	控制点文件名
String	
读写	
GCPDescription	控制点文件说明
String	
读写	
GCPNums	控制点数
Long	
只读	
PolynomialForm	坐标图与参考图的变换关系,多项式的次数
Enum_PolynomialForm_Msi	
读写	
ReSampleForm	加坐标图的重采样的插值方式
Enum_ReSampleType_Msi	
读写	
CompressionForm	压缩方式
Enum_Compression_Msi	
只读	
DataUseForm	数据使用方式
Long	
只写	
TileBlockSize	数据块的大小
Long	
只写	
PyramidLayerNums	金字塔层数
Long	
只读	

PyramidLayerIndexNums	金字塔层索引项数
Long	
只读	
SensorType	传感器类型
Long	
只写	
lResampleStep	采样步长
Long	
只写	
lStatisticsTag	统计计算标志
Long	
只写	
dXScale	象元 X 方向的分辨率
Double	
只写	
dYScale	象元 Y 方向的分辨率
Double	
只写	
dZScale	象元 Z 方向的分辨率
Double	
只写	
dXoCoor	X 的原点
Double	
只写	
dYoCoor	Y 的原点
Double	
只读	
dZoCoor	Z 的原点
Double	
只写	
lZUnit	属性坐标单位
Enum_COORDUNIT_Msi	
只写	
lCompressionQuality	压缩质量
Enum_CompressQuality_Msi	
读写	

DisplayForm	取图象的显示方式
Enum_DisplayForm_Msi	
只读	
DispProcessForm	取显示图象的处理方式
Enum_DisPlayProc_Msi	
只读	
Gl原因DisplayBandNo	Gl原因 显示方式的波段号参数
Long	
只读	

## 7.1.2、方法：

### OpenMsiFileForRead 方法

描述 只读方式打开图象文件

Visual Basic rtl = RasterArea.OpenMsiFileForRead(strMsiFileName)

C++ HRESULT pRasterArea-> OpenMsiFileForRead (BSTR \* strMsiFileName, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
strMsiFileName	String	[in]	打开的文件名
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

### OpenMsiFileForReadWrite 方法

描述 读写方式打开图象文件

Visual Basic rtl = RasterArea.OpenMsiFileForReadWrite(strMsiFileName)

C++ HRESULT pRasterArea->OpenMsiFileForReadWrite( BSTR\* strMsiFileName, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
strMsiFileName	String	[in]	打开的文件名
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

### CreateNewMsiFileForReadWrite 方法

描述 创建一个新的图象文件

Visual Basic rtl = RasterArea. CreateNewMsiFileForReadWrite( StrMsiFileName, ICellType, ILines, ILineCells, IBands, IBandsInfo)

C++ HRESULT pRasterArea->CreateNewMsiFileForReadWrite( BSTR \* strMsiFileName,long ICellType,long ILines, long ILineCells,

long lBands, Enum\_BandsInfo\_Msi lBandsInfo, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
strMsiFileName	String	[in]	打开的文件名
lCellType	Long	[in]	象元类型
lLines	Long	[in]	图象的行数
lLineCells	Long	[in]	每行的象元数
lBands	Long	[in]	波段数
lBandsInfo	Enum_BandsInfo_Msi	[in]	波段信息
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

#### GetBandsMinCell 方法

描述 得到一个波段的最小值

Visual Basic pdBandMinCell = RasterArea.GetBandsMinCell (lBandNo)

C++ HRESULT pRasterArea-> GetBandsMinCell (long lBandNo,  
double \* dBandMinCell);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
dBandMinCell	Double	[out]	该波段的最小值

#### GetBandsMaxCell 方法

描述 得到一个波段的最大值

Visual Basic dBandMaxCell = RasterArea.GetBandsMaxCell (lBandNo)

C++ HRESULT pRasterArea-> GetBandsMaxCell (long lBandNo,  
double \* dBandMaxCell);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
dBandMaxCell	Double	[out]	该波段的最大值

#### GetCellBitSize 方法

描述 得到象员坐标位数的大小

Visual Basic pVal = RasterArea.GetCellBitSize (lCellType)

C++ HRESULT pRasterArea-> GetCellBitSize (  
Enum\_CellType\_Msi lCellType , long \*pVal);

参数说明：

参数	类型	出口/入口	描述
lCellType	Enum_CellType_Msi	[in]	象元类型
pVal	Long	[out]	象元坐标位数的大小

**EqualizeALayer 方法**

描述 均衡化一个图层

Visual Basic RasterArea.EqualizeALayer lBandNo, pwHistReMapTab

C++ HRESULT pRasterArea-> EqualizeALayer (long lBandNo, USHORT \*pwHistReMapTab);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pwHistReMapTab	Integer	[in]	直方图的重映射表的指针

**GetHistsPointer 方法**

描述 取直方图的指针

Visual Basic Set ppdwHist = RasterArea.GetHistsPointer(lBandNo)

C++ HRESULT pRasterArea-> GetHistsPointer (long lBandNo, ILONGList \*\*ppdwHist);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
ppdwHist	LONGList	[out]	直方图的数据

**InitialHistReMapTab 方法**

描述 初始化直方图的重映射表

Visual Basic RasterArea.InitialHistReMapTab

C++ HRESULT pRasterArea-> InitialHistReMapTab();

参数说明： 无

**InitialOneHistReMapTab 方法**

描述 初始化直方图的一个重映射表

Visual Basic RasterArea.InitialOneHistReMapTab ReMapTabNo

C++ HRESULT pRasterArea-> InitialOneHistReMapTab( Enum\_ReMapTab\_Msi\* ReMapTabNo);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----



ReMapTabNo	Enum_ReMapTab_Msi	[in]	重映射表号
------------	-------------------	------	-------

### LinearALayerByHistSum 方法

描述 线形拉伸一个图层

Visual Basic RasterArea.LinearALayerByHistSum lBandNo, pwHistReMapTab, fX, fY

C++ HRESULT pRasterArea->LinearALayerByHistSum(long lBandNo,  
USHORT \*pwHistReMapTab,float fX,float fY);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pwHistReMapTab	Integer	[in]	直方图的重映射表的指针
fX	Single	[in]	拉伸在 X 方向的比例
fY	Single	[in]	拉伸在 Y 方向的比例

### BootALayerByHistSum 方法

描述 平方根拉伸一个图层

Visual Basic RasterArea.BootALayerByHistSum lBandNo, pwHistReMapTab, fX, fY

C++ HRESULT pRasterArea->BootALayerByHistSum(long lBandNo,  
USHORT \*pwHistReMapTab,float fX,float fY);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pwHistReMapTab	Integer	[in]	直方图的重映射表的指针
fX	Single	[in]	拉伸在 X 方向的比例
fY	Single	[in]	拉伸在 Y 方向的比例

### SquareALayerByHistSum 方法

描述 平方拉伸一个图层

Visual Basic RasterArea.SquareALayerByHistSum lBandNo, pwHistReMapTab, fX, fY

C++ HRESULT pRasterArea->SquareALayerByHistSum(long lBandNo,  
USHORT \*pwHistReMapTab,float fX,float fY);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pwHistReMapTab	Integer	[in]	直方图的重映射表的指针
fX	Single	[in]	拉伸在 X 方向的比例
fY	Single	[in]	拉伸在 Y 方向的比例

**NormalizeALayer 方法**

描述 高斯函数拉伸一个图层

Visual Basic RasterArea.NormalizeALayer lBandNo, pwHistReMapTab, fHalfWidth

C++ HRESULT pRasterArea->NormalizeALayer(long lBandNo,  
USHORT \* pwHistReMapTab, float fHalfWidth);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pwHistReMapTab	Integer	[in]	直方图的重映射表的指针
fHalfWidth	Single	[in]	高斯函数的分布半宽度

**InvertALayer 方法**

描述 负变换一个图层

Visual Basic RasterArea.InvertALayer lBandNo, pwHistReMapTab

C++ HRESULT pRasterArea-> InvertALayer (long lBandNo ,  
USHORT \*pwHistReMapTab);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pwHistReMapTab	Integer	[in]	直方图的重映射表的指针

**ComputeColorLutNums 方法**

描述 由像元的类型和波段信息求彩色查找表的数目

Visual Basic lColorLutNums = RasterArea.ComputeColorLutNums(lCellType,lBandsInfo)

C++ HRESULT pRasterArea->ComputeColorLutNums(Enum\_CellType\_Msi lCellType,  
Enum\_BandsInfo\_Msi lBandsInfo, long \*lColorLutNums);

参数说明：

参数	类型	出口/入口	描述
lCellType	Enum_CellType_Msi	[in]	象元类型
lBandsInfo	Enum_BandsInfo_Msi	[in]	波段信息
lColorLutNums	Long	[out]	彩色查找表的数目

**InvertALayerByHistSum 方法**

描述 负变换一个图层

Visual Basic RasterArea.InvertALayerByHistSum lBandNo, pwHistReMapTab, fX, fY

C++ HRESULT pRasterArea-> InvertALayerByHistSum (long lBandNo,

USHORT\* pwHistReMapTab, float fX, float fY);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pwHistReMapTab	Integer	[in]	直方图的重映射表的指针
fX	Single	[in]	拉伸在 X 方向的比例
fY	Single	[in]	拉伸在 Y 方向的比例

#### SetGrayColorLut 方法

描述 设置灰度彩色查找表

Visual Basic RasterArea.SetGrayColorLut

C++ HRESULT pRasterArea-> SetGrayColorLut ();

参数说明： 无

#### AllocControlPointFile 方法

描述 分配几何校正,图象坐标参照的控制点的信息的结构

Visual Basic RasterArea.AllocControlPointFile dwGCPNum

C++ HRESULT pRasterArea-> AllocControlPointFile (ULONG dwGCPNum);

参数说明：

参数	类型	出口/入口	描述
dwGCPNum	Long	[in]	控制点数目

#### FreeControlPointFile 方法

描述 释放分配几何校正,图象坐标参照的控制点的信息的结构

Visual Basic RasterArea.FreeControlPointFile

C++ HRESULT pRasterArea-> FreeControlPointFile ();

参数说明： 无

#### InitialControlPointFile 方法

描述 初始化几何校正,图象坐标参照的控制点的信息的结构

Visual Basic RasterArea.InitialControlPointFile dwGCPNum

C++ HRESULT pRasterArea-> InitialControlPointFile (ULONG dwGCPNum);

参数说明：

参数	类型	出口/入口	描述
dwGCPNum	Long	[in]	控制点数目

**SetDefaultControlPointFile 方法**

描述 设置缺省的控制点参数

Visual Basic RasterArea.SetDefaultControlPointFile

C++ HRESULT pRasterArea-> SetDefaultControlPointFile ();

参数说明： 无

**SetHMirrorControlPointFile 方法**

描述 设置水平翻转的控制点参数

Visual Basic RasterArea.SetHMirrorControlPointFile

C++ HRESULT pRasterArea-> SetHMirrorControlPointFile ();

参数说明： 无

**SetVMirrorControlPointFile 方法**

描述 设置垂直翻转的控制点参数

Visual Basic RasterArea.SetVMirrorControlPointFile

C++ HRESULT pRasterArea-> SetVMirrorControlPointFile ();

参数说明： 无

**SetOneGCP 方法**

描述 设置一个控制点的参数

Visual Basic RasterArea.SetOneGCP dwPointNo, dSX, dSY ,dDX, dDY

C++ HRESULT pRasterArea-> SetOneGCP (ULONG dwPointNo, double dSX, double dSY ,double dDX, double dDY);

参数说明：

参数	类型	出口/入口	描述
dwPointNo	Long	[in]	控制点号
dSX	Double	[in]	参考图(图形)的 X 坐标
dSY	Double	[in]	参考图(图形)的 Y 坐标
dDX	Double	[in]	加坐标图(图象)的 X 坐标
dDY	Double	[in]	加坐标图(图象)的 Y 坐标

**GetOneGCP 方法**

描述 得到一个控制点的参数

Visual Basic RasterArea.GetOneGCP (dwPointNo, dSX, dSY ,dDX, dDY)

C++ HRESULT pRasterArea-> SetOneGCP (ULONG dwPointNo, double \*dSX, double \*dSY ,double \*dDX, double \*dDY);

参数说明：

参数	类型	出口/入口	描述
dwPointNo	Long	[in]	控制点号
dSX	Double	[out]	参考图(图形)的 X 坐标
dSY	Double	[out]	参考图(图形)的 Y 坐标
dDX	Double	[out]	加坐标图(图象)的 X 坐标
dDY	Double	[out]	加坐标图(图象)的 Y 坐标

#### SetOnePointError 方法

描述 设置一个控制点的计算残差

Visual Basic RasterArea.SetOnePointError dwPointNo, dDErr

C++ HRESULT pRasterArea-> SetOnePointError (ULONG dwPointNo, double dDErr);

参数说明：

参数	类型	出口/入口	描述
dwPointNo	Long	[in]	控制点号
dDErr	Double	[in]	控制点的计算残差

#### GetOnePointError 方法

描述 取一个控制点的计算残差

Visual Basic dDErr = RasterArea.GetOnePointError dwPointNo

C++ HRESULT pRasterArea-> GetOnePointError (ULONG dwPointNo ,double \* dDErr);

参数说明：

参数	类型	出口/入口	描述
dwPointNo	Long	[in]	控制点数目
dDErr	Double	[out]	计算残差

#### InitialControlPointFile 方法

描述 初始化几何校正,图象坐标参照的控制点的信息的结构

Visual Basic RasterArea.InitialControlPointFile dwGCPNum

C++ HRESULT pRasterArea-> InitialControlPointFile (ULONG dwGCPNum);

参数说明：

参数	类型	出口/入口	描述
dwGCPNum	Long	[in]	控制点数目

#### SetOneGCPUseOnOff 方法

描述 设置一个控制点的使用开关

Visual Basic RasterArea.SetOneGCPUseOnOff dwPointNo, IUseOnOff

C++ HRESULT pRasterArea-> SetOneGCPUseOnOff (ULONG dwPointNo,

---

```
Enum_Onoff_Msi IUseOnOff);
```

参数说明：

参数	类型	出口/入口	描述
dwPointNo	Long	[in]	控制点号
IUseOnOff	Enum_Onoff_Msi	[in]	控制点的使用开关

### GetOneGCPUseOnOff 方法

描述 取一个控制点的使用开关

Visual Basic IUseOnOff = RasterArea.GetOneGCPUseOnOff (dwPointNo)

C++ HRESULT pRasterArea-> GetOneGCPUseOnOff (ULONG dwPointNo,  
Enum\_Onoff\_Msi \*IUseOnOff);

参数说明：

参数	类型	出口/入口	描述
dwPointNo	Long	[in]	控制点号
IUseOnOff	Enum_Onoff_Msi	[out]	控制点的使用开关

### SetOneGCPEditOnOff 方法

描述 设置一个控制点的编辑开关

Visual Basic RasterArea.SetOneGCPEditOnOff dwPointNo, IEditOnOff

C++ HRESULT pRasterArea-> SetOneGCPEditOnOff (ULONG dwPointNo,  
Enum\_Onoff\_Msi IEditOnOff);

参数说明：

参数	类型	出口/入口	描述
dwPointNo	Long	[in]	控制点号
IEditOnOff	Enum_Onoff_Msi	[in]	控制点的编辑开关

### GetOneGCPEditOnOff 方法

描述 取一个控制点的编辑开关

Visual Basic IEditOnOff = RasterArea. GetOneGCPEditOnOff (dwPointNo)

C++ HRESULT pRasterArea-> GetOneGCPEditOnOff (ULONG dwPointNo,  
Enum\_Onoff\_Msi \*IEditOnOff);

参数说明：

参数	类型	出口/入口	描述
dwPointNo	Long	[in]	控制点号
IEditOnOff	Enum_Onoff_Msi	[out]	控制点的编辑开关

### AddOneControlPoint 方法

描述 加控制点到 MSI 文件

Visual Basic RasterArea.AddOneControlPoint dwPointNo, lOnOff, lEditable, dSX, dSY, dDX, dDY

C++ HRESULT pRasterArea-> AddOneControlPoint (ULONG dwPointNo,  
Enum\_Onoff\_Msi lOnOff, Enum\_Onoff\_Msi lEditable,  
double dSX, double dSY, double dDX, double dDY);

参数说明：

参数	类型	出口/入口	描述
dwPointNo	Long	[in]	控制点号
lOnOff	Enum_Onoff_Msi	[in]	控制点的使用开关
lEditable	Enum_Onoff_Msi	[in]	控制点的编辑开关
dSX	Double	[in]	参考图(图形)的 X 坐标
dSY	Double	[in]	参考图(图形)的 Y 坐标
dDX	Double	[in]	加坐标图(图象)的 X 坐标
dDY	Double	[in]	加坐标图(图象)的 Y 坐标

### DeleteOneControlPoint 方法

描述 删除一个控制点

Visual Basic RasterArea.DeleteOneControlPoint dwPointNo

C++ HRESULT pRasterArea-> DeleteOneControlPoint (ULONG dwPointNo);

参数说明：

参数	类型	出口/入口	描述
dwPointNo	Long	[in]	控制点号

### UppdataOneControlPoint 方法

描述 更新控制点到 MSI 文件

Visual Basic RasterArea.UppdataOneControlPoint dwPointNo, lOnOff, lEditable,  
dSX, dSY, dDX, dDY

C++ HRESULT pRasterArea-> UppdataOneControlPoint (ULONG dwPointNo,  
Enum\_Onoff\_Msi lOnOff, Enum\_Onoff\_Msi lEditable,  
double dSX, double dSY, double dDX, double dDY);

参数说明：

参数	类型	出口/入口	描述
dwPointNo	Long	[in]	控制点号
lOnOff	Enum_Onoff_Msi	[in]	控制点的使用开关
lEditable	Enum_Onoff_Msi	[in]	控制点的编辑开关
dSX	Double	[in]	参考图(图形)的 X 坐标

dSY	Double	[in]	参考图(图形)的 Y 坐标
dDX	Double	[in]	加坐标图(图象)的 X 坐标
dDY	Double	[in]	加坐标图(图象)的 Y 坐标

### ComputeGCPErrors 方法

描述 计算控制点的残差

Visual Basic RasterArea.ComputeGCPErrors

C++ HRESULT pRasterArea-> ComputeGCPErrors ();

参数说明： 无

### MapPntToImgPnt 方法

描述 地理坐标转变到图像坐标

Visual Basic RasterArea.MapPntToImgPnt (dX,dY,pdX,pdY)

C++ HRESULT pRasterArea-> MapPntToImgPnt (double dX,double dY,  
double \*pdX, double pdY);

参数说明：

参数	类型	出口/入口	描述
dX	Double	[in]	地理坐标 X
dY	Double	[in]	地理坐标 Y
pdX	Double	[out]	图像坐标 X
pdY	Double	[out]	图像坐标 Y

### ImgPntToMapPnt 方法

描述 图像坐标转换到地理坐标

Visual Basic RasterArea.ImgPntToMapPnt (dX,dY,pdX,pdY)

C++ HRESULT pRasterArea-> ImgPntToMapPnt (double dX,double dY,  
double \*pdX, double pdY);

参数说明：

参数	类型	出口/入口	描述
dX	Double	[in]	图像坐标 X
dY	Double	[in]	图像坐标 Y
pdX	Double	[out]	地理坐标 X
pdY	Double	[out]	地理坐标 Y

### GetUserCoordinateRange 方法

描述 取用户坐标范围,用户坐标是当前图象显示所用的坐标

Visual Basic RasterArea.GetUserCoordinateRange (IDrawCoordSysType,



pdXmin, pdXmax, pdYmin, pdYmax)  
C++ HRESULT pRasterArea-> GetUserCoordinateRange (  
Enum\_COORDTYPE\_Msi lDrawCoordSysType,  
double \*pdXmin, double \*pdXmax, double \*pdYmin, double \*pdYmax)  
参数说明：

参数	类型	出口/入口	描述
lDrawCoordSysType	Enum_COORDTYPE_Msi	[in]	坐标类型
pdXmin	Double	[out]	用户坐标范围 Xmin
pdXmax	Double	[out]	用户坐标范围 Xmax
pdYmin	Double	[out]	用户坐标范围 Ymin
pdYmax	Double	[out]	用户坐标范围 Ymax

#### GetPointNums 方法

描述 取 POINT 数  
Visual Basic dwPointNums = RasterArea.GetPointNums (dwAoiNo,dwPolygonNo)  
C++ HRESULT pRasterArea-> GetPointNums (ULONG dwAoiNo  
ULONG DwPolygonNo, ULONG \* dwPointNums);

参数说明：

参数	类型	出口/入口	描述
dwAoiNo	Long	[in]	区域号
dwPolygonNo	Long	[in]	多边形号
dwPointNums	Long	[out]	POINT 数

#### GetPolygonNums 方法

描述 取 POLYGON 数  
Visual Basic dwPolygonNums = RasterArea. GetPolygonNums (dwAoiNo)  
C++ HRESULT pRasterArea-> GetPolygonNums (ULONG dwAoiNo,  
ULONG \* dwPolygonNums);

参数说明：

参数	类型	出口/入口	描述
dwAoiNo	Long	[in]	区域号
dwPolygonNums	Long	[out]	多边形数目

#### DeleteOneAoi 方法

描述 删除一个区  
Visual Basic RasterArea.DeleteOneAoi dwAoiNo  
C++ HRESULT pRasterArea-> DeleteOneAoi (ULONG dwAoiNo);

参数说明：

参数	类型	出口/入口	描述
dwAoiNo	Long	[in]	区域号

#### CopmuteTileCeff 方法

描述 计算分块参数

Visual Basic RasterArea.CopmuteTileCeff

C++ HRESULT pRasterArea-> CopmuteTileCeff ();

参数说明： 无

#### GetTileCeff 方法

描述 取分块参数

Visual Basic RasterArea.GetTileCeff (plTileHeight, plTileWidth)

C++ HRESULT pRasterArea-> GetTileCeff (long \*plTileHeight, long \*plTileWidth);

参数说明：

参数	类型	出口/入口	描述
plTileHeight	Long	[out]	分块高度
plTileWidth	Long	[out]	分块宽度

#### CopmutePyramidLayerNums 方法

描述 计算金字塔层数

Visual Basic RasterArea.CopmutePyramidLayerNums

C++ HRESULT pRasterArea-> CopmutePyramidLayerNums ();

参数说明： 无

#### SetDefaultPyramidLayerIndex 方法

描述 设置缺省的金字塔层索引项

Visual Basic RasterArea.SetDefaultPyramidLayerIndex

C++ HRESULT pRasterArea-> SetDefaultPyramidLayerIndex ();

参数说明： 无

#### SetDefaultGrayContrastTab 方法

描述 设置灰度变换 CONTRAST，变换表项均为 256 项

Visual Basic RasterArea. SetDefaultGrayContrastTab ()

C++ HRESULT pRasterArea-> SetDefaultGrayContrastTab ();

参数说明： 无

**SetDefaultHistogramTab 方法**

描述	设置波段直方图表
Visual Basic	RasterArea.SetDefaultHistogramTab
C++	HRESULT pRasterArea-> SetDefaultHistogramTab ();
参数说明：	无

**SetDefaultBandsMean 方法**

描述	设置各个波段的缺省均值
Visual Basic	RasterArea.SetDefaultBandsMean
C++	HRESULT pRasterArea-> SetDefaultBandsMean ();
参数说明：	无

**SetDefaultBandsMedin 方法**

描述	设置各个波段的波段中值
Visual Basic	RasterArea.SetDefaultBandsMedin
C++	HRESULT pRasterArea-> SetDefaultBandsMedin ();
参数说明：	无

**SetDefaultBandsMode 方法**

描述	设置各个波段的波段最频值
Visual Basic	RasterArea.SetDefaultBandsMode
C++	HRESULT pRasterArea-> SetDefaultBandsMode ();
参数说明：	无

**SetDefaultBandsStd 方法**

描述	设置各个波段的波段标准差
Visual Basic	RasterArea.SetDefaultBandsStd
C++	HRESULT pRasterArea-> SetDefaultBandsStd ();
参数说明：	无

**SetDefaultBandsEValue 方法**

描述	设置各个波段间特征值
Visual Basic	RasterArea.SetDefaultBandsEValue
C++	HRESULT pRasterArea-> SetDefaultBandsEValue ();
参数说明：	无

**SetDefaultBandsEVector 方法**

描述	设置各个特征向量
----	----------

Visual Basic RasterArea.SetDefaultBandsEVector  
C++ HRESULT pRasterArea-> SetDefaultBandsEVector ();  
参数说明： 无

### SetDefaultBandsCovt 方法

描述 设置各个波段间的协方差  
Visual Basic RasterArea.SetDefaultBandsCovt  
C++ HRESULT pRasterArea-> SetDefaultBandsCovt ();  
参数说明： 无

### GetCell 方法

描述 取象元  
Visual Basic dCellVal = RasterArea.GetCell (lPyramidLayerNo, lBandNo, lLineNo,lLineCellNo)  
C++ HRESULT pRasterArea-> GetCell (long lPyramidLayerNo,  
long lBandNo,long lLineNo,long lLineCellNo,double \* dCellVal);

参数说明：

参数	类型	出口/入口	描述
lPyramidLayerNo	Long	[in]	金字塔层号
lBandNo	Long	[in]	波段号
lLineNo	Long	[in]	行
lLineCellNo	Long	[in]	列
dCellVal	Double	[out]	象元值

### SetCell 方法

描述 设象元  
Visual Basic RasterArea. SetCell (lPyramidLayerNo, lBandNo, lLineNo,lLineCellNo, dCellVal)  
C++ HRESULT pRasterArea-> SetCell (long lPyramidLayerNo,  
long lBandNo,long lLineNo,long lLineCellNo,double dCellVal);

参数说明：

参数	类型	出口/入口	描述
lPyramidLayerNo	Long	[in]	金字塔层号
lBandNo	Long	[in]	波段号
lLineNo	Long	[in]	行
lLineCellNo	Long	[in]	列
dCellVal	Double	[in]	象元值

### UpdatePyramidLayerByCell 方法

描述 更新金字塔层

Visual Basic RasterArea.UpdatePyramidLayerByCell lBandNo, lLineNo, lLineCellNo, dCellValue

C++ HRESULT pRasterArea-> UpdatePyramidLayerByCell (long lBandNo,  
Long lLineNo,long lLineCellNo,double dCellValue);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
lLineNo	Long	[in]	行
lLineCellNo	Long	[in]	列
dCellVal	Double	[in]	象元值

### AllocInitialPyramidAndTileVar 方法

描述 初始分配金字塔与图块

Visual Basic RasterArea.AllocInitialPyramidAndTileVar

C++ HRESULT pRasterArea-> AllocInitialPyramidAndTileVar ();

参数说明： 无

### FreePyramidAndTileVar 方法

描述 释放金字塔与图块数据

Visual Basic RasterArea.FreePyramidAndTileVar

C++ HRESULT pRasterArea-> FreePyramidAndTileVar ();

参数说明： 无

### SetDefaultPyramidLayerIndexPoint 方法

描述 设置缺省金字塔层索引指针

Visual Basic RasterArea.SetDefaultPyramidLayerIndexPoint

C++ HRESULT pRasterArea-> SetDefaultPyramidLayerIndexPoint ();

参数说明： 无

### LoadOneTile 方法

描述 装入一个图块

Visual Basic RasterArea.LoadOneTile lTileIndex

C++ HRESULT pRasterArea-> LoadOneTile (long lTileIndex);

参数说明：

参数	类型	出口/入口	描述
lTileIndex	Long	[in]	图块索引

**UnLoadOneTile 方法**

描述 卸载一个图块

Visual Basic RasterArea.UnLoadOneTile lTileIndex

C++ HRESULT pRasterArea-> UnLoadOneTile (long lTileIndex);

参数说明：

参数	类型	出口/入口	描述
lTileIndex	Long	[in]	图块索引

**UnLoadAllTiles 方法**

描述 卸载所有的图块

Visual Basic RasterArea.UnLoadAllTiles

C++ HRESULT pRasterArea-> UnLoadAllTiles ();

参数说明： 无

**BulidPyarmidLayer 方法**

描述 建立金字塔层

Visual Basic RasterArea.BulidPyarmidLayer

C++ HRESULT pRasterArea-> BulidPyarmidLayer ();

参数说明： 无

**ComputerStaticalData 方法**

描述 计算统计信息

Visual Basic RasterArea.ComputerStaticalData

C++ HRESULT pRasterArea-> ComputerStaticalData ();

参数说明： 无

**SetBandsMean 方法**

描述 设置一个波段的均值

Visual Basic RasterArea.SetBandsMean lBandNo,pVal

C++ HRESULT pRasterArea-> SetBandsMean (long lBandNo,double pVal);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pVal	Double	[in]	波段的均值

### SetBandsMedian 方法

描述 设置一个波段的中值

Visual Basic RasterArea.SetBandsMedian lBandNo, pVal

C++ HRESULT pRasterArea-> SetBandsMedian (long lBandNo,double pVal);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pVal	Double	[in]	波段的中值

### SetBandsMode 方法

描述 设置一个波段的最频值

Visual Basic RasterArea.SetBandsMode lBandNo, pVal

C++ HRESULT pRasterArea-> SetBandsMode (long lBandNo,double pVal);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pVal	Double	[in]	波段的最频值

### SetBandsStd 方法

描述 设置一个波段的标准差

Visual Basic RasterArea.SetBandsStd lBandNo, pVal

C++ HRESULT pRasterArea-> SetBandsStd (long lBandNo,double pVal);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pVal	Double	[in]	波段的标准差

### SetBandsEValue 方法

描述 设置一个波段间的特征值

Visual Basic RasterArea.SetBandsEValue lEValueNo, pVal

C++ HRESULT pRasterArea-> SetBandsEValue (long lEValueNo,double pVal);

参数说明：

参数	类型	出口/入口	描述
lEValueNo	Long	[in]	波段号
pVal	Double	[in]	波段的特征值

**SetBandsEVector 方法**

描述 设置波段间特征向量

Visual Basic RasterArea.SetBandsEVector (IEVectorNo, pdEVector)

C++ HRESULT pRasterArea->SetBandsEVector(long IEVectorNo, double \*pdEVector);

参数说明：

参数	类型	出口/入口	描述
IEVectorNo	Long	[in]	特征向量编号
pdEVector	Double	[in,out]	波段间特征向量

**SetBandsCovt 方法**

描述 设置波段间的协方差

Visual Basic RasterArea.SetBandsCovt (ICovtNo, pdCovt)

C++ HRESULT pRasterArea-> SetBandsCovt (long ICovtNo, double \*pdCovt);

参数说明：

参数	类型	出口/入口	描述
ICovtNo	Long	[in]	协方差编号
pdCovt	Double	[in,out]	波段间的协方差

**SetBandsHist 方法**

描述 设置一个波段的直方图

Visual Basic RasterArea. SetBandsHist (lBandNo, pdwHist)

C++ HRESULT pRasterArea-> SetBandsHist (long lBandNo, ULONG\*pdwHist);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pdwHist	Long	[in,out]	波段的直方图

**GetPyramidLayerRowColTileNums 方法**

描述 取各个金字塔层的行列分块数

Visual Basic RasterArea.InitialControlPointFile lPyramidLayerNo, plRowTileNums, lColTileNums

C++ HRESULT pRasterArea-> InitialControlPointFile (long lPyramidLayerNo,  
Long \* plRowTileNums,long \* plColTileNums);

参数说明：

参数	类型	出口/入口	描述
lPyramidLayerNo	Long	[in]	金字塔层号
plRowTileNums	Long	[out]	金字塔层的行分块数
plColTileNums	Long	[out]	金字塔层的列分块数



### GetPyramidLayerSize 方法

描述 取各个金字塔层的行列数

Visual Basic RasterArea.GetPyramidLayerSize(IPyramidLayerNo, plLines, plLineCells)

C++ HRESULT pRasterArea-> GetPyramidLayerSize (long IPyramidLayerNo, Long \* plLines,long \* plLineCells);

参数说明：

参数	类型	出口/入口	描述
IPyramidLayerNo	Long	[in]	金字塔层号
plLines	Long	[out]	金字塔层的行数
plLineCells	Long	[out]	金字塔层的列数

### GetBandNoByTileNo 方法

描述 通过图块号得到波段号

Visual Basic lBandNo = RasterArea.GetBandNoByTileNo(lTileNo)

C++ HRESULT pRasterArea->GetBandNoByTileNo(long lTileNo, long \* lBandNo);

参数说明：

参数	类型	出口/入口	描述
lTileNo	Long	[in]	图块号
lBandNo	Long	[out]	波段号

### FindClipAoi 方法

描述 得到裁剪区

Visual Basic RasterArea.FindClipAoi (pAoiNo)

C++ HRESULT pRasterArea-> FindClipAoi (ULONG \*pAoiNo);

参数说明：

参数	类型	出口/入口	描述
pAoiNo	Long	[out]	裁剪区号

### FindFirstClassAoi 方法

描述 得到第一个分类区号

Visual Basic AoiNo = RasterArea.FindFirstClassAoi ()

C++ HRESULT pRasterArea-> FindFirstClassAoi (ULONG \* AoiNo);

参数说明：

参数	类型	出口/入口	描述
AoiNo	Long	[out]	区号

**FindNextClassAoi 方法**

描述 得到下一个分类区号

Visual Basic RasterArea.FindNextClassAoi (pAoiNo, AoiCurNo)

C++ HRESULT pRasterArea-> FindNextClassAoi (  
ULONG \*pAoiNo, ULONG AoiCurNo);

参数说明：

参数	类型	出口/入口	描述
pAoiNo	Long	[out]	下一个分类区号
AoiCurNo	Long	[in]	当前的分类区号

**DeleteClipAoi 方法**

描述 删除裁剪区

Visual Basic RasterArea.DeleteClipAoi

C++ HRESULT pRasterArea-> DeleteClipAoi ();

参数说明： 无

**DeleteAllClassAois 方法**

描述 删除所有的分类区

Visual Basic RasterArea.DeleteAllClassAois

C++ HRESULT pRasterArea-> DeleteAllClassAois ();

参数说明： 无

**MsiBootTest 方法**

描述 平方根显示

Visual Basic RasterArea.MsiBootTest

C++ HRESULT pRasterArea-> MsiBootTest ();

参数说明： 无

**MsiEqualizeView 方法**

描述 均衡化显示

Visual Basic RasterArea.MsiEqualizeView

C++ HRESULT pRasterArea-> MsiEqualizeView ();

参数说明： 无

**MsiInvertView 方法**

描述 反转显示

Visual Basic RasterArea.MsiInvertView

C++ HRESULT pRasterArea-> MsiInvertView ();

参数说明： 无

#### MsiLinearView 方法

描述 线形显示

Visual Basic RasterArea.MsiLinearView

C++ HRESULT pRasterArea-> MsiLinearView ();

参数说明： 无

#### MsiNoneView 方法

描述 原数据显示

Visual Basic RasterArea.MsiNoneView

C++ HRESULT pRasterArea-> MsiNoneView ();

参数说明： 无

#### MsiNormalizeView 方法

描述 正规化显示

Visual Basic RasterArea.MsiNormalizeView

C++ HRESULT pRasterArea-> MsiNormalizeView ();

参数说明： 无

#### MsiSquareView 方法

描述 平方显示

Visual Basic RasterArea.MsiSquareView

C++ HRESULT pRasterArea-> MsiSquareView ();

参数说明： 无

#### MsiBootViewA 方法

描述 平方根显示一个波段

Visual Basic RasterArea.MsiBootViewA lBandNo

C++ HRESULT pRasterArea-> MsiBootViewA (long lBandNo);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号

#### MsiEqualizeViewA 方法

描述 均衡化显示一个波段

Visual Basic RasterArea.MsiEqualizeViewA lBandNo

C++ HRESULT pRasterArea-> MsiEqualizeViewA (long lBandNo);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号

#### MsiInvertViewA 方法

描述 反转显示一个波段

Visual Basic RasterArea.MsiInvertViewA lBandNo

C++ HRESULT pRasterArea-> MsiInvertViewA (long lBandNo);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号

#### MsiLinearViewA 方法

描述 线形显示一个波段

Visual Basic RasterArea.MsiLinearViewA lBandNo

C++ HRESULT pRasterArea-> MsiLinearViewA (long lBandNo);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号

#### MsiNoneViewA 方法

描述 原数据显示一个波段

Visual Basic RasterArea.MsiNoneViewA lBandNo

C++ HRESULT pRasterArea-> MsiNoneViewA (long lBandNo);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号

#### MsiNormalizeViewA 方法

描述 正规化显示一个波段

Visual Basic RasterArea.MsiNormalizeViewA lBandNo

C++ HRESULT pRasterArea-> MsiNormalizeViewA (long lBandNo);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号

**MsiSquareViewA 方法**

描述 平方显示一个波段

Visual Basic RasterArea.MsiSquareViewA lBandNo

C++ HRESULT pRasterArea-> MsiSquareViewA (long lBandNo);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号

**GetBandsMean 方法**

描述 得到波段的均值

Visual Basic pVal = RasterArea.GetBandsMean (lBandNo)

C++ HRESULT pRasterArea-> GetBandsMean (long lBandNo, double \*pVal);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pVal	Double	[out]	波段的均值

**GetBandsStd 方法**

描述 得到波段的标准差

Visual Basic pVal = RasterArea.GetBandsStd (lBandNo)

C++ HRESULT pRasterArea-> GetBandsStd (long lBandNo, double \*pVal);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pVal	Double	[out]	波段的标准差

**GetBandsMedian 方法**

描述 得到波段的中值

Visual Basic pVal = RasterArea.GetBandsMedian (lBandNo)

C++ HRESULT pRasterArea-> GetBandsMedian (long lBandNo, double \*pVal);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pVal	Double	[out]	波段的中值

**GetBandsMode 方法**

描述 得到波段的最频值

Visual Basic pVal = RasterArea.GetBandsMode (lBandNo)

C++ HRESULT pRasterArea-> GetBandsMode (long lBandNo, double \*pVal);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pVal	Double	[out]	波段的最频值

**GetBandsEValue 方法**

描述 得到波段的特征值

Visual Basic pVal = RasterArea.GetBandsEValue (lBandNo)

C++ HRESULT pRasterArea-> GetBandsEValue (long lBandNo, double \*pVal);

参数说明：

参数	类型	出口/入口	描述
lBandNo	Long	[in]	波段号
pVal	Double	[out]	波段的特征值

**GetBandsEVector 方法**

描述 得到波段间的特征向量

Visual Basic pdEVector = RasterArea.GetBandsEVector (lEVectorNo, lBandNo)

C++ HRESULT pRasterArea-> GetBandsEVector (  
long lEVectorNo, long lBandNo, double \* pdEVector);

参数说明：

参数	类型	出口/入口	描述
lEVectorNo	Long	[in]	特征向量号
lBandNo	Long	[in]	波段号
pdEVector	Double	[out]	特征向量

**GetBandsCovt 方法**

描述 得到波段间的协方差

Visual Basic pdCovt = RasterArea.GetBandsCovt (lCovtNo, lBandNo)

C++ HRESULT pRasterArea-> GetBandsCovt(long lCovtNo, long lBandNo, double \*  
pdCovt);

参数说明：

参数	类型	出口/入口	描述
lCovtNo	Long	[in]	协方差行号

lBandNo	Long	[in]	波段号
pdCovt	Double	[out]	协方差

**SetDisplayFormToGRAY 方法**

描述 设置图象的显示方式为 GRAY 显示方式

Visual Basic RasterArea.SetDisplayFormToGRAY lGrayBandNo

C++ HRESULT pRasterArea-> SetDisplayFormToGRAY (long lGrayBandNo);

参数说明：

参数	类型	出口/入口	描述
lGrayBandNo	Long	[in]	波段号

**SetDisplayFormToRGB 方法**

描述 设置图象的显示方式为 RGB 显示方式

Visual Basic RasterArea.SetDisplayFormToRGB lRBnadNo, lGBandNo, lBBandNo

C++ HRESULT pRasterArea-> SetDisplayFormToRGB (long lRBnadNo, long lGBandNo, long lBBandNo);

参数说明：

参数	类型	出口/入口	描述
lRBnadNo	Long	[in]	红波段号
lGBandNo	Long	[in]	绿波段号
lBBandNo	Long	[in]	蓝波段号

**GetRGBDisplayBandNo 方法**

描述 取 RGB 显示方式的波段号参数

Visual Basic RasterArea.GetRGBDisplayBandNo(plRBnadNo, plGBandNo, plBBandNo)

C++ HRESULT pRasterArea-> GetRGBDisplayBandNo (long \*plRBnadNo, long \*plGBandNo, long \*plBBandNo);

参数说明：

参数	类型	出口/入口	描述
plRBnadNo	Long	[out]	红波段号
plGBandNo	Long	[out]	绿波段号
plBBandNo	Long	[out]	蓝波段号

**7.2、RasterLayer 对象**

描述 RasterLayer 对象封装了图象工作区的显示操作。

7.2.1、属性：与 MapLayer 对象属性相同

7.2.2、方法：与 MapLayer 对象接口相同

### 7.3、VectorToMsi 对象

描述 VectorToMsi 对象封装了矢量转栅格的功能。

7.3.1、属性：无

7.3.2、方法：

#### PntToMSIByDlg 方法

描述 点文件转栅格

Visual Basic VectorToMsi.PntToMSIByDlg pArea

C++ HRESULT pVectorToMsi-> PntToMSIByDlg (IPntArea \*pArea);

参数说明：

参数	类型	出口/入口	描述
pArea	IPntArea	[in]	点工作区

#### LinToMSIByDlg 方法

描述 线转栅格

Visual Basic VectorToMsi.LinToMSIByDlg pArea

C++ HRESULT pVectorToMsi-> LinToMSIByDlg (ILinArea \*pArea);

参数说明：

参数	类型	出口/入口	描述
pArea	ILinArea	[in]	线工作区

#### RegToMSIByDlg 方法

描述 区转栅格

Visual Basic VectorToMsi.RegToMSIByDlg pArea

C++ HRESULT pVectorToMsi-> RegToMSIByDlg (IRegArea \*pArea);

参数说明：

参数	类型	出口/入口	描述
pArea	IRegArea	[in]	区工作区



## 7.4、MsiToVector 对象

描述 MsiToVector 对象封装了栅格转矢量的功能。

### 7.4.1、属性：无

### 7.4.2、方法：

#### MsiToReg 方法

描述 栅格转矢量

Visual Basic MsiToVector.MsiToReg msiName, reMsiName, pArea, lBandNo, lMaxCellNumOfSamllReg

C++ HRESULT pMsiToVector-> MsiToReg (BSTR msiName, BSTR reMsiName, IRegArea \*pArea, long lBandNo, long lMaxCellNumOfSamllReg);

参数说明：

参数	类型	出口/入口	描述
msiName	String	[in]	转换的图象文件名称
reMsiName	String	[in]	合并小区后生成的图象名称
pArea	RegArea	[in]	转换生成区文件的对象指针
lBandNo	Long	[in]	转换的波段号
lMaxCellNumOfSamllReg	Long	[in]	进行小区合并的时候的小区象元数

#### MsiToReg2 方法

描述 栅格转矢量

Visual Basic MsiToVector.MsiToReg2 pRaster, reMsiName, pArea, lBandNo, lMaxCellNumOfSamllReg

C++ HRESULT pMsiToVector-> MsiToReg2 (IRasterArea \*pRaster, BSTR reMsiName, IregArea \*pArea, long lBandNo, long lMaxCellNumOfSamllReg);

参数说明：

参数	类型	出口/入口	描述
pRaster	IRasterArea	[in]	转换的图象文件对象指针
reMsiName	String	[in]	合并小区后生成的图象名称
pArea	RegArea	[in]	转换生成区文件的指针
lBandNo	Long	[in]	转换的波段号

IMaxCellNumOfSamllReg	Long	[in]	进行小区合并的时候的小区象元数
-----------------------	------	------	-----------------

### GetMsiToVectParam 方法

描述 得到栅格转矢量的参数

Visual Basic MsiToVector.GetMsiToVectParam (msiName, reMsiName, regName, lBandNo, IMaxCellNumOfSamllReg)

C++ HRESULT pMsiToVector-> GetMsiToVectParam ( BSTR \*msiName, BSTR \*reMsiName, BSTR \*regName, long \*lBandNo,long \*IMaxCellNumOfSamllReg);

参数说明：

参数	类型	出口/入口	描述
MsiName	String	[in,out]	转换的图象文件名称
reMsiName	String	[in,out]	合并小区后生成的图象名称
RegName	String	[in,out]	转换生成区文件名
lBandNo	Long	[out]	转换的波段号
IMaxCellNumOfSamllReg	Long	[out]	进行小区合并的时候的小区象元数

### GetMsiToVectParam2 方法

描述 得到栅格转矢量的参数

Visual Basic MsiToVector. GetMsiToVectParam2 (pRaster, reMsiName, regName, lBandNo, IMaxCellNumOfSamllReg)

C++ HRESULT pMsiToVector-> GetMsiToVectParam2 ( IRasterArea\* pRaster, BSTR \*reMsiName, BSTR \*regName, long \*lBandNo,long \*IMaxCellNumOfSamllReg);

参数说明：

参数	类型	出口/入口	描述
pRaster	IRasterArea	[in]	转换的图象文件指针
ReMsiName	String	[in,out]	合并小区后生成的图象名称
RegName	String	[in,out]	转换生成区文件名
lBandNo	Long	[out]	转换的波段号
IMaxCellNumOfSamllReg	Long	[out]	进行小区合并的时候的小区象元数

### MergSmallRegion 方法

描述 合并小区

Visual Basic MsiToVector.MergSmallRegion (  
msiName, reMsiName, lBandNo, lMaxCellNumOfSamllReg)  
C++ HRESULT pMsiToVector-> MergSmallRegion (BSTR msiName,  
BSTR reMsiName, long lBandNo,long lMaxCellNumOfSamllReg);

参数说明：

参数	类型	出口/入口	描述
msiName	String	[in]	转换的图象文件名称
reMsiName	String	[in]	合并小区后生成的图象名称
lBandNo	Long	[in]	转换的波段号
lMaxCellNumOfSamllReg	Long	[in]	进行小区合并的时候的小区 象元数

#### MergSmallRegion2 方法

描述 合并小区

Visual Basic MsiToVector.MergSmallRegion2  
pRaster, reMsiName, lBandNo, lMaxCellNumOfSamllReg  
C++ HRESULT pMsiToVector-> MergSmallRegion2 (IRasterArea \*pRaster,  
BSTR reMsiName, long lBandNo,long lMaxCellNumOfSamllReg);

参数说明：

参数	类型	出口/入口	描述
pRaster	IRasterArea	[in]	转换的图象文件指针
reMsiName	String	[in]	合并小区后生成的图象名称
lBandNo	Long	[in]	转换的波段号
lMaxCellNumOfSamllReg	Long	[in]	进行小区合并的时候的小区 象元数

#### GetMergeParam 方法

描述 得到合并小区的参数

Visual Basic MsiToVector.GetMergeParam(  
pRaster, reMsiName, lBandNo, lMaxCellNumOfSamllReg)  
C++ HRESULT pMsiToVector-> GetMergeParam(IRasterArea\* pRaster,  
BSTR \*reMsiName, long \*lBandNo,long \*lMaxCellNumOfSamllReg);

参数说明：

参数	类型	出口/入口	描述
pRaster	IRasterArea	[in,out]	转换的图象文件指针
reMsiName	String	[in,out]	合并小区后生成的图象名称
lBandNo	Long	[out]	转换的波段号

lMaxCellNumOfSamllReg	Long	[out]	进行小区合并的时候的小区象元数
-----------------------	------	-------	-----------------

### GetMergeParam 方法

描述 得到合并小区的参数

Visual Basic MsiToVector.GetMergeParam(  
msiName, reMsiName, lBandNo, lMaxCellNumOfSamllReg)

C++ HRESULT pMsiToVector-> GetMergeParam(BSTR \*msiName,  
BSTR \*reMsiName, long \*lBandNo,long \*lMaxCellNumOfSamllReg);

参数说明：

参数	类型	出口/入口	描述
MsiName	String	[in,out]	转换的图象文件名称
reMsiName	String	[in,out]	合并小区后生成的图象名称
lBandNo	Long	[out]	转换的波段号
lMaxCellNumOfSamllReg	Long	[out]	进行小区合并的时候的小区象元数

## 第八章 DBS 对象说明

DBS 对象的使用示例代码包含在 D:\MAPGIS61\VBDemo\VBForDBS 目录下。

### 8.1、DBS\_FramInfo 对象

**描述** DBS\_FramInfo 对象是底图图幅描述信息结构。

#### 8.1.1、属性：

属性	描述
mapMark	图幅标识
String	
可读写	
inputer	图幅文件的输入者
String	
可读写	
editor	图幅文件的编辑者
String	
可读写	
adjuster	图幅文件的校正者
String	
可读写	
note	图幅文件的注释
String	
可读写	
sysSaveNull	系统保留区
String	
可读写	

#### 8.1.2、方法：

##### Set 方法

**描述** 以传入的底图图幅描述信息集合替代原集合。

Visual Basic    rtl = DBS\_FramInfo.Set(ptNewVal)  
C++             HRESULT pDBS\_FramInfo-> Set(  
                 IDBS\_FramInfo \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	DBS_FramInfo	[in]	DBS_FramInfo 对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

## 8.2、MSystemTime 对象

描述      MSystemTime 对象是系统时间结构。

### 8.2.1、属性：

属性	描述
year	年
Integer 可读写	
month	月
Integer 可读写	
day	日
Integer 可读写	
dayofweek	星期
Integer 可读写	
hour	时
Integer 可读写	
minute	分
Integer 可读写	
second	秒
Integer 可读写	

milliseconds	百分秒
Integer	
可读写	

### 8.2.2、方法：无

## 8.3、DBS\_FramFile 对象

**描述** DBS\_FramFile 对象是底图单文件存取用结构。

### 8.3.1、属性：

属性	描述
filename	文件名(不含路径，不支持长于 31 个字符的文件名)
String	
可读写	
exist	文件存在标志。0/1：不存在/存在
Integer	
可读写	
sysSaveNull	系统保留区
String	
可读写	
classNo	文件相关的层类号
Integer	
可读写	
dtInDb	入库日期和时间
MsystemTime	
可读写	

### 8.3.2、方法：无

## 8.4、DBS\_FramFileSet 对象

**描述** DBS\_FramFileSet 对象是底图单文件集合。

### 8.4.1、属性：

属性	描述
count	底图单文件集合当前文件数
Long	
只读	
item(index)	第 index 个 DBS_FramFile 文件,index 为 Long 型
DBS_FramFile	
只读	

### 8.4.2、方法：

#### Append1 方法

**描述** 添加一个 DBS\_FramFile 文件

**Visual Basic** DBS\_FramFileSet.Append1 DbsFram

**C++** HRESULT pDBS\_FramFileSet-> Append1(IDBS\_FramFileSet\*pIDbsFram);

**参数说明：**

参数	类型	出口/入口	描述
DbsFram	DBS_FramFile	[in]	添加的 DBS_FramFile 文件

#### Append2 方法

**描述** 添加一个 DBS\_FramFile 文件

**Visual Basic** DBS\_FramFileSet.Append2 fileName, exist, sysSaveNull, classNo, dtInDbs

**C++** HRESULT pDBS\_FramFileSet-> Append2(BSTR fileName, short exist, BSTR sysSaveNull, short classNo, IMSystemTime\*pdtInDbs);

**参数说明：**

参数	类型	出口/入口	描述
fileName	String	[in]	文件名(不含路径,不支持长于 31 个字符的文件名)
exist	Integer	[in]	文件存在标志。0/1：不存在/存在
sysSaveNull	String	[in]	系统保留区



classNo	Integer	[in]	文件相关的层类号
pdtInDbcs	MsystemTime	[in]	入库日期和时间

**Insert1 方法**

**描述** 在指定位置插入一个 DBS\_FramFile 文件  
如果指定的位置>=集中的文件数，则添加一个 DBS\_FramFile 文件。

Visual Basic DBS\_FramFileSet.Insert1 index, DbsFram

C++ HRESULT pDBS\_FramFileSet-> Insert1(  
long index, IDBS\_FramFileSet \*pIDbsFram);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
DbsFram	DBS_FramFile	[in]	插入的 DBS_FramFile 文件

**Insert2 方法**

**描述** 在指定位置插入一个 DBS\_FramFile 文件  
如果指定的位置>=集中的文件数，则添加一个 DBS\_FramFile 文件。

Visual Basic DBS\_FramFileSet.Insert2

index, fileName, exist, sysSaveNull, classNo, dtInDbcs

C++ HRESULT pDBS\_FramFileSet-> Insert2(long index, BSTR filename,  
short exist, BSTR sysSaveNull, short classNo, IMSystemTime\*pdtInDbcs);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
fileName	String	[in]	文件名(不含路径,不支持长于 31 个字符的文件名)
exist	Integer	[in]	文件存在标志。0/1：不存在/存在
sysSaveNull	String	[in]	系统保留区
classNo	Integer	[in]	文件相关的层类号
pdtInDbcs	MsystemTime	[in]	入库日期和时间

**Remove 方法**

**描述** 删除集合中从 index 起 nCount 个 DBS\_FramFile 文件

Visual Basic DBS\_FramFileSet.Remove index, nCount

C++ HRESULT pDBS\_FramFileSet -> Remove(long index, long nCount);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

index	Long	[in]	集合的索引号
nCount	Long	[in]	删除的个数

**RemoveAll 方法**

描述 删除所有 DBS\_FramFile 文件

Visual Basic DBS\_FramFileSet.RemoveAll

C++ HRESULT pDBS\_FramFileSet-> RemoveAll();

参数说明： 无

**Set 方法**

描述 以传入的文件集合替代原集合

Visual Basic rtl = DBS\_FramFileSet.Set(ptNewVal)

C++ HRESULT pDBS\_FramFileSet->Set (IDBS\_FramFileSet \*ptNewVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptNewVal	DBS_FramFile	[in]	DBS_FramFile 对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Update1 方法**

描述 更新第 index 项数据

Visual Basic DBS\_FramFileSet.Update1 index, DbsFram

C++ HRESULT pDBS\_FramFileSet-> Update1(long index, IDBS\_FramFile\*pIDbsFram);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
DbsFram	DBS_FramFile	[in]	插入的 DBS_FramFile 文件

**Update2 方法**

描述 更新第 index 项数据

Visual Basic DBS\_FramFileSet.Update2 index, fileName, exist, sysSaveNull, classNo, dtInDbs

C++ HRESULT pDBS\_FramFileSet-> Update2(long index, BSTR fileName, short exist, BSTR sysSaveNull, short classNo, IMSystemTime\*pdInDbs);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

index	Long	[in]	集合的索引号
fileName	String	[in]	文件名(不含路径, 不支持长于 31 个字符的文件名)
exist	Integer	[in]	文件存在标志。0/1: 不存在/存在
sysSaveNull	String	[in]	系统保留区
classNo	Integer	[in]	文件相关的层类号
pdtInDb	MsystemTime	[in]	入库日期和时间

## 8.5、DBS\_GraIndex 对象

描述 DBS\_GraIndex 对象是底图图元表示用结构。

### 8.5.1、属性：

属性	描述
mapNdxNo	图幅索引号
Long	
可读写	
ClassNo	层类号
Integer	
可读写	
graphNo	图元号
Long	
可读写	

### 8.5.2、方法：无

## 8.6、DBS\_GraIndexSet 对象

描述 DBS\_GraIndexSet 对象是底图图元集合。

### 8.6.1、属性：

属性	描述
----	----

count	底图图元集合当前图元数
Long	
可读写	
item(index)	第 index 个 DBS_GraIndex 图元,index 为 Long 型
DBS_GraIndex	
只读	

## 8.6.2、方法：

### Append1 方法

描述 添加一个 DBS\_GraIndex 图元

Visual Basic DBS\_GraIndexSet.Append1 DbsGra

C++ HRESULT pIDBS\_GraIndexSet-> Append1(IDBS\_GraIndex\*pIDbsGra);

参数说明：

参数	类型	出口/入口	描述
DbsGra	DBS_GraIndex	[in]	添加的 DBS_GraIndex 图元

### Append2 方法

描述 添加一个 DBS\_GraIndex 图元

Visual Basic DBS\_GraIndexSet.Append2 mapNdxNo, classNo, graphNo

C++ HRESULT pDBS\_GraIndexSet-> Append2 (long mapNdxNo, short classNo, long graphNo);

参数说明：

参数	类型	出口/入口	描述
mapNdxNo	Long	[in]	图幅索引号
classNo	Integer	[in]	层类号
graphNo	Long	[in]	图元号

### Insert1 方法

描述 在指定位置插入一个 DBS\_GraIndex 图元

如果指定的位置>=集合中的图元数，则添加一个 DBS\_GraIndex 图元。

Visual Basic DBS\_GraIndexSet.Insert1 index, DbsGra

C++ HRESULT pDBS\_GraIndexSet-> Insert1 (long index, IDBS\_GraIndexSet\*pIDbsGra );

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

index	Long	[in]	集合的索引号
DbsGra	DBS_GraIndex	[in]	插入的 DBS_GraIndex 图元

**Insert2 方法**

**描述** 在指定位置插入一个 DBS\_GraIndex 图元  
如果指定的位置>=集合中的图元数，则添加一个 DBS\_GraIndex 图元。

Visual Basic DBS\_GraIndexSet.Insert2 mapNdxNo, classNo, graphNo

C++ HRESULT pDBS\_GraIndexSet-> Insert2(  
long mapNdxNo, short classNo, long graphNo);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
mapNdxNo	Long	[in]	图幅索引号
classNo	Integer	[in]	层类号
graphNo	Long	[in]	图元号

**Remove 方法**

**描述** 删除集合中从 index 起 nCount 个 DBS\_GraIndex 图元

Visual Basic DBS\_GraIndexSet.Remove index, nCount

C++ HRESULT pDBS\_GraIndexSet-> Remove(long index, long nCount);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
nCount	Long	[in]	删除的个数

**RemoveAll 方法**

**描述** 删除所有 DBS\_GraIndex 图元

Visual Basic DBS\_GraIndexSet.RemoveAll

C++ HRESULT pDBS\_GraIndexSet-> RemoveAll( );

**参数说明：** 无

**Set 方法**

**描述** 以传入的文件集合替代原集合

Visual Basic rtl = DBS\_GraIndexSet.Set(ptNewVal)

C++ HRESULT pDBS\_GraIndexSet-> Set (  
IDBS\_GraIndexSet \*ptNewVal, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
ptNewVal	DBS_GraIndex	[in]	DBS_GraIndex 对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Update1 方法**

描述 更新第 index 项数据

Visual Basic DBS\_GraIndex Set.Update1 index, DbsGra

C++ HRESULT pDBS\_GraIndexSet-> Update1(  
long index, IDBS\_GraIndex \*pIDbsGra);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
DbsGra	DBS_GraIndex	[in]	插入的 DBS_DbsGra 图元

**Update2 方法**

描述 更新第 index 项数据

Visual Basic DBS\_GraIndexSet.Update2 index, mapNdxNo, classNo, graphNo

C++ HRESULT DBS\_GraIndexSet-> Update2 (  
long mapNdxNo, short classNo, long graphNo);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号
mapNdxNo	Long	[in]	图幅索引号
classNo	Integer	[in]	层类号
graphNo	Long	[in]	图元号

**8.7、DBSArea 对象**

描述 DBSArea 对象封装了对图库工作区的操作。

**8.7.1、属性：**

属性	描述
Type	工作区类型
Enum_Area_Type	
只读	

Name	工作区名称
String	
可读写	
Changed	工作区已经改变标志
Boolean	
可读写	
ShowErrMsg	显示错误信息标志
Boolean	
可读写	
Warning	对不能后悔的操作是否给出警告提示
Boolean	
可读写	
Compact	保存工作区时压缩数据标志
Boolean	
可读写	
Empty	工作区是否为空标志
Boolean	
只读	
LastErrCode	最后错误码
Long	
只读	
rect	图库范围
D_Rect	
只读	

### 8.7.2、方法：

#### dsPackSaveFILEAs 方法

描述 压缩保存地图库为用户指定的文件。

Visual Basic rtl = DBSArea.dsPackSaveFILEAs()

C++ HRESULT pDBSArea-> dsPackSaveFILEAs();

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dsGetDBSIdxRegNumb 方法

描述 获取图库图幅(索引区)逻辑、物理数。

Visual Basic `rtl = DBSArea.dsGetDBSIIdxRegNumb(lgN, pyN)`

C++ `HRESULT pDBSArea-> dsGetDBSIIdxRegNumb(  
long *lgN,long *pyN, VARIANT_BOOL *rtl);`

参数说明：

参数	类型	出口/入口	描述
lgN	Long	[out]	图幅逻辑个数
pyN	Long	[out]	图幅物理个数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### dsGetDBSIndexRegAi 方法

描述 获取图库图幅索引工作区。

Visual Basic `Set pRArea = DBSArea.dsGetDBSIndexRegAi ()`

C++ `HRESULT pDBSArea-> dsGetDBSIndexRegAi(IRegArea **pRArea);`

参数说明：

参数	类型	出口/入口	描述
pRArea	RegArea	[out]	返回区工作区

### dsImportIdxToDBSArea 方法

描述 导入地图库索引区数据。

Visual Basic `rtl = DBSArea.dsImportIdxToDBSArea ([fName])`

C++ `HRESULT pDBSArea->dsImportIdxToDBSArea(  
BSTR fName = NULL,VARIANT_BOOL *rtl);`

参数说明：

参数	类型	出口/入口	描述
fName	String	[in]	传入的索引区文件名 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### dsExportIdxToREGArea 方法

描述 导出地图库索引区数据。

Visual Basic `rtl = DBSArea.dsExportIdxToREGArea([fName])`

C++ `HRESULT pDBSArea-> dsExportIdxToREGArea(  
BSTR fName = NULL,VARIANT_BOOL *rtl);`

参数说明：

参数	类型	出口/入口	描述
fName	String	[in]	传入的索引区文件名 可选；缺省值：Nothing



rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

**dsExportDBSJTBToPrj 方法**

描述 输出图库接图表工程制图文件。

Visual Basic rtl = DBSArea.dsExportDBSJTBToPrj([fName])

C++ HRESULT pDBSArea-> dsExportDBSJTBToPrj(  
BSTR fName = NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fName	String	[in]	输出的索引区文件名 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dsDBSClazzManger 方法**

描述 图库层类维护管理器。

Visual Basic rtl = DBSArea.dsDBSClazzManger()

C++ HRESULT pDBSArea-> dsDBSClazzManger(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dsDBSClazzViewer 方法**

描述 库层类浏览器。

Visual Basic rtl = DBSArea.dsDBSClazzViewer()

C++ HRESULT pDBSArea-> dsDBSClazzViewer(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dsGetDBSClazzNumb 方法**

描述 获取图库当前存在的层类数。

Visual Basic rtl = DBSArea.dsGetDBSClazzNumb(nCNumb)

C++ HRESULT pDBSArea-> dsGetDBSClazzNumb(  
short \*nCNumb, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
nCNumb	Integer	[out]	层类个数

rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

**dsGetDBSClassName 方法**

描述 获取指定类的层类名称。

Visual Basic clsName = DBSArea.dsGetDBSClassName(index)

C++ HRESULT pDBSArea-> dsGetDBSClassName(short index,BSTR \*clsName);

参数说明：

参数	类型	出口/入口	描述
index	Integer	[in]	层类索引号
clsName	String	[out]	成功返回图库中第 index 层的层类名，失败返回 Nothing

**dsGetDBSClassOnOff 方法**

描述 获取指定类的手动显示开关标志。

Visual Basic flag = DBSArea.dsGetDBSClassOnOff(index)

C++ HRESULT pDBSArea-> dsGetDBSClassOnOff(short index,short \*flag);

参数说明：

参数	类型	出口/入口	描述
index	Integer	[in]	层类索引号
flag	Integer	[out]	成功则返回图库中第 index 层的显示开关标志。 0/1：不显示/显示

**dsSetDBSClassOnOff 方法**

描述 设置图库指定层类的手动显示开关标志。

Visual Basic Flg = DBSArea.dsSetDBSClassOnOff(index, flag)

C++ HRESULT pDBSArea-> dsSetDBSClassOnOff(short index,short flag,short \*Flg);

参数说明：

参数	类型	出口/入口	描述
index	Integer	[in]	层类索引号
flag	Integer	[in]	需设置的显示标志
Flg	Integer	[out]	成功则返回图库中第 index 层被设置前显示开关标志。 0/1：不显示/显示

**dsDBSClassCanDisp 方法**

**描述** 检测图库指定层类的动态显示开关(受比例调整以实现变焦显示)。

**变焦显示：** 所有指定层类的图元在该层类的最大和最小显示范围之内显示。比如说该层类的最大显示比例和最小显示比例设置为 10 和 5。那么，在显示比例小于 5 时将不显示该层类的图元，只有大于 5 时将显示。到了系统的显示比例大于 10 时，该层类的图元将不显示。

Visual Basic flag = DBSArea.dsDBSClassCanDisp(index)

C++ HRESULT pDBSArea-> dsDBSClassCanDisp(short index,short \*flag);

**参数说明：**

参数	类型	出口/入口	描述
index	Integer	[in]	层类索引号
flag	Integer	[out]	成功则返回图库中第 index 层的动态显示开关标志。 0/1：没有打开/已经打开

**dsDBSRegClassOnlyDispLin 方法**

**描述** 获取图库指定面元层类的仅弧段显示标志。

Visual Basic IsODispLin = DBSArea.dsDBSRegClassOnlyDispLin(index)

C++ HRESULT pDBSArea-> dsDBSRegClassOnlyDispLin(  
short index,short\* IsODispLin);

**参数说明：**

参数	类型	出口/入口	描述
index	Integer	[in]	层类索引号
IsODispLin	Integer	[out]	成功则返回图库中第 index 层的仅弧段显示标志。 >0/0：仅显弧段/没有设置仅显弧段

**dsSETRegClassOnlyDispLin 方法**

**描述** 设置图库指定面元层类的仅弧段显示标志。

Visual Basic rtl = DBSArea.dsSETRegClassOnlyDispLin(index, IsODispLin)

C++ HRESULT pDBSArea-> dsSETRegClassOnlyDispLin(  
short index,short \*IsODispLin,VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
index	Integer	[in]	层类索引号
IsODispLin	Integer	[in]	是否仅显弧段标志
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dsGetDBSClassDataType 方法**

描述 获取图库指定层类的数据类型。

Visual Basic CType = DBSArea.dsGetDBSClassDataType(index)

C++ HRESULT pDBSArea-> dsGetDBSClassDataType(  
short index, Enum\_Dbs\_File\_Type \*CType);

参数说明：

参数	类型	出口/入口	描述
index	Integer	[in]	层类索引号
CType	Enum_Dbs_File_Type	[out]	图库中指定层类文件类型

**dsIsOneFileDBSClsLayer 方法**

描述 检测图库指定层是否为单文件专题图要素层。

Visual Basic rtl = DBSArea.dsIsOneFileDBSClsLayer(index)

C++ HRESULT pDBSArea-> dsIsOneFileDBSClsLayer(  
short index, VARIATN\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
index	Integer	[in]	层类索引号
rtl	Boolean	[out]	若是单文件专题图要素层返回 TRUE，不是则返回 FALSE

**dsGetDBSClassAttStru 方法**

描述 获取图库指定层类的属性结构。

Visual Basic Set pIRcodHd = DBSArea.dsGetDBSClassAttStru(index)

C++ HRESULT pDBSArea-> dsGetDBSClassAttStru(  
short index, IRecord\_Head \*\*pIRcodHd);

参数说明：

参数	类型	出口/入口	描述
index	Integer	[in]	层类索引号
pIRcodHd	Record_Head	[out]	图库指定层类的属性结构 (仅对点、线、区要素层)

**dsGetDBSClassDirPat 方法**

描述 获取图库指定层类数据所在目录路径。

Visual Basic cDirPath = DBSArea.dsGetDBSClassDirPath(index)

C++ HRESULT pDBSArea-> dsGetDBSClassDirPath(  
short index, BSTR \*cDirPath);

参数说明：

参数	类型	出口/入口	描述
index	Integer	[in]	层类索引号
cDirPath	String	[out]	图库指定层类的所在目录路径

#### dsModClassStateBYScale 方法

描述 根据比例调整层类动态(变焦)开关状态。

Visual Basic rtl = DBSArea.dsModClassStateBYScale(curScale)

C++ HRESULT pDBSArea-> dsModClassStateBYScale(  
double curScale, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
curScale	double	[in]	显示比例分母
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dsGetClassNoBYName 方法

描述 根据层类名称获取层类号。

Visual Basic clsNo = DBSArea.dsGetClassNoBYName(clsName)

C++ HRESULT pDBSArea-> dsGetClassNoBYName(  
BSTR clsName,short\*clsNo);

参数说明：

参数	类型	出口/入口	描述
clsName	String	[in]	层类名称
clsNo	Integer	[out]	成功则返回层类号

#### dsSelDBSCurClass 方法

描述 由对话框选择图库当前层类。

Visual Basic clsNo = DBSArea.dsSelDBSCurClass()

C++ HRESULT pDBSArea-> dsSelDBSCurClass (short\*clsNo);

参数说明：

参数	类型	出口/入口	描述
clsNo	Integer	[out]	成功则返回层类号

#### dsGetDBSMapParam 方法

描述 获取图库投影参数。

Visual Basic rtl = DBSArea.dsGetMapParam(pMapPara)

C++ HRESULT pDBSArea-> dsGetDBSMapParam(

---

```
IMap_Para **pMapPara, VARIANT_BOOL *rtl);
```

参数说明：

参数	类型	出口/入口	描述
pMapPara	Map_Para	[out]	获取的投影参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dsSetDBSMapParam 方法

描述 设置图库投影参数。

Visual Basic rtl = DBSArea.dsSetMapParam(pMapPara)

C++ HRESULT pDBSArea->dsSetDBSMapParam(  
IMap\_Para \*pMapPara, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pMapPara	Map_Para	[in]	设置的投影参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dsEditDBSMapParam 方法

描述 对话框编辑图库投影参数。

Visual Basic rtl = DBSArea.dsEditDBSMapParam()

C++ HRESULT pDBSArea->dsEditDBSMapParam(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dsViewDBSMapParam 方法

描述 对话框浏览图库投影参数。

Visual Basic rtl = DBSArea.dsViewDBSMapParam()

C++ HRESULT pDBSArea->dsViewDBSMapParam(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### dsSetDBSDispCordParam 方法

描述 设置图库坐标提示用投影参数、坐标单位。

Visual Basic rtl = DBSArea.dsSetDBSDispCordParam()

C++ HRESULT pDBSArea->dsSetDBSDispCordParam(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE
-----	---------	-------	----------------------

**dsCvtPntDISPCord 方法**

描述 将图库坐标转换为图库提示坐标。

Visual Basic rtl = DBSArea.dsCvtPntDISPCord(x, xLen, y, yLen, xx, yy)

C++ HRESULT pDBSArea-> dsCvtPntDISPCord(  
double \*x,short xLen,double\*y, short yLen,double \*xx,double \*yy);

参数说明：

参数	类型	出口/入口	描述
x	Double	[in]	图库坐标 x
xLen	Integer	[in]	坐标 x 长度
y	Double	[in]	图库坐标 y
yLen	Integer	[in]	坐标 y 长度
xx	Double	[out]	图库显示坐标 x
yy	Double	[out]	图库显示坐标 y
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dsGetPntDISPCordStr 方法**

描述 获取图库显示坐标提示串。

Visual Basic str = DBSArea.dsGetPntDISPCordStr(x, y)

C++ HRESULT pDBSArea-> dsGetPntDISPCordStr(double x,double y,BSTR\*str);

参数说明：

参数	类型	出口/入口	描述
x	Double	[in]	图库坐标 x
y	Double	[in]	图库坐标 y
str	String	[out]	成功则返回提示字符串

**dsGetDBSCoorUnitName 方法**

描述 获取图库数据坐标数据单位名。

Visual Basic uName = DBSArea.dsGetDBSCoorUnitName()

C++ HRESULT pDBSArea-> dsGetDBSCoorUnitName(BSTR \*uName);

参数说明：

参数	类型	出口/入口	描述
uName	String	[out]	成功则返回图库数据坐标数据单位名字串

**dsGetDBSCoorUnit 方法**

描述 获取图库数据坐标数据单位号。

Visual Basic uNo = DBSArea.dsGetDBSCoorUnit()

C++ HRESULT pDBSArea->dsGetDBSCoorUnit(Enum\_SpatialUnit\_Type\*uNo);

参数说明：

参数	类型	出口/入口	描述
uNo	Enum_SpatialUnit_Type	[out]	成功则返回图库数据坐标数据单位号

**dsGetDBSCoorDispUnitName 方法**

描述 获取图库数据坐标数据当前显示单位名。

Visual Basic uName = DBSArea.dsGetDBSCoorDispUnitName()

C++ HRESULT pDBSArea-> dsGetDBSCoorDispUnitName(BSTR\*uName);

参数说明：

参数	类型	出口/入口	描述
uName	String	[out]	成功则返回图库数据坐标数据当前显示单位名字符串

**dsGetDBSCoorDispUnit 方法**

描述 获取图库数据坐标数据当前显示单位号。

Visual Basic uNo = DBSArea.dsGetDBSCoorDispUnit()

C++ HRESULT pDBSArea-> dsGetDBSCoorDispUnit(Enum\_SpatialUnit\_Type \*uNo);

参数说明：

参数	类型	出口/入口	描述
uNo	Enum_SpatialUnit_Type	[out]	成功则返回图库数据坐标数据当前显示单位号

**dsDBSFramDatManger 方法**

描述 弹出图库图幅文件数据维护管理器，对图幅文件数据进行维护和管理。

Visual Basic rtl = DBSArea.dsDBSFramDatManger(lFramNo)

C++ HRESULT pDBSArea-> dsDBSFramDatManger(long lFramNo,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
lFramNo	Long	[in]	图库图幅号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE



**dsDBSFramDatViewer 方法**

**描述** 用来弹出图库图幅文件数据浏览器，对图幅文件数据进行浏览。

**Visual Basic** rtl = DBSArea.dsDBSFramDatViewer(lFramNo)

**C++** HRESULT pDBSArea-> dsDBSFramDatViewer(  
long lFramNo, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
lFramNo	Long	[in]	图库图幅号
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsDelDBSFramDat 方法**

**描述** 用来删除指定图幅文件数据。

**Visual Basic** rtl = DBSArea.dsDelDBSFramDat(lFramNo)

**C++** HRESULT pDBSArea-> dsDelDBSFramDat(  
long lFramNo, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
lFramNo	Long	[in]	图库图幅号
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsGetDBSFramDat 方法**

**描述** 获取图库图幅描述信息及层类文件表数据。

**Visual Basic** rtl = DBSArea.dsGetDBSFramDat(lFramNo, [fi], [ffs])

**C++** HRESULT pDBSArea-> dsGetDBSFramDat(  
long lFramNo, IDBS\_FramInfo \*\*fi = NULL,  
DBS\_FramFileSet \*\*ffs = NULL, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
lFramNo	Long	[in]	图库图幅号
fi	IDBS_FramInfo	[out]	获取的指定图幅描述信息
ffs	IDBS_FramFileSet	[out]	获取的指定图幅层类文件表数据
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsGetDBSFramInfo 方法**

**描述** 获取图幅描述信息函数。

**Visual Basic** rtl = DBSArea.dsGetDBSFramInfo(lFramNo,fi)

**C++** HRESULT pDBSArea-> dsGetDBSFramInfo(  
long lFramNo, IDBS\_FramInfo \*\*fi, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
lFramNo	Long	[in]	图库图幅号
fi	DBS_FramInfo	[out]	获取的指定图幅描述信息
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

#### dsGetDBSFramFileDat 方法

描述 获取图库图幅层类文件表数据。

Visual Basic rtl = DBSArea.dsGetDBSFramFileDat(lFramNo, ffs)

C++ HRESULT pDBSArea-> dsGetDBSFramFileDat(  
long lFramNo, IDBS\_FramFileSet \*\*ffs, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
lFramNo	Long	[in]	图库图幅号
ffs	DBS_FramFileSet	[out]	获取的指定图幅层类文件表数据
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

#### dsUpdateDBSFramDat 方法

描述 更新图库图幅描述信息及层类文件表数据。

Visual Basic rtl = DBSArea.dsUpdateDBSFramDat(lFramNo, [fi], [ffs])

C++ HRESULT pDBSArea-> dsUpdateDBSFramDat(  
long lFramNo, IDBS\_FramInfo \*fi = NULL,  
IDBS\_FramFileSet \*ffs = NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
lFramNo	Long	[in]	图库图幅号
fi	DBS_FramInfo	[in]	更新的指定图幅描述信息 可选；缺省值：Nothing
ffs	DBS_FramFileSet	[in]	更新的指定图幅层类文件表数据 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

#### dsUpdateDBSFramInfo 方法

描述 更新图幅描述信息。

Visual Basic rtl = DBSArea.dsUpdateDBSFramInfo(lFramNo, fi)

C++ HRESULT pDBSArea->dsUpdateDBSFramInfo(  
long lFramNo, IDBS\_FramInfo \*fi, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
lFramNo	Long	[in]	图库图幅号
fi	DBS_FramInfo	[out]	更新的指定图幅描述信息
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsUpdateDBSFramFileDat 方法**

描述 更新图幅层类文件表数据。

Visual Basic rtl = DBSArea.dsUpdateDBSFramFileDat(lFramNo, ffs)

C++ HRESULT pDBSArea-> dsUpdateDBSFramFileDat(  
long lFramNo, IDBS\_FramFileSet \*ffs,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
lFrameNo	Long	[in]	图库图幅号
ffs	DBS_FramFileSet	[in]	更新的指定图幅层类文件表数据
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsGetMapDBSfRcClip 方法**

描述 用来从所有已打开的层类中提取在指定矩形区域内的图元数据及其属性。

Visual Basic rtl = DBSArea.dsGetMapDBSfRcClip(  
prjHand, pDRect, [bAllFrmOut], [bMegr ], [dfErrRad])

C++ HRESULT pDBSArea-> dsGetMapDBSfRcClip(ID\_Rect\*pDRect,  
VARIANT\_BOOL bAllFrmOut = FALSE, VARIANT\_BOOL bMegr=FALSE,  
double dfErrRad = 0.0, IMap\*\* ppMap, short \*rtl);

参数说明：

参数	类型	出口/入口	描述
pDRect	D_Rect	[in]	矩形区域
bAllFrmOut	Boolean	[in]	指定区域内的图幅数据是否全部剪出 可选；缺省值：FALSE
bMegr	Boolean	[in]	跨图幅图元是否合并标志 可选；缺省值：FALSE
dfErrRad	Double	[in]	跨图幅图元合并时的限差 可选；缺省值：0.0
ppMap	IMap	[in,out]	Map 对象
rtl	Integer	[out]	0/1:局部错 / 成功

### dsRectAskDBSClsDat 方法

**描述** 在给定区域内提取当前指定点、线、区层类的数据到工作区中并浏览其属性。若当前指定的是线或区类数据，则可由 IsMerg 指定跨图幅图元是否归并。

Visual Basic Set pWArea = DBSArea.dsRectAskDBSClsDat(clsNo, pDRect, IsMerg)

C++ HRESULT pDBSArea-> dsRectAskDBSClsDat(  
short clsNo, ID\_Rect \*pDRect, short IsMerg = 0, IWorkArea \*\*pWArea);

参数说明：

参数	类型	出口/入口	描述
clsNo	Integer	[in]	层类索引号
pDRect	D_Rect	[in]	矩形区域
IsMerg	Integer	[in]	跨图幅图元归并标志 0/1：不归并/归并 可选；缺省值：0
pWArea	WorkArea	[out]	结果工作区

### dsAttAskGraDataToArea 方法

**描述** 对指定层类在给定区域内设定属性条件提取图形数据到工作区中。

Visual Basic Set pWArea = DBSArea.dsAttAskGraDataToArea(clsNo, pDRect, IsMerg)

C++ HRESULT pDBSArea-> dsAttAskGraDataToArea(  
short clsNo, ID\_Rect \*pDRect, short IsMerg = 0, IWorkArea \*\*pWArea);

参数说明：

参数	类型	出口/入口	描述
clsNo	Integer	[in]	层类索引号
pDRect	D_Rect	[in]	矩形区域
IsMerg	Integer	[in]	跨图幅图元归并标志 0/1：不归并/归并 可选；缺省值：0
pWArea	WorkArea	[out]	结果工作区

### dsRgnAttAskDBSClsDat 方法

**描述** 在给定区域按一定属性条件提取指定层类数据到工作区中。

Visual Basic Set pWArea = DBSArea.dsRgnAttAskDBSClsDat (clsNo, pDRect, IsMerg)

C++ HRESULT pDBSArea-> dsRgnAttAskDBSClsDat (  
short clsNo, ID\_Rect \*pDRect, short IsMerg = 0, IWorkArea \*\*pWArea);

参数说明：

参数	类型	出口/入口	描述
clsNo	Integer	[in]	层类索引号

pDDotSet	D_DotSet	[in]	区域边界坐标点集对象
IsMerg	Integer	[in]	跨图幅图元归并标志 0/1：不归并/归并 可选；缺省值：0
pWArea	WorkArea	[out]	结果工作区

**dsSeekDBSIIdxReg 方法**

描述 获取指定点所在图库图幅号(索引区序号)。

Visual Basic lFramNo = DBSArea.dsSeekDBSIIdxReg(pDDot)

C++ HRESULT pDBSArea-> dsSeekDBSIIdxReg (ID\_Dot\*pDDot, long\*lFramNo);

参数说明：

参数	类型	出口/入口	描述
pDDot	D_Dot	[in]	点坐标
lFramNo	Long	[out]	成功则返回图库图幅号

**dsNearDBSIndexRegArc 方法**

描述 获取离指定点所在位置最近的弧段号。

Visual Basic lNo = DBSArea.dsNearDBSIndexRegArc (pDDot, [dis], [dislen])

C++ HRESULT pDBSArea-> dsNearDBSIndexRegArc (  
ID\_Dot \*pDDot, double \*dis = NULL, long dislen = 0, long \*lNo);

参数说明：

参数	类型	出口/入口	描述
pDDot	D_Dot	[in]	点坐标
dis	Double	[in]	查找半径 可选；缺省值：Nothing (即在系统缺省范围查找)
disLen	Long	[in]	半径长度 可选；缺省值：0
lNo	Long	[out]	成功则返回最近弧段号

**dsGetDBSFrmNoByMark 方法**

描述 根据给定的图幅名获得相应图幅号。

Visual Basic lFramNo = DBSArea.dsGetDBSFrmNoByMark(framMark)

C++ HRESULT pDBSArea-> dsGetDBSFrmNoByMark (  
BSTR framMark, long \*lFramNo);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

framMark	String	[in]	图幅名(大小写不敏感)
lFramNo	Long	[out]	成功则返回图库图幅号

**dsGetDBSFrmNoByMark 方法**

描述 获取指定图幅的允许图形范围。

Visual Basic rtl = DBSArea.dsGetDBSFrmRect(lFramNo,pDRect)

C++ HRESULT pDBSArea-> dsGetDBSFrmRect (  
long lFramNo,ID\_Rect \*pDRect,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
lFramNo	Long	[in]	图库图幅号
pDRect	D_Rect	[out]	图幅范围
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsRectAskDBSFrmToList 方法**

描述 选取与给定区域相交的图幅编号到列表中。

Visual Basic Set framLst = DBSArea.dsRectAskDBSFrmToList(pD\_Rect)

C++ HRESULT pDBSArea-> dsRectAskDBSFrmToList (  
ID\_Rect \*pDRect, ILONGList \*\*framLst);

参数说明：

参数	类型	出口/入口	描述
pDRect	D_Rect	[in]	矩形区域
framLst	LONGList	[out]	图幅编号列表

**dsFreeDBSFrmSelLst 方法**

描述 释放由 dsRectAskDBSFrmToList 方法产生的图库图幅选择表。

Visual Basic framLst = DBSArea.dsFreeDBSFrmSelLst()

C++ HRESULT pDBSArea-> dsFreeDBSFrmSelLs (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsSelDBSFramByDlg 方法**

描述 通过对话框选择单个图幅。

Visual Basic lFramNo = DBSArea.dsSelDBSFramByDlg()

C++ HRESULT pDBSArea-> dsSelDBSFramByDlg (long \*lFramNo);

参数说明：

参数	类型	出口/入口	描述
lFramNo	Long	[out]	图库图幅号

**dsSelDBSFramsByDlg 方法**

描述 通过对话框选择多个图幅。

Visual Basic Set framLst = DBSArea.dsSelDBSFramsByDlg()

C++ HRESULT pDBSArea-> dsSelDBSFramsByDlg (ILONGList \*\*framLst);

参数说明：

参数	类型	出口/入口	描述
framLst	LONGList	[out]	图幅编号列表

**dsFreeFramLstMem 方法**

描述 释放由 dsSelDBSFramsByDlg 方法产生的图库图幅选择表。

Visual Basic rtl = DBSArea.dsFreeFramLstMem()

C++ HRESULT pDBSArea-> dsFreeFramLstMem (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
<i>rtl</i>	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsFramLstClipProc 方法**

描述 根据给定图幅列表提取图库数据处理。

Visual Basic rtl = DBSArea.dsFramLstClipProc(framLst)

C++ HRESULT pDBSArea-> dsFramLstClipProc (ILONGList \*framLst, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
framLst	LONGList	[in]	图幅编号列表
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsNearDBSGraIdx 方法**

描述 在指定层中查找离指定点最近的图元。

Visual Basic rtl = DBSArea.dsNearDBSGraIdx(clsNo, IsArc, pDDot, pGIdx, dis)

C++ HRESULT pDBSArea-> dsNearDBSGraIdx(short clsNo, short IsArc, ID\_Dot \*pDDot, IDBS\_GraIndex \*\*pGIdx, double \*dis = NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
clsNo	Integer	[in]	层类索引号
IsArc	Integer	[in]	仅显弧段标志
pDDot	D_Dot	[in]	点坐标
pGIdx	DBS_GraIndex	[out]	查询结果
dis	Double	[out]	指定点到结果图元的距离
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsNearDBSPnt 方法**

**描述** 在指定层中查找离指定点最近的点图元。

Visual Basic rtl = DBSArea.dsNearDBSPnt(clsNo,pDDot,DBS\_GraIndex,dis)

C++ HRESULT pDBSArea-> dsNearDBSPnt(short clsNo,  
ID\_Dot \*pDDot,IDBS\_GraIndex \*\*pGIdx,  
double \*dis = NULL, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
clsNo	Integer	[in]	层类索引号
pDDot	D_Dot	[in]	点坐标
pGIdx	DBS_GraIndex	[out]	结果点图元
dis	Double	[out]	指定点到结果点图元的距离
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsNearDBSLin 方法**

**描述** 在指定层中查找离指定点最近的线图元。

Visual Basic rtl = DBSArea.dsNearDBSLin (clsNo, pDDot, DBS\_GraIndex, dis)

C++ HRESULT pDBSArea-> dsNearDBSLin (short clsNo,  
ID\_Dot \*pDDot,IDBS\_GraIndex \*\*pGIdx,  
double \*dis = NULL, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
clsNo	Integer	[in]	层类索引号
pDDot	D_Dot	[in]	点坐标
pGIdx	DBS_GraIndex	[out]	结果线图元
dis	Double	[out]	指定点到结果线图元的距离
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE



**dsSeekDBSReg 方法**

描述 在指定层中查找离指定点最近的区图元。

Visual Basic rtl = DBSArea.dsSeekDBSReg (clsNo, pDDot, DBS\_GraIndex)

C++ HRESULT pDBSArea-> dsSeekDBSReg (short clsNo,  
ID\_Dot \*pDDot, IDBS\_GraIndex \*\*pGIdx, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
clsNo	Integer	[in]	层类索引号
pDDot	D_Dot	[in]	点坐标
pGIdx	DBS_GraIndex	[out]	结果区图元
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsRectAskDBSPntToList 方法**

描述 在指定矩形区域查询点图元。

Visual Basic Set pGIdxSet = DBSArea.dsRectAskDBSPntToList(pDRect)

C++ HRESULT pDBSArea-> dsRectAskDBSPntToList (  
ID\_Rect \*pDRect, IDBS\_GraIndexSet \*\*pGIdxSet);

参数说明：

参数	类型	出口/入口	描述
pDRect	D_Rect	[in]	矩形区域
pGIdxSet	DBS_GraIndexSet	[out]	成功则返回结果点图元

**dsRectAskDBSLinToList 方法**

描述 在指定矩形区域查询线图元。

Visual Basic Set pGIdxSet = DBSArea.dsRectAskDBSLinToList(pDRect)

C++ HRESULT pDBSArea-> dsRectAskDBSLinToList (  
ID\_Rect \*pDRect, IDBS\_GraIndexSet \*\*pGIdxSet);

参数说明：

参数	类型	出口/入口	描述
pDRect	D_Rect	[in]	矩形区域
pGIdxSet	DBS_GraIndexSet	[out]	成功则返回结果线图元

**dsRectAskDBSRegToList 方法**

描述 在指定矩形区域查询区图元。

Visual Basic Set pGIdxSet = DBSArea.dsRectAskDBSRegToList (pDRect)

C++ HRESULT pDBSArea-> dsRectAskDBSRegToList (  
ID\_Rect \*pDRect, IDBS\_GraIndexSet \*\*pGIdxSet);

参数说明：

参数	类型	出口/入口	描述
pDRect	D_Rect	[in]	矩形区域
pGIdxSet	DBS_GraIndexSet	[out]	成功则返回结果区图元

#### dsFreeDBSPntSelList 方法

描述 释放由 dsRectAskDBSPntToList 方法选择的图库点图元表内存。

Visual Basic rtl = DBSArea.dsFreeDBSPntSelList()

C++ HRESULT pDBSArea-> dsFreeDBSPntSelList(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

#### dsFreeDBSLinSelList 方法

描述 释放由 dsRectAskDBSLinToList 方法选择的图库线图元表内存。

Visual Basic rtl = DBSArea.dsFreeDBSLinSelList()

C++ HRESULT pDBSArea-> dsFreeDBSLinSelList(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

#### dsFreeDBSRegSelList 方法

描述 释放由 dsRectAskDBSRegToList 方法选择的图库区图元表内存。

Visual Basic rtl = DBSArea.dsFreeDBSRegSelList()

C++ HRESULT pDBSArea-> dsFreeDBSRegSelList(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

#### dsGetDBSPnt 方法

描述 取图库点图元数据。

Visual Basic rtl = DBSArea.dsGetDBSPnt(pGIdx, pDDot, data, pPInf)

C++ HRESULT pDBSArea-> dsGetDBSPnt(IDBS\_GraIndex \*pGIdx, ID\_Dot \*\*pDDot, BSTR \*data, IPnt\_Info \*\*pPInf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待查询点图元

pDDot	D_Dot	[out]	点坐标
data	String	[out]	点信息
pPInf	Pnt_Info	[out]	点图形信息
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsGetDBSLin 方法**

描述 取图库线图元数据。

Visual Basic rtl = DBSArea.dsGetDBSLin(pGIdx, pDDotSet, pLInf)

C++ HRESULT pDBSArea-> dsGetDBSLin(IDBS\_GraIndex \*pGIdx,  
ID\_Dot \*\*pDDot, ILin\_Info \*\*pLInf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待查询线图元
pDDotSet	D_DotSet	[out]	线点坐标
pLInf	Lin_Info	[out]	线图形信息
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsGetDBSReg 方法**

描述 取图库区图元数据。

Visual Basic rtl = DBSArea.dsGetDBSReg(pGIdx, pRdat, pRInf)

C++ HRESULT pDBSArea-> dsGetDBSReg (IDBS\_GraIndex \*pGIdx,  
ILONGList \*\*pRdat, IReg\_Info \*\*pRInf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待查询区图元
pRdat	LONGList	[out]	区弧段号集合
pRInf	Reg_Info	[out]	区图形信息
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsGetDBSPntAtt 方法**

描述 取图库点图元属性信息。

Visual Basic rtl = DBSArea.dsGetDBSPntAtt(pGIdx, att)

C++ HRESULT pDBSArea-> dsGetDBSPntAtt (  
IDBS\_GraIndex \*pGIdx, IRecord \*\*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

pGIdx	DBS_GraIndex	[in]	待查询点图元
att	Record	[out]	属性记录值
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsGetDBSLinAtt 方法**

描述 取图库线图元属性信息。

Visual Basic rtl = DBSArea.dsGetDBSLinAtt(pGIdx, att)

C++ HRESULT pDBSArea-> dsGetDBSLinAtt (  
IDBS\_GraIndex \*pGIdx, IRecord \*\*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待查询线图元
att	Record	[out]	属性记录值
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsGetDBSRegAtt 方法**

描述 取图库区图元属性信息。

Visual Basic rtl = DBSArea.dsGetDBSRegAtt(pGIdx, att)

C++ HRESULT pDBSArea-> dsGetDBSRegAtt (  
IDBS\_GraIndex \*pGIdx, IRecord \*\*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待查询区图元
att	Record	[out]	属性记录值
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsViewDBSPntAtt 方法**

描述 浏览图库点图元属性信息。

Visual Basic rtl = DBSArea.dsViewDBSPntAtt (pGIdx)

C++ HRESULT pDBSArea-> dsViewDBSPntAtt (  
IDBS\_GraIndex \*pGIdx, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待浏览点图元
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsViewDBSLinAtt 方法**

描述 浏览图库线图元属性信息。

Visual Basic rtl = DBSArea.dsViewDBSLinAtt (pGIdx)

C++ HRESULT pDBSArea-> dsViewDBSLinAtt (  
IDBS\_GraIndex \*pGIdx, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	IDBS_GraIndex	[in]	待浏览线图元
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsViewDBSRegAtt 方法**

描述 浏览图库区图元属性信息。

Visual Basic rtl = DBSArea.dsViewDBSRegAtt (pGIdx)

C++ HRESULT pDBSArea-> dsViewDBSRegAtt (  
IDBS\_GraIndex \*pGIdx, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	IDBS_GraIndex	[in]	待浏览区图元
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsGetDBSEntireLinAI 方法**

描述 获取图库选定的跨图幅完整线到工作区中。

Visual Basic Set pLArea = DBSArea.dsGetDBSEntireLinAI(clsNo, pDDot)

C++ HRESULT pDBSArea-> dsGetDBSEntireLinAI (  
short clsNo, ID\_Dot \*pDDot, ILinArea \*\*pLArea);

参数说明：

参数	类型	出口/入口	描述
clsNo	Integer	[in]	层类索引号 可传入-1 以表示未指定层类
pDDot	ID_Dot	[in]	点坐标
pLArea	ILinArea	[out]	结果线工作区

**dsGetDBSEntireRegAI 方法**

描述 获取图库选定的跨图幅完整区到工作区中。

Visual Basic Set pRArea = DBSArea.dsGetDBSEntireRegAI(clsNo, pDDot)

C++ HRESULT pDBSArea-> dsGetDBSEntireRegAI (  
short clsNo, ID\_Dot \*pDDot, IRegArea \*\*pRArea);

参数说明：

参数	类型	出口/入口	描述
clsNo	Integer	[in]	层类索引号 可传入-1 以表示未指定层类
pDDot	D_Dot	[in]	点坐标
pRArea	RegArea	[out]	结果区工作区

#### dsViewDBSGraLayerName 方法

描述 浏览图库指定图元的层类信息。

Visual Basic DBSArea.dsViewDBSGraLayerName pGIdx

C++ HRESULT pDBSArea-> dsViewDBSGraLayerName ( IDBS\_GraIndex \*pGIdx);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待浏览图元

#### dsViewDBSInfo 方法

描述 浏览图库信息。

Visual Basic DBSArea.dsViewDBSInfo

C++ HRESULT pDBSArea-> dsViewDBSInfo( );

参数说明： 无

#### dsOutDBSDocDat 方法

描述 输出图库文档数据。

Visual Basic rtl = DBSArea.dsOutDBSDocDat()

C++ HRESULT pDBSArea-> dsOutDBSDocDat(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

#### dsUpdateDBSPnt 方法

描述 更新图库指定点图元。

Visual Basic rtl = DBSArea.dsUpdateDBSPnt(pGIdx, pDDot, data, pPInf)

C++ HRESULT pDBSArea-> dsUpdateDBSPnt (IDBS\_GraIndex \*pGIdx, ID\_Dot \*pDDot, BSTR data, IPnt\_Info \*pPInf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

pGIdx	DBS_GraIndex	[in]	待更新点图元
pDDot	D_Dot	[in]	点坐标
data	String	[in]	点信息
pPInf	Pnt_Info	[in]	点图形信息
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsUpdateDBSPntInfo 方法**

描述 更新图库指定点图元的图形信息。

Visual Basic rtl = DBSArea.dsUpdateDBSPntInfo(pGIdx, pPInf)

C++ HRESULT pDBSArea-> dsUpdateDBSPntInfo (  
IDBS\_GraIndex \*pGIdx, IPnt\_Info \*pPInf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待更新点图元
pPInf	Pnt_Info	[in]	点图形信息
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsUpdateDBSPntPos 方法**

描述 更新图库指定点图元的坐标。

Visual Basic rtl = DBSArea.dsUpdateDBSPntPos(pGIdx, pDDot)

C++ HRESULT pDBSArea-> dsUpdateDBSPntPos (  
IDBS\_GraIndex \*pGIdx, ID\_Dot \*pDDot, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待更新点图元
pDDot	D_Dot	[in]	点坐标
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsUpdateDBSLin 方法**

描述 更新图库指定线图元。

Visual Basic rtl = DBSArea.dsUpdateDBSLin(pGIdx, pDDotSet, pLInf)

C++ HRESULT pDBSArea-> dsUpdateDBSLin(IDBS\_GraIndex \*pGIdx,  
ID\_DotSet \*pDDotSet, ILin\_Info \*pLInf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待更新线图元

pDDotSet	D_DotSet	[in]	线点坐标
pLInf	Lin_Info	[in]	线图形信息
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsUpdateDBSLinDat 方法**

描述 更新图库指定线图元的数据。

Visual Basic rtl = DBSArea.dsUpdateDBSLinDat(pGIdx, pDDotSet)

C++ HRESULT pDBSArea-> dsUpdateDBSLinDat(  
IDBS\_GraIndex \*pGIdx, ID\_DotSet \*pDDotSet,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待更新线图元
pDDotSet	D_DotSet	[in]	线点坐标
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsUpdateDBSLinInfo 方法**

描述 更新图库指定线图元的图形信息。

Visual Basic rtl = DBSArea.dsUpdateDBSLinInfo(pGIdx, pLInf)

C++ HRESULT pDBSArea-> dsUpdateDBSLinInfo (  
IDBS\_GraIndex \*pGIdx, ILin\_Info \*pLInf,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待更新线图元
pLInf	Lin_Info	[in]	线图形信息
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsUpdateDBSReg 方法**

描述 更新图库指定区图元。

Visual Basic rtl = DBSArea.dsUpdateDBSReg(pGIdx, pRdat, pRInf)

C++ HRESULT pDBSArea-> dsUpdateDBSReg (IDBS\_GraIndex \*pGIdx,  
ILongList \*\*pRdat, IReg\_Info \*\*pRInf,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待更新区图元
pRdat	LongList	[in]	区弧段号集合
pRInf	Reg_Info	[in]	区图形信息



rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE
-----	---------	-------	-----------------------

**dsUpdateDBSRegDat 方法**

描述 更新图库指定区图元数据。

Visual Basic rtl = DBSArea.dsUpdateDBSRegDat(pGIdx, pRdat)

C++ HRESULT pDBSArea-> dsUpdateDBSRegDat (  
IDBS\_GraIndex \*pGIdx, ILongList \*\*pRdat, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待更新区图元
pRdat	LongList	[in]	区弧段号集合
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsUpdateDBSRegInfo 方法**

描述 更新图库指定区图元的图形信息。

Visual Basic rtl = DBSArea.dsUpdateDBSRegInfo (pGIdx, pRInf)

C++ HRESULT pDBSArea-> dsUpdateDBSRegInfo (  
IDBS\_GraIndex \*pGIdx, IReg\_Info \*\*pRInf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待更新区图元
pRInf	Reg_Info	[in]	区图形信息
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsUpdateDBSPntAtt 方法**

描述 更新图库指定点图元的属性信息。

Visual Basic rtl = DBSArea.dsUpdateDBSPntAtt(pGIdx, att)

C++ HRESULT pDBSArea-> dsUpdateDBSPntAtt (  
IDBS\_GraIndex \*pGIdx, IRecord \*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	DBS_GraIndex	[in]	待更新点图元
att	Record	[in]	属性记录值
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsUpdateDBSLinAtt 方法**

描述 更新图库指定线图元的属性信息。

Visual Basic rtl = DBSArea.dsUpdateDBSLinAtt(pGIdx, att)

C++ HRESULT pDBSArea-> dsUpdateDBSLinAtt (  
IDBS\_GraIndex \*pGIdx, IRecord \*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	IDBS_GraIndex	[in]	待更新线图元
att	IRecord	[in]	属性记录值
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

**dsUpdateDBSRegAtt 方法**

描述 更新图库指定区图元的属性信息。

Visual Basic rtl = DBSArea.dsUpdateDBSRegAtt(pGIdx, att)

C++ HRESULT pDBSArea-> dsUpdateDBSRegAtt (  
IDBS\_GraIndex \*pGIdx, IRecord \*att, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pGIdx	IDBS_GraIndex	[in]	待更新区图元
att	IRecord	[in]	属性记录值
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

**TrasDDDDFFMMXXX\_ToMMXXX 方法**

描述 检测传入的浮点经纬度分秒值的有效性并正确转换到经纬秒格式。

Visual Basic rtl = DBSArea.TrasDDDDFFMMXXX\_ToMMXXX(dfm, mVal)

C++ HRESULT pDBSArea-> TrasDDDDFFMMXXX\_ToMMXXX (  
double dfm, double \*mVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dfm	Double	[in]	含 0.XXX 的经纬度分秒值
mVal	Double	[out]	含 0.XXX 的经纬秒值
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

**TrasDDDDFFMM\_ToMM 方法**

描述 检测传入的整型经纬度分秒值的有效性并正确转换到经纬秒格式。

Visual Basic rtl = DBSArea.TrasDDDDFFMM\_ToMM (dfm, mVal)

C++ HRESULT pDBSArea-> TrasDDDDFFMM\_ToMM (

---

```
long dfm, long *mVal, VARIANT_BOOL *rtl);
```

参数说明：

参数	类型	出口/入口	描述
dfm	Long	[in]	不含 0.XXX 的经纬度分秒值
mVal	Long	[out]	不含 0.XXX 的经纬秒值
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

---

#### TrasMM\_ToDDDDFFMM 方法

描述 转换整型经纬秒值到经纬度分秒格式。

Visual Basic rtl = DBSArea.TrasMM\_ToDDDDFFMM (mVal, dfmVal)

C++ HRESULT pDBSArea-> TrasMM\_ToDDDDFFMM (  
long mVal, long \*dfmVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
mVal	Long	[in]	不含 0.XXX 的经纬秒值
dfmVal	Long	[out]	不含 0.XXX 的经纬度分秒值
<i>rtl</i>	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

---

#### DisMM\_ToDFM 方法

描述 分解经纬秒值到度、分、秒。

Visual Basic rtl = DBSArea.DisMM\_ToDFM (mVal, d, f, m)

C++ HRESULT pDBSArea-> DisMM\_ToDFM (  
long \*mVal, short \*d,short \*f,short \*m,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
mVal	Long	[in]	不含 0.XXX 的经纬秒值
d	Integer	[out]	分离后的度数
f	Integer	[out]	分离后的分数
m	Integer	[out]	分离后的秒数
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

---

#### DisDDDDFFMM\_ToDFM 方法

描述 分解经纬度分秒值到度、分、秒。

Visual Basic rtl = DBSArea.DisDDDDFFMM\_ToDFM(dfm, d, f, m)

C++ HRESULT pDBSArea-> DisDDDDFFMM\_ToDFM (  
long \*dfm, short \*d,short \*f,short \*m ,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dfm	Long	[in]	不含 0.XXX 的经纬度分秒值
d	Integer	[out]	分离后的度数
f	Integer	[out]	分离后的分数
m	Integer	[out]	分离后的秒数
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

### GetStdJwStepBYScale 方法

描述 根据比例尺分母值获取标准分幅的经纬跨度(度分秒格式)。

Visual Basic rtl = DBSArea.GetStdJwStepBYScale(scale, lonStep, lanStep)

C++ HRESULT pDBSArea-> GetStdJwStepBYScale (  
long scale, double \*lonStep, double\*lanStep, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
scale	Long	[in]	比例尺分母值
lonStep	Double	[out]	标准分幅的经跨度
lanStep	Double	[out]	标准分幅的纬跨度
rtl	Boolean	[out]	成功返回 TRUE 失败返回 FALSE

## 8.8、DBSLayer 对象

描述 DBSLayer 对象封装了图库工作区显示输出有关的操作。

### 8.8.1、属性：无

### 8.8.2、方法：

#### dsDispDBSArea 方法

描述 显示地图库数据。

Visual Basic rtl = DBSLayer.dsDispDBSArea(pMyDC, [IsGraDisp], [IsArcDisp])

C++ HRESULT pDBSLayer-> dsDispDBSArea (  
IMapGisDC \*pMyDC, VARIANT\_BOOL IsGraDisp = TRUE,  
VARIANT\_BOOL IsArcDisp = FALSE, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pMyDC	IMapGisDC	[in]	MapGis 显示对象

IsGraDisp	Boolean	[in]	图库显示方式 可选；缺省值：TRUE
IsArcDisp	Boolean	[in]	TRUE/FALSE：图形/接图表方式 图形显示方式时 可选；缺省值：FALSE TRUE/FALSE：区域弧段显示/屏蔽
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

**dsSetDBSBkClr 方法**

描述 设置接图表绘制背景颜色。

Visual Basic bBkClr = DBSLayer.dsSetDBSBkClr(aBkClr)

C++ HRESULT pDBSLayer-> dsSetDBSBkClr(long aBkClr, long \*bBkClr);

参数说明：

参数	类型	出口/入口	描述
aBkClr	Long	[in]	MapGis 颜色号
bBkClr	Long	[out]	成功则返回以前的背景颜色号

**dsSetDBSBkClr 方法**

描述 获取接图表绘制背景颜色。

Visual Basic nBkClr = DBSLayer.dsGetDBSBkClr()

C++ HRESULT pDBSLayer-> dsSetDBSFrClr(long\* nBkClr);

参数说明：

参数	类型	出口/入口	描述
nBkClr	Long	[out]	成功则返回当前的背景颜色号

**dsSetDBSBkClr 方法**

描述 设置接图表绘制前景颜色。

Visual Basic bFrClr = DBSLayer.dsSetDBSBkClr(aFrClr)

C++ HRESULT pDBSLayer-> dsSetDBSFrClr(long aFrClr, long \*bFrClr);

参数说明：

参数	类型	出口/入口	描述
aFrClr	Long	[in]	MapGis 颜色号
bFrClr	Long	[out]	成功则返回以前的前景颜色号

**dsGetDBSFrClr 方法**

描述 获取接图表绘制前景颜色。

Visual Basic nFrClr = DBSLayer.dsGetDBSFrClr()  
C++ HRESULT pDBSLayer-> dsGetDBSFrClr(long\* nFrClr);

参数说明：

参数	类型	出口/入口	描述
nFrClr	Long	[out]	成功则返回当前的前景颜色号

### dsDrawDBSIndexReg 方法

描述 绘制地图库指定的索引区。

Visual Basic rtl = DBSLayer.dsDrawDBSIndexReg (pMyDC, lFramNo, nClr)  
C++ HRESULT pDBSLayer-> dsDrawDBSIndexReg (  
IMapGisDC \*pMyDC, long lFramNo, long nClr, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pMyDC	IMapGisDC	[in]	MapGis 显示对象
lFramNo	Long	[in]	图幅号
nClr	Long	[in]	颜色号
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

### dsDrawDBSIndexReg1 方法

描述 绘制地图库指定的索引区。

Visual Basic rtl = DBSLayer.dsDrawDBSIndexReg1(hWnd, lFramNo, nClr)  
C++ HRESULT pDBSLayer-> dsDrawDBSIndexReg1 (  
long hWnd, long lFramNo, long nClr, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
hWnd	Long	[in]	绘制对象句柄
lFramNo	Long	[in]	图幅号
nClr	Long	[in]	颜色号
rtl	Boolean	[out]	成功返回 TRUE ,失败返回 FALSE

### dsDispDBSGra 方法

描述 显示图库点、线、区。

Visual Basic rtl = DBSLayer.dsDispDBSGra(pMyDC, pGIdx, gTyp, [clr])  
C++ HRESULT pDBSLayer-> dsDispDBSGra (IMapGisDC \*pMyDC,  
IDBS\_GraIndex \*pGIdx, Enum\_Entity\_Type gTyp,  
long clr = 0, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pMyDC	IMapGisDC	[in]	MapGis 显示对象
pGIdx	DBS_GraIndex	[in]	待显示图元
gType	Enum_Entity_Type	[in]	显示类型
clr	Long	[in]	颜色号 可选；缺省值：0
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dsDispDBSPnt 方法**

描述 显示图库点。

Visual Basic rtl = DBSLayer.dsDispDBSPnt(pMyDC, pGIdx, [clr])

C++ HRESULT pDBSLayer-> dsDispDBSPnt (IMapGisDC \*pMyDC,  
IDBS\_GraIndex \*pGIdx, long clr = 0, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pMyDC	IMapGisDC	[in]	MapGis 显示对象
pGIdx	DBS_GraIndex	[in]	待显示点图元
clr	Long	[in]	颜色号 可选；缺省值：0
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**dsDispDBSLin 方法**

描述 显示图库线。

Visual Basic rtl = DBSLayer.dsDispDBSLin(pMyDC, pGIdx, [clr])

C++ HRESULT pDBSLayer-> dsDispDBSLin (IMapGisDC \*pMyDC,  
IDBS\_GraIndex \*pGIdx, long clr = 0, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pMyDC	IMapGisDC	[in]	MapGis 显示对象
pGIdx	DBS_GraIndex	[in]	待显示线图元
clr	Long	[in]	颜色号 可选；缺省值：0
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### dsDispDBSReg 方法

描述 显示图库区。

Visual Basic rtl = DBSLayer.dsDispDBSReg (pMyDC, pGIdx, [clr])

C++ HRESULT pDBSLayer-> dsDispDBSReg (IMapGisDC \*pMyDC, IDBS\_GraIndex \*pGIdx, long clr = 0, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pMyDC	IMapGisDC	[in]	MapGis 显示对象
pGIdx	DBS_GraIndex	[in]	待显示区图元
clr	Long	[in]	颜色号 可选；缺省值：0
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### dsEnlargeMapDBS 方法

描述 以当前光标所在位置放大图库显示窗口。

Visual Basic DBSLayer.dsEnlargeMapDBS hWnd

C++ HRESULT pDBSLayer-> dsEnlargeMapDBS(long hWnd);

参数说明：

参数	类型	出口/入口	描述
hWnd	Long	[in]	显示对象句柄

### dsDelargeMapDBS 方法

描述 以当前光标所在位置缩小图库显示窗口。

Visual Basic DBSLayer.dsDelargeMapDBS hWnd

C++ HRESULT pDBSLayer-> dsDelargeMapDBS(long hWnd);

参数说明：

参数	类型	出口/入口	描述
hWnd	Long	[in]	显示对象句柄

### dsStrollMapDBS 方法

描述 鼠标滚动移图。

Visual Basic DBSLayer.dsStrollMapDBS hWnd, sx, sy

C++ HRESULT pDBSLayer-> dsStrollMapDBS(long hWnd, int sx, int sy);

参数说明：

参数	类型	出口/入口	描述
hWnd	Long	[in]	显示对象句柄



## 第八章 DBS 对象说明

---

sx	Integer	[in]	光标在窗口上的坐标
sy	Integer	[in]	光标在窗口上的坐标

---

# 第九章 数字高程模型对象说明

DEM 对象的使用示例代码包含在 D:\MAPGIS61\VBDemo\VBForDEM 目录下。

## 9.1、DemInfoStru 对象

描述 DemInfoStru 对象是网格高程数据信息结构。

### 9.1.1、属性：

属性	描述
nx	列数
Long 可读写	
ny	行数
Long 可读写	
xmin	最小 x
Double 可读写	
xmax	最小 x
Double 可读写	
ymin	最小 y
Double 可读写	
ymax	最大 y
Double 可读写	
zmin	最小 z
Double 可读写	
zmax	最大 z
Double 可读写	

### 9.1.2、方法：无

## 9.2、DemInsObj 对象

描述 DemInsObj 对象是规则网 TIN 工作区插密对象。

### 9.2.1、属性：

属性	描述
InsOBJ	规则网 TIN 工作区插密参数
Long	
可读写	

### 9.2.2、方法：无

## 9.3、DistPowGridStru 对象

描述 DistPowGridStru 对象是距离反比网格化参数。

### 9.3.1、属性：

属性	描述
nPower	距离幂次数(缺省为 2)
Integer	
可读写	
fSmooth	距离幂反比插值时的光滑参数(缺省为 0)
Double	
可读写	
fRatio	数据几何各向异性之长短轴比率(1)
Double	
可读写	
fAngl	数据几何各向异性之长轴与 X 轴向夹角(360 度制、度为单位)(0)
Double	
可读写	

### 9.3.2、方法：无

## 9.4、KringGridStru 对象

**描述** KringGridStru 对象是克立金网格化参数。

### 9.4.1、属性：

属性	描述
nKringTyp Integer 可读写	克立金插值模型(0/1/2:幂指数/线性/球状模型)
nDrift Integer 可读写	漂移次数(0/1/2/3 有效)(缺省为 0)
dfcValue Double 可读写	基台值(缺省设为高程数据的总变差即:(Z-Z 均值)的平方和)
dferrValue Double 可读写	测量误差效应值(缺省为 0)
dfmErrValue Double 可读写	微结构误差效应值(缺省为 0)
dfRaudisA Double 可读写	变程数据值(A 缺省设为数据点最小包围矩形对角线的一半长度)
dfRatio Double 可读写	数据各向几何异性之比率(缺省为 1 表示数据各向同性)
dfAngle Double 可读写	异向性参数:主轴与 X 轴向夹角(缺省为 0)

### 9.4.2、方法：无

## 9.5、LabelFontStru 对象

**描述** LabelFontStru 对象是高程注记用字体数据结构。

## 9.5.1、属性：

属性	描述
FontNo Integer 可读写	字体号
ColorNo Long 可读写	颜色号
FixSize Single 可读写	尺寸

## 9.5.2、方法：无

## 9.6、LabelFormatStru 对象

描述 LabelFormatStru 对象是高程注记用格式数据结构。

## 9.6.1、属性：

属性	描述
AbsFlag Boolean 可读写	绝对值标志
CommaFlag Boolean 可读写	采用千位分隔符号
Prefix String 可读写	前缀符号
Suffix String 可读写	后缀符号
DigitNum Integer 可读写	数据小数位数

FormatNo	格式标号(0/1/2:定点/科学/通常)
Long	
可读写	

### 9.6.2、方法：无

## 9.7、SearchDataStru 对象

描述 SearchDataStru 对象是网格化离散数据搜索结构。

### 9.7.1、属性：

属性	描述
nSeekWay	搜索方式(1\2\3\4-->所有点\K-临近\四方向\八方向)
Integer	
可读写	
nDotPerSec	每扇区使用的最多点数
Integer	
可读写	
nMinAllDot	允许最少使用的总点数(少于设定点数该点无效)
Integer	
可读写	
nMaxNullSec	最多允许的空扇区(无效扇区)数
Integer	
可读写	
fRadius	最大搜索圆半径(缺省为数据点最小包围矩形对角线的一半长度)
Double	
可读写	

### 9.7.2、方法：无

## 9.8、SlopeLineStru 对象

描述 SlopeLineStru 对象是示坡线线型参数结构。

### 9.8.1、属性：

属性	描述
nSltp Interger 可读写	线型
nSubSltp Interger 可读写	辅助线型
fxScal Single 可读写	X 系数
fyScal Single 可读写	Y 系数

### 9.8.2、方法：无

## 9.9、TinNetDispStru 对象

描述 TinNetDispStru 对象是高程点、三角网显示参数结构。

### 9.9.1、属性：

属性	描述
PntColNo Long 可读写	顶点颜色号
LinColNo Long 可读写	网线颜色号
IsDispPntNo Boolean 可读写	是否显示高程点号
IsDispTriNo Boolean 可读写	是否显示三角网号

IsDispCode	是否显示地性码
Boolean	
可读写	
IsDispValue	是否显示高程值
Boolean	
可读写	
sDemValDn	高程值显示小数位数
Integer	
可读写	

### 9.9.2、方法：无

## 9.10、TinPnt 对象

描述 TinPnt 对象是三角网中顶点结构。

### 9.10.1、属性：

属性	描述
FDemX	点的 X 值
Double	
可读写	
FDemY	点的 Y 值
Double	
可读写	
FDemZ	点的 Z 值
Double	
可读写	
WCode	地性特征码
Integer	
可读写	
CExist	删除标志
Integer	
可读写	
Null	系统保留
Integer	
可读写	



### 9.10.2、方法：无

## 9.11、TinPntSet 对象

**描述** TinPntSet 对象是三角网顶点集合。

### 9.11.1、属性：

属性	描述
Count	三角网顶点集合的个数
Long	
只读	

### 9.11.2、方法：

#### Append 方法

**描述** 添加一个 TinPnt 三角网顶点(数据从 1 开始存放)

Visual Basic TinNetSet.Append TinP

C++ HRESULT pTinNetSet ->Append(ITinPnt\*pITinP);

**参数说明：**

参数	类型	出口/入口	描述
TinP	TinPnt	[in]	TinPnt 三角网顶点

#### GetItem 方法

**描述** 获取第 index 项数据

Visual Basic TinP = TinNetSet .GetItem (index)

C++ HRESULT pTinNetSet->GetItem(long index, ITinPnt \*\*pITinP);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号(从 0 开始,最大值为 count-1)
TinP	TinPnt	[out]	TinPnt 三角网顶点

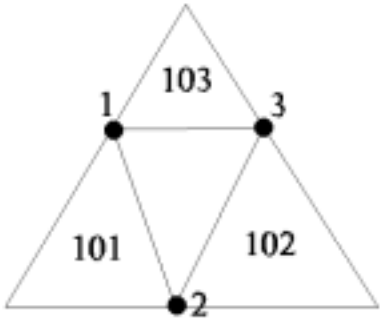
## 9.12、TinNet 对象

**描述** TinNet 对象是三角网结构(三角形序号是其在三角形表中的序号)。

9.12.1、属性：

属性	描述
IVerTex1 Long 可读写	三角形三顶点序号(按逆时针)
IVerTex2 Long 可读写	三角形三顶点序号(按逆时针)
IVerTex3 Long 可读写	三角形三顶点序号(按逆时针)
ITriNo1 Long 可读写	三角形三边对应的三角形序号
ITriNo2 Long 可读写	三角形三边对应的三角形序号
ITriNo3 Long 可读写	三角形三边对应的三角形序号
cExist Integer 可读写	删除标志
Null Integer 可读写	系统保留

如图所示：1、2、3 表示三角形三顶点序号，101、102、103 表示三角形三边对应的三角形序号



### 9.12.2、方法：无

## 9.13、TinNetSet 对象

**描述** TinNetSet 对象是三角网集合。

### 9.13.1、属性：

属性	描述
Count	三角网集合的个数
Long	
只读	

### 9.13.2、方法：

#### Append 方法

**描述** 添加一个 TinNet 三角网(数据从 1 开始存放)

Visual Basic TinNetSet.Append TinN

C++ HRESULT pTinNetSet -> Append(ITinNet \*pITinN);

**参数说明：**

参数	类型	出口/入口	描述
TinN	TinNet	[in]	TinNet 三角网

#### GetItem 方法

**描述** 获取第 index 项数据

Visual Basic pITinN = TinNetSet .GetItem (index)

C++ HRESULT pTinNetSet-> GetItem (long index,ITinNet \*\*pITinN);

**参数说明：**

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号(从 0 开始,最大值为 count-1)
pITinN	TinNet	[out]	TinNet 三角网

## 9.14、ZLevelStru 对象

**描述** ZLevelStru 对象是高程等值追踪用数据结构。

### 9.14.1、属性：

属性	描述
linf	等值线图形参数
Lin_Info	
可读写	
rinf	填充区图形参数
Reg_Info	
可读写	
fZdem	等值线层值
Double	
可读写	
mskOn	该层注记开关
Integer	
可读写	

### 9.14.2、方法：无

## 9.15、ZLevelStruSet 对象

描述 ZLevelStruSet 对象是等值线追踪层数据集合。

### 9.15.1、属性：

属性	描述
count	等值线追踪层数据集合的个数
Long	
只读	

### 9.15.2、方法：

#### Append 方法

描述 添加一个等值线追踪层数据(存放数据从 0 开始)

Visual Basic ZLevelStruSet.Append ZVel

C++ HRESULT pZLevelStruSet ->Append(IZLevelStru \*pIZVel);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

ZVel	ZLevelStru	[in]	ZLevelStru 等值线追踪层数据
------	------------	------	---------------------

**GetItem 方法**

描述 获取第 index 项数据

Visual Basic ZVel = TinNetSet.GetItem (index)

C++ HRESULT pZLevelStruSet -> GetItem(long index, IZLevelStru \*\* pIZVel);

参数说明：

参数	类型	出口/入口	描述
index	Long	[in]	集合的索引号(从 0 开始,最大值为 count-1)
Zvel	ZLevelStru	[out]	ZLevelStru 等值线追踪层数据

**9.16、TinArea 对象**

描述 TinArea 对象封装了对 Tin 工作区的基本操作。

**9.16.1、属性：**

属性	描述
Type	工作区类型
Enum_Area_Type	
只读	
Name	工作区名称
String	
可读写	
Changed	工作区已经改变标志
Boolean	
可读写	
ShowErrMsg	显示错误信息标志
Boolean	
可读写	
Warning	对不能后悔的操作是否给出警告提示
Boolean	
可读写	
Compact	保存工作区时压缩数据标志
Boolean	
可读写	

Empty	工作区是否为空标志
Boolean	
只读	
LastErrCode	最后错误码
Long	
只读	
rect	图形范围
D_Rect	
只读	
pTinModel	TinModel 对象
TinModel	
只读	
pGridModel	GridModel 对象
GridModel	
只读	
pAppModel	AppModel 对象
AppModel	
只读	

## 9.16.2、方法：

### GetFileType 方法

描述 提取高程文件类型

Visual Basic pVal = TinArea.GetFileType (fname)

C++ HRESULT pTinArea->GetFileType(  
BSTR fname, Enum\_TinArea\_File\_Type \*pVal);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名
PVal	Enum_TinArea_File_Type	[out]	高程文件类型

### LoadFile 方法

描述 装入高程文件

Visual Basic rtl = TinArea.LoadFile([fname])

C++ HRESULT pTinArea->LoadFile(  
BSTR fname = NULL,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

fname	<i>String</i>	[in]	文件名 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**PackSaveFileAs 方法**

描述 压缩存文件

Visual Basic rtl = TinArea.PackSaveFileAs ()

C++ HRESULT pTinArea-> PackSaveFileAs (VARIANT\_BOOL \* rtl);

参数说明：

参数	类型	出口/入口	描述
Rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SaveFile 方法**

描述 保存高程文件

Visual Basic rtl = TinArea.SaveFile ([fname])

C++ HRESULT pTinArea-> SaveFile (  
BSTR fname = NULL, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	文件名 可选；缺省值：Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ExportTinDat 方法**

描述 输出数据高程文件(输出格式为\*.det,\*.bdm)

Visual Basic rtl = TinArea.ExportTinDat ([fname])

C++ HRESULT pTinArea-> ExportTinDat (BSTR fname = NULL,  
VARIANT\_BOOL \* rtl);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	输出数据高程文件名 可选 ;缺省值 :Nothing
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetTinPntNum 方法**

描述 获取当前 TIN 工作区三角网顶点数

Visual Basic rtl = TinArea.GetTinPntNum (logN, phyN)

C++ HRESULT pTinArea-> GetTinPntNum (long \*logN, long \*phyN,  
VARIANT\_BOOL \* rtl);

参数说明：

参数	类型	出口/入口	描述
logN	Long	[out]	高程点逻辑数，即未被删除的顶点数
phyN	Long	[out]	高程点物理数（物理数=逻辑数+被删除点数+1）
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetTinNetNum 方法

描述 获取当前 TIN 工作区三角网数

Visual Basic rtl = TinArea.GetTinNetNum (logN, phyN)

C++ HRESULT pTinArea-> GetTinNetNum (long \*logN, long \*phyN, VARIANT\_BOOL \* rtl);

参数说明：

参数	类型	出口/入口	描述
logN	Long	[out]	三角网的逻辑数
phyN	Long	[out]	三角网的物理数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### IsGridNet 方法

描述 判断当前工作区中文件是否为规则格网文件

Visual Basic rtl = TinArea.IsGridNet ()

C++ HRESULT pTinArea-> IsGridNet (VARIANT\_BOOL \* rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	是规则网返回 TRUE，否则返回 FALSE

### GetTinZValRange 方法

描述 获取当前工作区高程值的范围

Visual Basic rtl = TinArea.GetTinZValRange (maxZval, minZval,)

C++ HRESULT pTinArea->GetTinZValRange (double \* maxZval, double \* minZval, VARIANT\_BOOL \* rtl);

参数说明：

参数	类型	出口/入口	描述
maxZval	Double	[out]	高程值的最大值
minZval	Double	[out]	高程值的最小值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE



**SetTinZValRange 方法**

描述 设置当前工作区高程值的范围

Visual Basic rtl = TinArea.SetTinZValRange (maxZval, minZval,)

C++ HRESULT pTinArea->SetTinZValRange (double maxZval,  
double minZval, VARIANT\_BOOL \* rtl);

参数说明：

参数	类型	出口/入口	描述
maxZval	Double	[in]	高程值的最大值
minZval	Double	[in]	高程值的最小值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetTinRangeFromFile 方法**

描述 由 TIN 高程文件名直接获取高程文件范围

Visual Basic rtl = TinArea.GetTinRangeFromFile (fname, frc)

C++ HRESULT pTinArea->GetTinRangeFromFile (BSTR fname,  
ID\_Rect \*\*frc, VARIANT\_BOOL \* rtl );

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	TIN 高程文件名
frc	D_Rect	[out]	返回图形范围
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetTinMapParamFromFile 方法**

描述 由 TIN 高程文件名直接获取地图投影参数

Visual Basic rtl = TinArea.GetTinMapParamFromFile (fname, tnMapPara)

C++ HRESULT pTinArea->GetTinMapParamFromFile (BSTR fname,  
IMap\_Para \*\*tnMapPara, VARIANT\_BOOL \* rtl);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	TIN 高程文件名
tnMapPara	Map_Para	[out]	返回地图投影参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetTinDispMapParamFromFile 方法**

描述 由 TIN 高程文件名直接获取显示地图投影参数

Visual Basic rtl = TinArea.GetTinDispMapParamFromFile (fname, tnDspMapPara)

C++ HRESULT pTinArea->GetTinDispMapParamFromFile (BSTR fname,

IMap\_Para \*\* tnDspMapPara,VARIANT\_BOOL \* rtl);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	TIN 高程文件名
tnDspMapPara	Map_Para	[out]	返回显示地图投影参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetTinMapParam 方法

描述 由 TIN 工作区获取地图投影参数

Visual Basic rtl = TinArea.GetTinMapParam (dbMapParam)

C++ HRESULT pTinArea-> GetTinMapParam (IMap\_Para \*\*dbMapParam,  
VARIANT\_BOOL \* rtl);

参数说明：

参数	类型	出口/入口	描述
dbMapParam	Map_Para	[out]	返回地图投影参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### SetTinMapParam 方法

描述 设置地图投影参数

Visual Basic rtl = TinArea.SetTinMapParam (dbMapParam)

C++ HRESULT pTinArea-> SetTinMapParam (IMap\_Para \*dbMapParam,  
VARIANT\_BOOL \* rtl);

参数说明：

参数	类型	出口/入口	描述
dbMapParam	Map_Para	[out]	返回地图投影参数
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### EditTinMapParam 方法

描述 对话框编辑地图投影参数

Visual Basic rtl = TinArea.EditTinMapParam ()

C++ HRESULT pTinArea-> EditTinMapParam (VARIANT\_BOOL \* rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ViewTinMapParam 方法**

描述 对话框浏览地图投影参数

Visual Basic TinArea.ViewTinMapParam

C++ HRESULT pTinArea-> ViewTinMapParam ();

参数说明： 无

**SetTinDispCordParam 方法**

描述 对话框设置 TIN 高程坐标显示用投影参数、坐标单位编辑

Visual Basic rtl = TinArea.SetTinDispCordParam ()

C++ HRESULT pTinArea-> SetTinDispCordParam (VARIANT\_BOOL \* rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CvtPntDispCord 方法**

描述 将 TIN 高程坐标转换为 TIN 高程显示坐标

Visual Basic rtl = TinArea.CvtPntDispCord (xx,yy)

C++ HRESULT pTinArea-> CvtPntDispCord (double \*xx, double \*yy, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
xx	Double	[in,out]	[in]TIN 高程坐标 x 值, [out] TIN 高程显示坐标 x 值
yy	Double	[in,out]	[in]TIN 高程坐标 y 值, [out] TIN 高程显示坐标 y 值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetTinCoorUnit 方法**

描述 获取 TIN 高程数据坐标数据显示单位号

Visual Basic psUnit = TinArea.GetTinCoorUnit ()

C++ HRESULT pTinArea-> GetTinCoorUnit (short \*psUnit);

参数说明：

参数	类型	出口/入口	描述
psUnit	Integer	[out]	TIN 高程数据坐标数据显示单位号

**GetTinCoorDispUnit 方法**

描述 取 TIN 高程数据坐标数据当前显示单位号

Visual Basic psUnit = TinArea.GetTinCoorDispUnit ()  
C++ HRESULT pTinArea-> GetTinCoorDispUnit (short \*psUnit);

参数说明：

参数	类型	出口/入口	描述
psUnit	Integer	[out]	TIN 高程数据坐标数据当前显示单位号

### GetPntDispCordStr 方法

描述 获取 TIN 高程显示坐标提示串

Visual Basic Str = TinArea. GetPntDispCordStr (x,y)

C++ HRESULT pTinArea-> GetPntDispCordStr (double x, double y, BSTR \*Str);

参数说明：

参数	类型	出口/入口	描述
x	Double	[in]	字符串 x 坐标
y	Double	[in]	字符串 y 坐标
Str	String	[out]	成功返回坐标提示串

### GetTinCoorUnitName 方法

描述 取 TIN 高程数据坐标数据显示单位名

Visual Basic Str = TinArea. GetTinCoorUnitName ()

C++ HRESULT pTinArea-> GetTinCoorUnitName (BSTR \*Str);

参数说明：

参数	类型	出口/入口	描述
Str	String	[out]	返回高程数据坐标数据显示单位名

### GetTinCoorDispUnitName 方法

描述 取 TIN 高程数据坐标数据当前显示单位名

Visual Basic Str = TinArea. GetTinCoorDispUnitName ()

C++ HRESULT pTinArea-> GetTinCoorDispUnitName (BSTR \*Str);

参数说明：

参数	类型	出口/入口	描述
Str	String	[out]	返回高程数据坐标数据当前显示单位名

### EditCurTinPnt 方法

描述 编辑当前高程点数据(弹出对话框)

Visual Basic pResult = TinArea.EditCurTinPnt (triPnt )

C++ HRESULT pTinArea-> EditCurTinPnt (ITinPnt \*triPnt,short \* pResult);

参数说明：

参数	类型	出口/入口	描述
triPnt	TinPnt	[in]	当前高程点数据信息
pResult	Integer	[out]	返回值 1 表示确认,0 表示取消或失败

**GetNearestTinPntNo 方法**

描述 根据坐标点获取高程点号

Visual Basic li = TinArea.GetNearestTinPntNo (triPnt )

C++ HRESULT pTinArea-> GetNearestTinPntNo (ID\_Dot \*Pnt, long \*li);

参数说明：

参数	类型	出口/入口	描述
Pnt	D_Dot	[in]	坐标点
li	Long	[out]	高程点号

**GetTinDatInfo 方法**

描述 获取给定高程工作区高程数据的描述信息

Visual Basic rtl = TinArea.GetTinDatInfo (mode, nx, ny, nxWth, nyDth)

C++ HRESULT pTinArea-> GetTinDatInfo (short \*mode, long \*nx, long \*ny, double \*nxWth, double \*nyDth, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
mode	Integer	[out]	文件类型
nx	Long	[out]	网格列数
ny	Long	[out]	网格行数
nxWth	Double	[out]	网格宽度
nyDth	Double	[out]	网格高度
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetGridTinDatHdInfo 方法**

描述 获取给定高程工作区高程数据的描述信息

Visual Basic pResult = TinArea.GetGridTinDatHdInfo (demHeadInfo)

C++ HRESULT pTinArea-> GetGridTinDatHdInfo (IDemInfoStru \*\*demHeadInfo, short \*pResult );

参数说明：

参数	类型	出口/入口	描述
demHeadInfo	DemInfoStru	[out]	高程数据的描述信息
pResult	Integer	[out]	返回值 2 表示规则网,1 表示非规则网,0 表示失败

**ViewTinDatInfo 方法**

描述 TIN 工作区数据信息对话框显示

Visual Basic TinArea.ViewTinDatInfo

C++ HRESULT pTinArea-> GetGridTinDatHdInfo ();

参数说明： 无

**SameDotProc 方法**

描述 TIN 工作区重复高程点处理

Visual Basic li = TinArea.SameDotProc (way)

C++ HRESULT pTinArea-> SameDotProc  
(Enum\_TinArea\_Duplicated\_Point\_Management\_Type way, long \*li );

参数说明：

参数	类型	出口/入口	描述
Way	Enum_TinArea_Duplicated _Point_Management_Typ	[in]	重复点处理方式
Li	Long	[out]	处理掉的重复点数

**GetGridDatByTinPnt 方法**

描述 将 TIN 工作区离散点建立三角网后内插网格化

Visual Basic rtl = TinArea. GetGridDatByTinPnt (pclipRai)

C++ HRESULT pTinArea-> GetGridDatByTinPnt (IRegArea \* pclipRai,  
VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pclipRai	RegArea	[in]	裁剪区域（缺省值为空）
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**OutBufSectionMap 方法**

描述 由剖面线基线和内插的各段表面生成剖面图

Visual Basic rtl = TinArea.OutBufSectionMap (fdLptr, ptD3Sec, plai, prai, poutlai, poutpai)

C++ HRESULT pTinArea-> OutBufSectionMap (  
ID\_DotSet \*fdLptr, ID\_3DotSet \*ptD3Sec, ILineArea \*plai,  
IRegArea \*prai, ILineArea \*poutlai, IPntArea \*poutpai, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fdLptr	ID_DotSet	[in]	剖面定位线数据

ptD3Sec	D_3DotSet	[in]	剖面定位线内插生成的高程表面数据
plai	LinArea	[in]	传进的线工作区
prai	RegArea	[in]	传进的区工作区
poutlai	LinArea	[in]	剖面线插值处理结果输出线工作区
poutpai	PntArea	[in]	剖面线插值处理结果输出点工作区
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**EditLabelFont 方法**

描述 给出对话框编辑标注字体参数

Visual Basic rtl = TinArea.EditLabelFont (LabelFont)

C++ HRESULT pTinArea-> EditLabelFont (  
ILabelFontStru \*LabelFont, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
LabelFont	LabelFontStru	[in]	标注字体参数对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**EditLabelFormat 方法**

描述 给出对话框编辑标注格式参数

Visual Basic rtl = TinArea.EditLabelFormat (LabelFormat)

C++ HRESULT pTinArea-> EditLabelFormat (  
ILabelFormatStru \*LabelFormat, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
LabelFormat	LabelFormatStru	[in]	标注格式参数对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetLabelByFmt 方法**

描述 按标注格式形成标注

Visual Basic rtl = TinArea.GetLabelByFmt (LabelFmt, LabelValue, Label)

C++ HRESULT pTinArea-> GetLabelByFmt (ILabelFormatStru \*LabelFmt,  
double LabelValue, BSTR \*Label, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
LabelFmt	LabelFormatStru	[in]	标注格式参数对象
LabelValue	Double	[in]	高程注记值
Label	String	[out]	标注

rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE
-----	---------	-------	------------------------

**MakeColScale 方法**

描述 生成等值线图色阶

Visual Basic rtl = TinArea.MakeColScale (prai, ppai, plai, ptZVelSet, sclN, lbFmt, lbFnt, x0, y0, xWidth, rtl)

C++ HRESULT pTinArea-> MakeColScale ( IRegArea \*prai, IPntArea \*ppai, ILineArea \*plai, IZLevelStruSet \*ptZVelSet, short sclN, ILabelFormatStru \*lbFmt, ILabelFontStru \*lbFnt, double x0, double y0, double xWidth, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
prai	RegArea	[in]	结果区工作区对象
ppai	PntArea	[in]	结果点工作区对象
plai	LineArea	[in]	结果线工作区对象
ptZVelSet	ZLevelStruSet	[in]	等值线追踪层数据集(元素个数为 sclN+1)
sclN	Integer	[in]	标尺级数
lbFmt	LabelFormatStru	[in]	高程注记用格式数据
lbFnt	LabelFontStru	[in]	高程注记用字体数据
x0	Double	[in]	最小包络矩形的 x 最小值.
y0	Double	[in]	最小包络矩形的 y 最小值
xWidth	Double	[in]	制图数据 X 向幅面宽度
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

**MakeDemOrTinFrame 方法**

描述 生成 DEM 或 TIN 数据的最小包络矩形外框

Visual Basic rtl = TinArea. MakeDemOrTinFrame (plai, x0, y0, x1,y1)

C++ HRESULT pTinArea-> MakeDemOrTinFrame (ILineArea \*plai, double x0, double y0, double x1, double y1, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
plai	LineArea	[in]	结果线工作区对象
x0	Double	[in]	最小包络矩形的 x 最小值.
y0	Double	[in]	最小包络矩形的 y 最小值
x1	Double	[in]	最小包络矩形的 x 最大值
y1	Double	[in]	最小包络矩形的 y 最大值



rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE
-----	---------	-------	------------------------

**TraceGridDEM 方法**

描述 追踪等值线

Visual Basic rtl = TinArea.TraceGridDEM (dmHdInf, xl, yl, varDat, ptZVel, scln, lbFmt, lbFmt, pSlopLinPrm, bSaveEdg, nlCps, colscl, nNoteDir, prai, ppai, plai, pTs)

C++ HRESULT pTinArea-> TraceGridDEM (IDemInfoStru \*dmHdInf, double xl, double yl, VARIANT varDat, IZLevelStruSet \*ptZVel, short scln, ILabelFormatStru \*lbFmt, ILabelFontStru \*lbFmt, ISlopeLineStru \*pSlopLinPrm, VARIANT\_BOOL bSaveEdg, short nlCps, short colscl, short nNoteDir, IRegArea \*prai, IPntArea \*ppai, ILinArea \*plai, IPrjTrans \*pTs, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dmHdInf	DemInfoStru	[in]	高程数据信息
xl	Double	[in]	X 向制图数据长度
yl	Double	[in]	Y 向制图数据宽度
varDat	Variant	[in]	高程数据, Double 型数组
ptZVel	ZLevelStruSet	[in]	等值线追踪层数据集合(元素个数为 sclN+1)
scln	Integer	[in]	标尺级数
lbFmt	LabelFormatStru	[in]	高程注记用格式数据
lbFmt	LabelFontStru	[in]	高程注记用字体数据
pSlopLinPrm	SlopeLineStru	[in]	示坡线线型参数
bSaveEdg	Boolean	[in]	边框标志
nlCps	Integer	[in]	光滑等级(<1/1/2/>=3: 禁止/低度/中度/高度 光滑)
colscl	Integer	[in]	是否画色阶
nNoteDir	Integer	[in]	是否加注记并确定注记字头朝向 (<1/1/2/其它->不注记/斜坡上方/斜坡下方/图幅上方)
prai	RegArea	[in]	结果区工作区对象
ppai	PntArea	[in]	结果点工作区对象
plai	LinArea	[in]	结果线工作区对象
pTs	PrjTrans	[in]	对原始数据投影用投影类对象, 缺省为 NULL
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

## VcTraceGridDEM 方法

描述 追踪等值线

Visual Basic rtl = TinArea.VcTraceGridDEM (dmHdInf, xl, yl, varDat, ptZVel, scln, lbFmt, lbFnt, pSlopLinPrm, bSaveEdg, nLCps, colscl, nNoteDir, prai, ppai, plai, pTs)

C++ HRESULT pTinArea-> VcTraceGridDEM (IDemInfoStru \*dmHdInf, double xl, double yl, VARIANT varDat, IZLevelStruSet \*ptZVel, short scln, ILabelFormatStru \*lbFmt, ILabelFontStru \*lbFnt, ISlopeLineStru \*pSlopLinPrm, VARIANT\_BOOL bSaveEdg, short nLCps, short colscl, short nNoteDir, IRegArea \*prai, IPntArea \*ppai, ILinArea \*plai, IPrjTrans \*pTs, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dmHdInf	DemInfoStru	[in]	高程数据信息
xl	Double	[in]	X 向制图数据长度
yl	Double	[in]	Y 向制图数据宽度
varDat	Variant	[in]	高程数据,Double 型数组
ptZVel	ZLevelStruSet	[in]	等值线追踪层数据集合(元素个数为 sclN+1)
scln	Integer	[in]	标尺级数
lbFmt	LabelFormatStru	[in]	高程注记用格式数据
lbFnt	LabelFontStru	[in]	高程注记用字体数据
pSlopLinPrm	SlopeLineStru	[in]	示坡线线型参数
bSaveEdg	Boolean	[in]	边框标志
nLCps	Integer	[in]	光滑等级(<1/1/2/>=3:禁止/低度/中度/高度光滑)
colscl	Integer	[in]	是否画色阶
nNoteDir	Integer	[in]	是否加注记并确定注记字头朝向 (<1/1/2/其它->不注记/斜坡上方/斜坡下方/图幅上方)
prai	RegArea	[in]	结果区工作区对象
ppai	PntArea	[in]	结果点工作区对象
plai	LinArea	[in]	结果线工作区对象
pTs	PrjTrans	[in]	对原始数据投影用投影类对象,缺省为 NULL
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

**TraceBinDEMToREG 方法**

描述 追踪一层等值线并形成区

Visual Basic rtl = TinArea.TraceBinDEMToREG (xn, yn, xl, yl, datN, dat, ptZVel, nlCps, prai)

C++ HRESULT pTinArea->TraceBinDEMToREG (long xn, long yn, double xl, double yl, long datN, short \*dat, IZLevelStruSet \*ptZVel, short nlCps, IRegArea \*prai, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
xn	Long	[in]	X 向点数
yn	Long	[in]	Y 向点数
xl	Double	[in]	X 向长度
yl	Double	[in]	Y 向长度
datN	Long	[in]	dat 数组的元素个数
dat	Integer	[in]	数组
ptZVel	ZLevelStruSet	[in]	等值线追踪层数据集合（共两个） 如果 ptZVel=NULL,则使用缺省线、 区参数(不处理无效值区域)
nlCps	Integer	[in]	光滑等级(<1/1/2/>=3:禁止/低度/中度/高度光滑)
prai	RegArea	[in]	结果区工作区对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**TraceBinDEMToLIN 方法**

描述 追踪一层等值线并形成线

Visual Basic rtl = TinArea.TraceBinDEMToLIN (xn, yn, xl, yl, datN, dat, ptZVel, nlCps, plai)

C++ HRESULT pTinArea->TraceBinDEMToLIN (long xn, long yn, double xl, double yl, long datN, short \*dat, IZLevelStru \*ptZVel, short nlCps, ILineArea \*plai, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
xn	Long	[in]	X 向点数
yn	Long	[in]	Y 向点数
xl	Double	[in]	X 向长度
yl	Double	[in]	Y 向长度
datN	Long	[in]	dat 数组的元素个数
dat	Integer	[in]	数组

ptZVel	ZLevelStruSet	[in]	等值线追踪层数据集合（共两个） 如果 ptZVel=NULL,则使用缺省线、 区参数(不处理无效值区域)
nlCps	Integer	[in]	光滑等级(<1/1/2/>=3:禁止/低度/中 度/高度光滑)
plai	RegArea	[in]	线工作区对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GrdDemTraceContourByBuf 方法**

描述 规则网等值线追踪

Visual Basic rtl = TinArea. GrdDemTraceContourByBuf (pHdInf, varDat, plai,  
ppai, prai, [flagdfNaN], [dfNaN], [bIsVector])

C++ HRESULT pTinArea-> GrdDemTraceContourByBuf (IDemInfoStru \*pHdInf,  
VARIANT varDat, ILineArea \*plai, IPntArea \*ppai, IRegArea \*prai,  
VARIANT\_BOOL flagdfNaN, double dfNaN, VARIANT\_BOOL bIsVector,  
VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pHdInf	DemInfoStru	[in]	高程数据信息
varDat	Variant	[in]	高程数据，Double 型数组，数组大 小为 (PtHdInf ->nx * PtHdInf ->ny)
plai	LineArea	[in]	结果线工作区对象
ppai	PntArea	[in]	结果点工作区对象
prai	RegArea	[in]	结果区工作区对象
flagdfNaN	Boolean	[in]	标志是否有未知高程值,有未知高程 值为 True,无未知高程值为 False。 缺省为 False
dfNaN	Double	[in]	未知高程值。若 flagUnKnowZVal = False，UnKnowZVal 的值没有意义， 可以输入也可以用缺省值；若 flagUnKnowZVal = True， UnKnowZVal 的值表示未知高程值
bIsVector	Boolean	[in]	TRUE/FALSE:矢量/栅格法，缺省为 TRUE
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GrdDemTraceContourByBuf 方法**

描述 规则网等值线追踪(含参数设置对话框)

Visual Basic rtl = TinArea. GrdDemTraceContourByBuf (pHdInf, varDat, plai, ppai, prai, [flagdfNaN], [dfNaN], [bIsVector])

C++ HRESULT pTinArea-> GrdDemTraceContourByBuf (IDemInfoStru \*pHdInf, VARIANT varDat, ILinArea \*plai, IPntArea \*ppai, IRegArea \*prai, VARIANT\_BOOL flagdfNaN, double dfNaN, VARIANT\_BOOL bIsVector, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pHdInf	DemInfoStru	[in]	高程数据信息
varDat	Variant	[in]	高程数据，Double 型数组，数组大小为(PtHdInf ->nx * PtHdInf ->ny)
plai	LinArea	[in]	结果线工作区对象
ppai	PntArea	[in]	结果点工作区对象
prai	RegArea	[in]	结果区工作区对象
flagdfNaN	Boolean	[in]	标志是否有未知高程值,有未知高程值为 True,无未知高程值为 False。缺省为 False
dfNaN	Double	[in]	未知高程值。若 flagUnKnowZVal = False，UnKnowZVal 的值没有意义，可以输入也可以用缺省值；若 flagUnKnowZVal = True，UnKnowZVal 的值表示未知高程值
bIsVector	Boolean	[in]	TRUE/FALSE:矢量/栅格法，缺省为 TRUE
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GrdDemTraceContourByBuf 方法**

描述 规则网高程工作区等值线追踪(含参数设置对话框)

Visual Basic rtl = TinArea. GrdDemTraceContourByBuf (plai, ppai, prai, [bIsVector])

C++ HRESULT pTinArea-> GrdDemTraceContourByBuf (ILinArea \*plai, IPntArea \*ppai, IRegArea \*prai, VARIANT\_BOOL bIsVector, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
plai	LinArea	[in]	结果线工作区对象
ppai	PntArea	[in]	结果点工作区对象

prai	RegArea	[in]	结果区工作区对象
bIsVector	Boolean	[in]	TRUE/FALSE:矢量/栅格法, 缺省为 TRUE
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**TraceContourByTin 方法**

描述 三角网高程工作区等值线追踪(含参数设置对话框)

Visual Basic rtl= TinArea. TraceContourByTin(ptZVel, zVelNb, lbFmt, lbFnt, pSlopLinPrm, cnCps, bSaveEdg, colscl, nNoteDir, prai, ppai, plai, dfNoteStep, pTs)

C++ HRESULT pTinArea-> TraceContourByTin(IZLevelStruSet \*ptZVel, short zVelNb, ILabelFormatStru \*lbFmt, ILabelFontStru \*lbFnt, ISlopeLineStru \*pSlopLinPrm, short cnCps, VARIANT\_BOOL bSaveEdg, short colscl, short nNoteDir, IRegArea \*prai, IPntArea \*ppai, ILineArea \*plai, double dfNoteStep, IPrjTrans \*pTs, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptZVel	ZLevelStruSet	[in]	等值线追踪层数据集合,(元素个数为 zVelNb+1)
zVelNb	Integer	[in]	标尺级数
lbFmt	LabelFormatStru	[in]	高程注记用格式数据
lbFnt	LabelFontStru	[in]	高程注记用字体数据
pSlopLinPrm	SlopeLineStru	[in]	示坡线线型参数
cnCps	Integer	[in]	光滑等级(<1/1/2/>=3:禁止/低度/中度/高度光滑)
bSaveEdg	Boolean	[in]	边框标志
colscl	Integer	[in]	是否画色阶
nNoteDir	Integer	[in]	是否加注记并确定注记字头朝向(<1/1/2/其它->不注记/斜坡上方/斜坡下方/图幅上方)
plai	LineArea	[in]	结果线工作区对象
ppai	PntArea	[in]	结果点工作区对象
prai	RegArea	[in]	结果区工作区对象
dfNoteStep	Double	[in]	等值线注记间距
pTs	PrjTrans	[in]	对原始数据投影用投影类对象, 缺省为 NULL
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**TinTriNetTraceContour 方法**

描述 三角网高程工作区等值线追踪

Visual Basic rtl = TinArea.TinTriNetTraceContour (plai, ppai, prai)

C++ HRESULT pTinArea-> TinTriNetTraceContour (ILinArea \*plai, IPntArea \*ppai, IRegArea \*prai, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
plai	LinArea	[in]	结果区工作区对象
ppai	PntArea	[in]	结果点工作区对象
prai	RegArea	[in]	结果区工作区对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetTinBorderLin 方法**

描述 获取给定高程工作区的内外边界数据, 存入线文件

Visual Basic rtl = TinArea.GetTinBorderLin (plai)

C++ HRESULT pTinArea-> GetTinBorderLin (ILinArea \*plai, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
plai	LinArea	[in]	结果线工作区对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ReduceAllLin 方法**

描述 线数据抽稀处理

Visual Basic rtl = TinArea.ReduceAllLin (plai)

C++ HRESULT pTinArea-> ReduceAllLin (ILinArea \*plai, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
plai	LinArea	[in]	线工作区对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**LinDatDiscreting 方法**

描述 按线通过栅格方式离散化线为栅格 GRD(获取的 GRD 数据存入文件)

Visual Basic rtl = TinArea.LinDatDiscreting (plai, mzk, mHdInf, fname)

C++ HRESULT pTinArea-> LinDatDiscreting (ILinArea \*plai, short mzk, IDemInfoStru \*mHdInf, BSTR fname, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

plai	LinArea	[in]	高程等值线工作区对象
mzk	Integer	[in]	线属性中的高程域序号
mHdInf	DemInfoStru	[in]	网格化参数(zmin->dltX,zmax->dltY)
fname	String	[out]	待生成的 GRD 文件名
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetDiscreteGrdDatByLin 方法**

描述 按线通过栅格方式离散化线为栅格 GRD(含参数设置对话框)

Visual Basic rtl = TinArea. GetDiscreteGrdDatByLin (plai)

C++ HRESULT pTinArea-> GetDiscreteGrdDatByLin (  
ILinArea \*plai, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
plai	LinArea	[in]	高程等值线工作区对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**LinCbsProc 方法**

描述 贝叶斯不过点光滑例程

Visual Basic rtl = TinArea. LinCbsProc (plai)

C++ HRESULT pTinArea-> LinCbsProc (ILinArea \*plai, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
plai	LinArea	[in]	光滑线工作区
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**TraceDEMGridToLin 方法**

描述 追踪待追踪等值线数据的有效区边界

Visual Basic rtl = TinArea. TraceDEMGridToLin (xs, ys, xdlt, ydlt, xn, yn, varZdat, NaN, plai)

C++ HRESULT pTinArea-> TraceDEMGridToLin (double xs, double ys, double xdlt,  
double ydlt, long xn, long yn, VARIANT varZdat, double NaN,  
ILinArea \*plai, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
xs	Long	[in]	X 向起点坐标
ys	Long	[in]	Y 向起点坐标
xdlt	Double	[in]	X 向网格步长
ydlt	Double	[in]	Y 向网格步长



xn	Long	[in]	X 向点数
yn	Long	[in]	Y 向点数
varZdat	Variant	[in]	原始数据,Double 型数组,大小为 xn*yn
NaN	Double	[in]	无效标识数据
plai	LinArea	[in]	结果线工作区对象
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

**TraceBinDEMDatToLin 方法**

描述 追踪待追踪等值线数据的有效区边界

Visual Basic rtl = TinArea. TraceBinDEMDatToLin (xs, ys, xdlt, ydlt, xn, yn, datN, dat, plai)

C++ HRESULT pTinArea-> TraceBinDEMDatToLin (double xs, double ys,  
double xdlt, double ydlt, long xn, long yn,long datN, short \*dat,  
ILinArea \* plai, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
xs	Long	[in]	X 向起点坐标
ys	Long	[in]	Y 向起点坐标
xdlt	Double	[in]	X 向网格步长
ydlt	Double	[in]	Y 向网格步长
xn	Long	[in]	X 向点数
yn	Long	[in]	Y 向点数
datN	datN	[in]	数组 dat 的元素个数
dat	Integer	[in]	原始数据, Integer 型数组,大小为 xn*yn = datN
plai	LinArea	[in]	结果线工作区对象
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

**ConvertDemTo2ValRbm 方法**

描述 将规则网数据写成可矢量化的 RBM 二值文件

Visual Basic rtl = TinArea. ConvertDemTo2ValRbm (szRbmName, dHdInf, varZVal)

C++ HRESULT pTinArea-> ConvertDemTo2ValRbm (BSTR szRbmName,  
IDemInfoStru \*dHdInf, VARIANT varZVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
szRbmName	String	[in]	待生成的文件名
dHdInf	DemInfoStru	[in]	规则网高程数据信息

varZVal	Variant	[in,out]	规则网高程数据,Double 型数组,大小为 dHdInf->nx* dHdInf->ny;同时返回结果数据
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

**DrainageAreaAnaly 方法**

描述 水文表面流域分析 (含对话框以设置分析参数)

Visual Basic rtl = TinArea. DrainageAreaAnaly (hWnd)

C++ HRESULT pTinArea-> DrainageAreaAnaly (long hWnd, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
hWnd	Long	[in]	窗口句柄
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

**9.17、TinModel 对象**

描述 TinModel 对象封装了对 Tin 模型的操作,该对象不能直接被创建,但能被 TinArea 对象内部实例化。可以通过 TinArea 对象的 pTinModel 属性或查询 TinArea 对象的 iTinModel 接口来引用 TinModel 对象。具体使用方法如下示：

在 VB 中：

```
dim iTinArea as new TinArea '创建 TinArea 对象
dim iTinModel as TinModel
Set iTinModel = iTinArea.pTinModel '间接创建 TinModel 对象
Set iTinModel = iTinArea '此种方式也可创建 TinModel 对象
iTinModel.AppendTinPnt(triPnt) '调用 TinModel 对象的方法
```

在 VC 中：

```
ITinArea *ptTinArea;
hr=CoCreateInstance(CLSID_TinArea,
                    NULL,
                    CLSCTX_INPROC_SERVER,
                    IID_ITinArea,
                    (void**)&ptTinArea);// 创建TinArea对象

ITinModel *ptTinModel;
ptTinArea->get_pTinModel(&ptTinModel);// 间接创建TinModel对象
ptTinArea->QueryInterface(IID_ITinModel,(void **)& ptTinArea); //此种方式也可创建
//TinModel对象
```

---

ptTinModel->AppendTinPnt(triPnt);// 调用TinModel对象的方法

### 9.17.1、属性：无

### 9.17.2、方法：

#### AppendTinPnt 方法

描述 添加三角网顶点

Visual Basic li = TinModel.AppendTinPnt (triPnt)

C++ HRESULT pTinModel->AppendTinPnt (ITinPnt \*triPnt, long \*li);

参数说明：

参数	类型	出口/入口	描述
triPnt	TinPnt	[in]	三角网高程点结构信息
li	Long	[out]	成功返回点序号

#### GetTinPnt 方法

描述 获取三角网顶点

Visual Basic rtl = TinModel.GetTinPnt (li ,triPnt)

C++ HRESULT pTinModel->GetTinPnt (long li, ITinPnt \*\*triPnt,  
VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
triPnt	TinPnt	[out]	三角网高程点结构信息
li	Long	[in]	所要进行操作的三角网顶点序号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### UpdateTinPnt 方法

描述 更新三角网顶点

Visual Basic rtl = TinModel.UpdateTinPnt (li, triPnt)

C++ HRESULT pTinModel->UpdateTinPnt (long li, ITinPnt \*triPnt,  
VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
triPnt	TinPnt	[in]	三角网高程点结构信息
li	Long	[in]	所要进行操作的三角网顶点序号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UpdateTinPntZVal 方法**

描述 更新三角网顶点高程值

Visual Basic rtl = TinModel.UpdateTinPntZVal (li, zVal)

C++ HRESULT pTinModel-> UpdateTinPntZVal (long li, double zVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	所要进行操作的三角网顶点序号
zVal	Double	[in]	三角网顶点高程值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DelTinPnt 方法**

描述 删除三角网顶点

Visual Basic rtl = TinModel.DelTinPnt (li)

C++ HRESULT pTinModel-> DelTinPnt (long li, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	所要进行操作的三角网顶点序号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UnDelTinPnt 方法**

描述 恢复三角网顶点

Visual Basic rtl = TinModel.UnDelTinPnt (li)

C++ HRESULT pTinModel-> UnDelTinPnt (long li, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	所要进行操作的三角网顶点序号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetTinPntExist 方法**

描述 判断某个三角网顶点是否存在

Visual Basic pExist = TinModel.GetTinPntExist (li)

C++ HRESULT pTinModel-> GetTinPntExist (long li, short \*pExist);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	所要进行操作的三角网顶点序号

pExist	Integer	[out]	返回值 0 表示不存在,1 表示存在,-1 表示被删除
--------	---------	-------	-----------------------------

**AppendTinNet 方法**

描述 添加三角网

Visual Basic li = TinModel.AppendTinNet (triNet)

C++ HRESULT pTinModel-> AppendTinNet (ITinNet \*triNet, long \*li);

参数说明：

参数	类型	出口/入口	描述
triPnt	TinNet	[in]	三角网结构信息
li	Long	[out]	成功返回三角网序号

**GetTinNet 方法**

描述 获取三角网顶点

Visual Basic rtl = TinModel.GetTinNet (li, triNet)

C++ HRESULT pTinModel-> GetTinNet (long li, ITinNet \*\*triNet,  
VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
triNet	TinNet	[out]	三角网结构信息
li	Long	[in]	所要进行操作的三角网序号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**UpdateTinNet 方法**

描述 更新三角网顶点

Visual Basic rtl = TinModel.UpdateTinNet (li ,triNet)

C++ HRESULT pTinModel-> UpdateTinNet (long li, ITinNet \*triNet,  
VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
triNet	TinNet	[in]	三角网结构信息
li	Long	[in]	所要进行操作的三角网序号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DelTinNet 方法**

描述 删除三角网顶点

Visual Basic rtl = TinModel.DelTinNet (li)

C++ HRESULT pTinModel-> DelTinNet (long li, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	所要进行操作的三角网序号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### UnDelTinNet 方法

描述 恢复三角网顶点

Visual Basic rtl = TinModel. UnDelTinNet (li)

C++ HRESULT pTinModel-> UnDelTinNet (long li, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	所要进行操作的三角网序号
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetTinNetExist 方法

描述 判断某个三角网是否存在

Visual Basic pExist = TinModel. GetTinNetExist (li)

C++ HRESULT pTinModel-> GetTinNetExist (long li, short \*pExist);

参数说明：

参数	类型	出口/入口	描述
li	Long	[in]	所要进行操作的三角网序号
pExist	Integer	[out]	返回值 0 表示不存在, 1 表示存在, -1 表示被删除

### ClearTinNetDat 方法

描述 清除三角网拓扑结构数据

Visual Basic rtl = TinModel. ClearTinNetDat ()

C++ HRESULT pTinModel-> ClearTinNetDat (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### ClearTinNetAdjTop 方法

描述 清除三角网邻接拓扑结构

Visual Basic rtl = TinModel. ClearTinNetAdjTop ()

C++ HRESULT pTinModel-> ClearTinNetAdjTop (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ReCreateTinNetAdjTop 方法**

描述 清除三角网邻接拓扑结构

Visual Basic rtl = TinModel. ClearTinNetAdjTop ( )

C++ HRESULT pTinModel-> ClearTinNetAdjTop (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**InitCreatTinNet 方法**

描述 两步法中三角网初始构建处理

Visual Basic rtl = TinModel. InitCreatTinNet ( )

C++ HRESULT pTinModel-> InitCreatTinNet (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**OptimizeTinNet 方法**

描述 两步法中三角网优化处理

Visual Basic rtl = TinModel. OptimizeTinNet ( )

C++ HRESULT pTinModel-> OptimizeTinNet (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CreateTinNetInTinArea 方法**

描述 两步法集成生成优化三角网处理函数

Visual Basic rtl = TinModel. CreateTinNetInTinArea ( )

C++ HRESULT pTinModel-> CreateTinNetInTinArea (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CrearTinInAreaByLin 方法**

描述 TIN 中离散点在 lai 中线约束下三角网化一步处理

Visual Basic rtl = TinModel. CreatTinInAreaByLin (hWnd, plai)  
C++ HRESULT pTinModel-> CreatTinInAreaByLin (long hWnd, ILinArea \*plai, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
hWnd	Long	[in]	句柄
plai	ILinArea	[in]	线工作区对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### CreateDelaunayTin 方法

描述 由点线数据直接生成 TIN 三角网数据(此时高程工作区为空)

Visual Basic rtl = TinModel. CreateDelaunayTin ([pctlAi], [ppai], [pzk], [plai], [lzk], [ptCtlPnt], [lCtlPnt], [holDat], [holes], convexFlg, cHasNot)  
C++ HRESULT pTinModel-> CreateDelaunayTin (ILinArea \*pctlAi, IPntArea \*ppai, short pzk, ILinArea \*plai, short lzk, ID\_3DotSet \*ptCtlPnt, long lCtlPnt, ID\_DotSet \*holDat, short holes, short convexFlg, short cHasNot, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pctlAi	ILinArea	[in]	特征线(3d 线)工作区对象, 无则缺省
ppai	IPntArea	[in]	高程特征点工作区对象, 无则缺省
pzk	Integer	[in]	高程特征点的高程属性域号, 无则缺省
plai	ILinArea	[in]	等高线工作区对象, 无则缺省
lzk	Integer	[in]	等高线的高程属性域号, 无则缺省
ptCtlPnt	ID_3DotSet	[in]	控制点数据集合, 无则缺省
lCtlPnt	Long	[in]	控制点个数, 无控制点数据则缺省
holDat	ID_DotSet	[in]	在存在封闭边界的情况下指明空洞的位置坐标集合, 无则缺省
holes	Integer	[in]	每空洞一个坐标点, 表明空洞的个数, 无则缺省
convexFlg	Integer	[in]	是否约束外凸壳以避免删除(1,0 表示)
cHasNot	Integer	[in]	是否给出进度提示对话框(1,0 表示)
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### ExtractPntToTin 方法

描述 从点数据文件中提取高程点到 TIN 工作区中(TIN 工作区为空)

Visual Basic rtl = TinModel. ExtractPntToTin (ppai)



C++ HRESULT pTinModel-> ExtractPntToTin (IPntArea \*ppai, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ppai	PntArea	[in]	点工作区对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### ReduceLinToTin 方法

描述 从线数据文件中提取高程点到 TIN 工作区中(TIN 工作区为空)

Visual Basic rtl = TinModel. ReduceLinToTin (plai)

C++ HRESULT pTinModel-> ReduceLinToTin (ILinArea \*plai, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
plai	LinArea	[in]	线工作区对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### AppendTinPntByContour 方法

描述 在三角网高程数据工作区中由等值线插值三角网质心点高程

Visual Basic rtl = TinModel. AppendTinPntByContour (hWnd, plai, pmNumbFlg)

C++ HRESULT pTinModel-> AppendTinPntByContour (long hWnd, ILinArea \*plai, short pmNumbFlg, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
hWnd	Long	[in]	窗口句柄
plai	LinArea	[in]	高程等值线工作区对象
pmNumbFlg	Integer	[in]	线高程插值时采用 8/4 剖面标志 (1/0)
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### AppendTinPntByContour1 方法

描述 在三角网高程数据工作区中由等值线插值一高程点(单 TIN 点高程)

Visual Basic rtl = TinModel. AppendTinPntByContour1 (hWnd, plai, pmNumbFlg, plDot)

C++ HRESULT pTinModel-> AppendTinPntByContour1 (long hWnd, ILinArea \*plai, short pmNumbFlg, IL\_Dot \*plDot, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
hWnd	Long	[in]	窗口句柄
plai	LinArea	[in]	高程等值线工作区对象

pmNumbFlg	Integer	[in]	线高程插值时采用 8/4 剖面标志 (1/0)
plDot	L_Dot	[in]	高程点实体
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ReWriteTinAreaAllDat 方法**

描述 一次性写入非规则网高程 TIN 点及三角网数据工作区  
(数据的合法性由客户自己保证)

Visual Basic rtl = TinModel.ReWriteTinAreaAllDat (ptTriPntSet, ptTriNetSet)

C++ HRESULT pTinModel-> ReWriteTinAreaAllDat (ITinPntSet \*ptTriPntSet,  
ITinNetSet \*ptTriNetSet, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
ptTriPntSet	TinPntSet	[in]	高程 TIN 点集合(数据从 1 开始有效)
ptTriNetSet	TinNetSet	[in]	三角网数据集合(数据从 1 开始有效)
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetAllTinPntDat 方法**

描述 获取规则网所有高程点

Visual Basic rtl = TinModel.GetAllTinPntDat (Zptr )

C++ HRESULT pTinModel-> GetAllTinPntDat (VARIANT \* Zptr, VARIANT\_BOOL  
\*rtl);

参数说明：

参数	类型	出口/入口	描述
Zptr	Variant	[out]	Double 型数组, 规则网中所有高程点的高程值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetAllTinPntDat1 方法**

描述 获取非规则网所有高程点

Visual Basic pVal = TinModel.GetAllTinPntDat1 ( )

C++ HRESULT pTinModel-> GetAllTinPntDat1 (ITinPntSet \*\*pVal);

参数说明：

参数	类型	出口/入口	描述
pVal	TinPntSet	[out]	非规则网中所有高程点的集合

**GetAllTinNetDat 方法**

描述 获取所有三角网获取

Visual Basic pVal = TinModel. GetAllTinNetDat ( )

C++ HRESULT pTinModel-> GetAllTinNetDat (ITinNetSet \*\*pVal);

参数说明：

参数	类型	出口/入口	描述
pVal	TinNetSet	[out]	三角网的集合

**Bdm5To6TinCvt 方法**

描述 5.32 版 BDM 数据升级到 6.0 版的 TIN 数据

Visual Basic TinModel. Bdm5To6TinCvt

C++ HRESULT pTinModel-> Bdm5To6TinCvt ();

参数说明： 无

**GetTinConvexHull 方法**

描述 获取 TIN 工作区中离散点的凸包到线工作区

Visual Basic rtl = TinModel. GetTinConvexHull (hWnd, plai)

C++ HRESULT pTinModel-> GetTinConvexHull (long hWnd, ILinArea \*plai,  
VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
hWnd	Long	[in]	窗口句柄
plai	LinArea	[in]	结果线工作区对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetTinZFieldNo 方法**

描述 得到指定高程值的字段序号

Visual Basic pResult = TinModel. GetTinZFieldNo (plai, type)

C++ HRESULT pTinModel->GetTinZFieldNo (  
IWorkArea \*pai, short type, short \*pResult);

参数说明：

参数	类型	出口/入口	描述
plai	LinArea	[in]	点线工作区
type	Integer	[in]	属性数据域类型定义
pResult	Integer	[out]	若成功返回字段序号

**SwapEdge 方法**

**描述** 在装入数据的 Tin 工作区，交换指定点附近的，两邻接三角形构成的四边形的对角边

**Visual Basic** rtl = TinModel. SwapEdge (pnt)

**C++** HRESULT pTinModel-> SwapEdge (ID\_Dot \*pnt, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
Pnt	D_Dot	[in]	指定点
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DeleteEdge 方法**

**描述** 在装入数据的 Tin 工作区，删除指定点附近的，三角网的任一边

**Visual Basic** rtl = TinModel. DeleteEdge (pnt)

**C++** HRESULT pTinModel-> DeleteEdge (ID\_Dot \*pnt, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
Pnt	D_Dot	[in]	指定点
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DeletePartEdge 方法**

**描述** 删除指定范围内的三角网边

**Visual Basic** rtl = TinModel. DeletePartEdge (range)

**C++** HRESULT pTinModel-> DeletePartEdge (ID\_Rect \*range, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
range	D_Rect	[in]	删除范围
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**TrimTinNet 方法**

**描述** 整理三角网,从边缘起去掉狭长三角形(含对话框)

**Visual Basic** rtl = TinModel. TrimTinNet ()

**C++** HRESULT pTinModel-> TrimTinNet (VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DelInvalidTinNet 方法**

描述 删除无效三角形(指顶点中含有无效点的三角形)

Visual Basic rtl = TinModel. DelInvalidTinNet ()

C++ HRESULT pTinModel-> DelInvalidTinNet (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**9.18、GridModel 对象**

描述 GridModel 对象封装了对 Grd 模型的操作,该对象不能直接被创建，但能被 TinArea 对象内部实例化。可以通过 TinArea 对象的 pGridModel 属性或查询 TinArea 对象的 iGridModel 接口来引用 GridModel 对象。具体使用同 TinModel：

**9.18.1、属性：无****9.18.2、方法：****GetDemInsOBJ 方法**

描述 创建规则网格稀疏插密对象

Visual Basic pInsOBJ = GridModel. GetDemInsOBJ (demHdInf, varDemDat, IType, [flag], [InvalidVal])

C++ HRESULT pGridModel-> GetDemInsOBJ (IDemInfoStru \*demHdInf, VARIANT varDemDat, Enum\_TinArea\_Insert\_Type IType, VARIANT\_BOOL flag, double InvalidVal, IDemInsObj \*\*pInsOBJ);

参数说明：

参数	类型	出口/入口	描述
demHdInf	DemInfoStru	[in]	高程数据信息
varDemDat	Variant	[in]	高程数据,Double 型数组
IType	Enum_TinArea_Insert_Type	[in]	网格插密类型
flag	Boolean	[in]	有无高程无效点标志.缺省为 False,表示无
InvalidVal	Double	[in]	高程无效点值,若无用缺省值
pInsOBJ	DemInsObj	[out]	插密对象

**DeleteDemInsOBJ 方法**

描述 删除规则网格稀疏插密对象

Visual Basic rtl = GridModel. DeleteDemInsOBJ (insOBJ)

C++ HRESULT pGridModel-> DeleteDemInsOBJ (  
IDemInsObj \* insOBJ, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
insOBJ	DemInsObj	[in]	插密对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**GetCurPntDemVal 方法**

描述 取已知规则网中当前点的插密高程值

Visual Basic pResult = GridModel. GetCurPntDemVal (insOBJ, x,y)

C++ HRESULT pGridModel-> GetCurPntDemVal (  
IDemInsObj \* insOBJ, double x, double y, double \*pResult);

参数说明：

参数	类型	出口/入口	描述
insOBJ	DemInsObj	[in]	插密对象
x	Double	[in]	当前点 x 坐标
y	Double	[in]	当前点 y 坐标
pResult	Double	[out]	返回当前点的插密高程值

**InsDemDatToBuf 方法**

描述 16 或 4 点数据加密到内存缓冲区中

Visual Basic pResult = GridModel. InsDemDatToBuf(  
insOBJ, curNx, curNy, pvarZVal, rtl)

C++ HRESULT pGridModel-> InsDemDatToBuf(IDemInsObj \* insOBJ,  
long curNx, long curNy, VARIANT \*pvarZVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
insOBJ	DemInsObj	[in]	插密对象
curNx	Long	[in]	x 方向插密值
curNy	Long	[in]	y 方向插密值
pvarZVal	Variant	[out]	加密后的高程值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**InsDemDatToGrd 方法**

描述 16 或 4 点数据加密到 GRD 文件中

Visual Basic pResult = GridModel.InsDemDatToGrd (insOBJ, dx, dy, grdHf)

C++ HRESULT pGridModel-> InsDemDatToGrd (IDemInsObj \* insOBJ, double dx, double dy, int grdHf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
insOBJ	DemInsObj	[in]	插密对象
dx	Double	[in]	x 方向插密值
dy	Double	[in]	y 方向插密值
grdHf	Long	[in]	GRD 文件句柄
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Tin16Or4PointsInsertToGrd 方法**

描述 将 TIN 工作区数据插值到 GRD 或 INS 文件中

Visual Basic pResult = GridModel.Tin16Or4PointsInsertToGrd (Is16p, dx, dy, hFile)

C++ HRESULT pGridModel-> Tin16Or4PointsInsertToGrd (VARIANT\_BOOL Is16p, double dx, double dy, int hFile, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
Is16p	Boolean	[in]	是否采用 16 点双三次插值
dx	Double	[in]	x 方向插密值
dy	Double	[in]	y 方向插密值
grdHf	Long	[in]	文件句柄
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DemGridInsProc 方法**

描述 规则网 TIN 工作区插密处理

Visual Basic rtl = GridModel.DemGridInsProc ()

C++ HRESULT pGridModel-> DemGridInsProc (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**ContourLinPntGridding 方法**

描述 TIN 中离散点在 lai 中线约束下三角网化一步处理(此时高程工作区为空)

Visual Basic rtl = GridModel.ContourLinPntGridding (dmHdInf, [pctIAi], [ppai], [pzkl], [plai],

```

C++      [lzk], [ptCtlPnt,] [lCtlPnt], [holDat], [holes], convexFlg, [pclipRai], grdfname)
HRESULT pGridModel->ContourLinPntGridding (IDemInfoStru *dmHdInf,
      ILinArea *pctlAi, IPntArea *ppai, short pzk, ILinArea *plai, short lzk,
      ID_3DotSet *ptCtlPnt, long lCtlPnt, ID_DotSet *holDat, short holes,
      short convexFlg, IRegArea *pclipRai, BSTR grdfname, VARIANT_BOOL *rtl);

```

参数说明：

参数	类型	出口/入口	描述
dmHdInf	DemInfoStru	[in]	网格化参数
pctlAi	LinArea	[in]	特征线(3d 线)工作区对象，无则缺省
ppai	PntArea	[in]	高程特征点工作区对象，无则缺省
pzk	Integer	[in]	高程特征点的高程属性域号，无则缺省
plai	LinArea	[in]	等高线工作区对象，无则缺省
lzk	Integer	[in]	等高线的高程属性域号，无则缺省
ptCtlPnt	D_3DotSet	[in]	控制点数据集合，无则缺省
lCtlPnt	Long	[in]	控制点个数，无控制点数据则缺省
holDat	D_DotSet	[in]	在存在封闭边界的情况下指明空洞的位置坐标集合,无则缺省
holes	Integer	[in]	每空洞一个坐标点，表明空洞的个数,无则缺省
convexFlg	Integer	[in]	是否约束外凸壳以避免删除(1,0 表示)
pclipRai	RegArea	[in]	裁剪区工作区对象，无则缺省
grdfname	String	[in]	待生成的 GRD 文件名
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### TinDatGridding 方法

描述 由 TIN 三角网离散数据网格化生成 GRD 处理

Visual Basic rtl = GridModel.TinDatGridding (dmHdInf, grdfname)

```

C++      HRESULT pGridModel->TinDatGridding (IDemInfoStru *dmHdInf,
      BSTR grdfname,VARIANT_BOOL *rtl);

```

参数说明：

参数	类型	出口/入口	描述
dmHdInf	DemInfoStru	[in]	网格化参数
grdfname	String	[in]	待生成的 GRD 文件名
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### TinDatGriddingToBuf 方法

描述 将给定等值线网格化并将结果保存到 Double 型数组中



Visual Basic    rtl = GridModel. TinDatGriddingToBuf (dmHdInf, pvarZBuf)  
 C++            HRESULT pGridModel-> TinDatGriddingToBuf (IDemInfoStru \*dmHdInf,  
                  VARIANT \*pvarZBuf, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
dmHdInf	DemInfoStru	[in]	网格化参数
pvarZBuf	Variant	[out]	Double 型数组
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### ContourDatGridding 方法

描述            等值线高程数据栅格化处理(获取的数据存入文件)

Visual Basic    rtl = GridModel. ContourDatGridding (  
                  plai, mzk, mHdInf, pmNumbFlg, dfVsDis, fname)  
 C++            HRESULT pGridModel-> ContourDatGridding (ILinArea \*plai, short mzk,  
                  IDemInfoStru \*mHdInf, short pmNumbFlg, double dfVsDis,  
                  BSTR fname, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
plai	LinArea	[in]	高程等值线工作区对象
mzk	Integer	[in]	线属性中的高程域序号
mHdInf	DemInfoStru	[in]	网格化参数(zmin->dltX,zmax->dltY)
pmNumbFlg	Integer	[in]	线高程插值时采用 8/4 剖面标志 (1/0)
dfVsDis	Double	[in]	有效交点限制范围
fname	String	[in]	待生成的 GRD 文件名
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### ContourDatGriddingToBuf 方法

描述            等值线高程数据栅格化处理(获取的数据放入数组)

Visual Basic    rtl = GridModel. ContourDatGriddingToBuf (  
                  plai, mzk, mHdInf, pmNumbFlg, dfVsDis, fname)  
 C++            HRESULT pGridModel-> ContourDatGriddingToBuf (ILinArea \*plai, short mzk,  
                  IDemInfoStru \*mHdInf, short pmNumbFlg, double dfVsDis,  
                  VARIANT \* pvarZptr, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
plai	LinArea	[in]	高程等值线工作区对象

mzk	Integer	[in]	线属性中的高程域序号
mHdInf	DemInfoStru	[in]	网格化参数(zmin->dltX,zmax->dltY)
pmNumbFlg	Integer	[in]	线高程插值时采用 8/4 剖面标志 (1/0)
dfVsDis	Double	[in]	有效交点限制范围
pvarZptr	Varinat	[out]	获取的高程数据,Double 型数组
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

**GetGridDatByContour 方法**

描述 等值线高程数据栅格化处理

Visual Basic rtl = GridModel.GetGridDatByContour (plai, pmNumbFlg)

C++ HRESULT pGridModel-> GetGridDatByContour (ILinArea \*plai,  
short pmNumbFlg, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
plai	LinArea	[in]	高程等值线工作区对象
pmNumbFlg	Integer	[in]	线高程插值时采用 8/4 剖面标志 (1/0)
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

**GetGridDatByTin 方法**

描述 把工作区中的三角网数据网格化(弹出对话框设置网格化参数)

Visual Basic pResult = GridModel.GetGridDatByTin ()

C++ HRESULT pGridModel-> GetGridDatByTin (short \* pResult);

参数说明：

参数	类型	出口/入口	描述
pResult	Integer	[out]	1 表示成功,0 表示取消

**GetDemDatFromGrd 方法**

描述 从 GRD 文件读取高程数据

Visual Basic rtl = GridModel.GetDemDatFromGrd (demFname, nx, ny, pvarZptr)

C++ HRESULT pGridModel-> GetDemDatFromGrd (BSTR demFname, long nx,  
long ny, VARIANT \* pvarZptr, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
demFname	String	[in]	GRD 文件名
nx	Long	[in]	规则网数据行数

ny	Long	[in]	规则网数据列数
pvarZptr	Variant	[out]	高程数据,Double 型数组
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**GetDemInfoFromGrd 方法**

描述 从 GRD 文件读取高程数据

Visual Basic rtl = GridModel. GetDemInfoFromGrd (demFname, demHdInfo)

C++ HRESULT pGridModel-> GetDemInfoFromGrd (BSTR demFname,  
IDemInfoStru \*\*demHdInfo, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
demFname	String	[in]	GRD 文件名
demHdInfo	DemInfoStru	[out]	高程数据信息
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**CalMathDemGrdProc 方法**

描述 规则网数据数学运算处理

Visual Basic rtl = GridModel. CalMathDemGrdProc ()

C++ HRESULT pGridModel-> CalMathDemGrdProc (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**YieldGridDatByMathFuncTion 方法**

描述 数学方法生成 GRD

Visual Basic rtl = GridModel. YieldGridDatByMathFuncTion ()

C++ HRESULT pGridModel-> YieldGridDatByMathFuncTion (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**EditDotSearchParam 方法**

描述 编辑离散点网格化点搜索参数函数(弹出对话框)

Visual Basic rtl = GridModel. EditDotSearchParam (defRad, allDn, PtSdat)

C++ HRESULT pGridModel-> EditDotSearchParam (double defRad, long allDn,  
ISearchDataStru \*\*PtSdat, VARIANT\_BOOL \*rtl)

参数说明：

参数	类型	出口/入口	描述
defRad	Double	[in]	缺省搜索半径
allDn	Long	[in]	离散点总数
PtSdat	SearchDataStru	[in,out]	待编辑的网格化离散数据搜索参数对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**EditDistInsParam 方法**

描述 编辑离散点网格化距离幂反比处理参数 (弹出对话框)

Visual Basic rtl = GridModel. EditDistInsParam (PtDis)

C++ HRESULT pGridModel-> EditDistInsParam (IDistPowGridStru \*PtDis, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
PtDis	DistPowGridStru	[in,out]	待编辑的离散点网格化距离幂反比参数对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**DistInsGriddingToBuf 方法**

描述 将给定离散点网格化并将结果保存到 Double 型数组中

Visual Basic rtl = GridModel. DistInsGriddingToBuf (varTmpXPtr, varTmpYPtr, varTmpZPtr, allDatNum, mHdInf, PtSdat, PtDis, pvarZptr)

C++ HRESULT pGridModel-> DistInsGriddingToBuf (VARIANT varTmpXPtr, VARIANT varTmpYPtr, VARIANT varTmpZPtr, long allDatNum, IDemInfoStru \*mHdInf, ISearchDataStru \*PtSdat, IDistPowGridStru \*PtDis, VARIANT \* pvarZptr, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
varTmpXPtr	Varinat	[in]	离散数据的 X 分量,Double 型数组
varTmpYPtr	Varinat	[in]	离散数据的 Y 分量,Double 型数组
varTmpZPtr	Varinat	[in]	离散数据的 Z 分量,Double 型数组
allDatNum	Long	[in]	离散数据的总点数
mHdInf	DemInfoStru	[in]	网格化参数(其中：zmin->代表 X 向步长,zmax->代表 Y 向步长)
PtSdat	SearchDataStru	[in]	离散点搜索参数对象
PtDis	DistPowGridStru	[in]	距离幂反比处理参数对象
pvarZptr	Varinat	[out]	网格化结果,Double 型数组

rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE
-----	---------	-------	------------------------

**DistInsGridding 方法**

**描述** 离散点距离幂反比加权网格化处理函数(弹出对话框),结果保存在文件中

**Visual Basic** rtl = GridModel.DistInsGriddingToBuf (varTmpXPtr, varTmpYPtr, varTmpZPtr, allDatNum, mHdInf, PtSdat, PtDis, pvarZptr,)

**C++** HRESULT pGridModel-> DistInsGriddingToBuf (VARIANT varTmpXPtr, VARIANT varTmpYPtr, VARIANT varTmpZPtr, long allDatNum, IDemInfoStru \*mHdInf, ISearchDataStru \*PtSdat, IDistPowGridStru \*PtDis, BSTR grdFname, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
varTmpXPtr	Varinat	[in]	离散数据的 X 分量,Double 型数组
varTmpYPtr	Varinat	[in]	离散数据的 Y 分量,Double 型数组
varTmpZPtr	Varinat	[in]	离散数据的 Z 分量,Double 型数组
allDatNum	Long	[in]	离散数据的总点数
mHdInf	DemInfoStru	[in]	网格化参数(其中: zmin->代表 X 向步长,zmax->代表 Y 向步长)
PtSdat	SearchDataStru	[in]	离散点搜索参数对象
PtDis	DistPowGridStru	[in]	距离幂反比处理参数对象
grdFname	String	[in]	网格化结果生成的 GRD 文件名串
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

**KringGriddingToBuf 方法**

**描述** 将给定离散点网格化并将结果保存到 Double 型数组中

**Visual Basic** rtl = GridModel. KringGriddingToBuf (varTmpXPtr, varTmpYPtr, varTmpZPtr, allDatNum, mHdInf, PtSdat, PtKrgParm, pvarZptr,)

**C++** HRESULT pGridModel-> KringGriddingToBuf (VARIANT varTmpXPtr, VARIANT varTmpYPtr, VARIANT varTmpZPtr, long allDatNum, IDemInfoStru \*mHdInf, ISearchDataStru \*PtSdat, IKringGridStru \*PtKrgParm, VARIANT \* pvarZptr, VARIANT\_BOOL \*rtl);

**参数说明：**

参数	类型	出口/入口	描述
varTmpXPtr	Varinat	[in]	离散数据的 X 分量,Double 型数组
varTmpYPtr	Varinat	[in]	离散数据的 Y 分量,Double 型数组
varTmpZPtr	Varinat	[in]	离散数据的 Z 分量,Double 型数组
allDatNum	Long	[in]	离散数据的总点数

mHdInf	DemInfoStru	[in]	网格化参数(其中：zmin->代表 X 向步长,zmax->代表 Y 向步长)
PtSdat	SearchDataStru	[in]	离散点搜索参数对象
PtKrgParm	KringGridStru	[in]	克立金权系数插值处理参数对象
pvarZptr	Varinat	[out]	网格化结果,Double 型数组
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

### KringGridding 方法

描述 离散点距离幂反比加权网格化处理函数(弹出对话框),结果保存在文件中

Visual Basic rtl = GridModel. KringGridding (varTmpXPtr, varTmpYPtr, varTmpZPtr, allDatNum, mHdInf, PtSdat, PtKrgParm, pvarZptr,)

C++ HRESULT pGridModel-> DistInsGriddingToBuf (VARIANT varTmpXPtr, VARIANT varTmpYPtr, VARIANT varTmpZPtr, long allDatNum, IDemInfoStru \*mHdInf, ISearchDataStru \*PtSdat, IKringGridStru\* PtKrgParm, BSTR grdFname, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
varTmpXPtr	Varinat	[in]	离散数据的 X 分量,Double 型数组
varTmpYPtr	Varinat	[in]	离散数据的 Y 分量,Double 型数组
varTmpZPtr	Varinat	[in]	离散数据的 Z 分量,Double 型数组
allDatNum	Long	[in]	离散数据的总点数
mHdInf	DemInfoStru	[in]	网格化参数(其中：zmin->代表 X 向步长,zmax->代表 Y 向步长)
PtSdat	SearchDataStru	[in]	离散点搜索参数对象
PtKrgParm	IKringGridStru	[in]	克立金权系数插值处理参数对象
grdFname	String	[in]	网格化结果生成的 GRD 文件名串
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

### GetASCIIDatColumn 方法

描述 获取 ASCII 数据串中数据元的个数: 如:\_tnGetASCIIDatColumn("1 5,7,8")=4.

Visual Basic pCount = GridModel. GetASCIIDatColumn (datStr)

C++ HRESULT pGridModel-> GetASCIIDatColumn (BSTR datStr, short \*pCount);

参数说明：

参数	类型	出口/入口	描述
datStr	String	[in]	ASCII 数据串
pCount	Integer	[out]	返回 ASCII 数据串中数据元的个数

**LoadASCIIDatFileToGridding 方法**

描述 装入高程 ASCII 数据文件并网格化 (含对话框)

Visual Basic rtl = GridModel. LoadASCIIDatFileToGridding ([fname])

C++ HRESULT pGridModel-> LoadASCIIDatFileToGridding (  
BSTR fname, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	高程 ASCII 数据文件名,缺省为空
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**LoadMbDatFileToGridding 方法**

描述 装入 MB 稠密高程 ASCII 数据文件并网格化 (含对话框)

Visual Basic rtl = GridModel. LoadMbDatFileToGridding ([fname])

C++ HRESULT pGridModel-> LoadMbDatFileToGridding (  
BSTR fname, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fname	String	[in]	MB 稠密高程 ASCII 数据文件名,缺省为空
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**MakeDemDatToBMP 方法**

描述 将指定的高程数据写成 BITMAP 文件并显示

Visual Basic rtl = GridModel. MakeDemDatToBMP (PtHdInf, varZptr, UnKnowZVal, bmpFname, nGradStep, bToViewBMP)

C++ HRESULT pGridModel-> MakeDemDatToBMP (IDemInfoStru \*PtHdInf, VARIANT varZptr, double UnKnowZVal, BSTR bmpFname, short nGradStep, VARIANT\_BOOL bToViewBMP, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
PtHdInf	DemInfoStru	[in]	高程数据信息
varZptr	Variant	[in]	高程数据,Double 型数组
UnKnowZVal	double	[in]	未知高程值
bmpFname	String	[in]	BITMAP 文件名
nGradStep	Integer	[in]	灰度级别 (0-255)
bToViewBMP	Boolean	[in]	是否浏览位图文件
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**OutDemDatToGrd 方法**

描述 将指定的高程数据写成 GRD 文件

Visual Basic rtl = GridModel. OutDemDatToGrd (demHeadInfo, varDatPtr, tpHfile, InValidVal)

C++ HRESULT pGridModel-> OutDemDatToGrd (IDemInfoStru \*demHeadInfo,  
VARIANT varDatPtr, HFILE tpHfile, double InValidVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
demHeadInfo	DemInfoStru	[in]	高程数据信息
varDatPtr	Variant	[in]	高程数据,Double 型数组
tpHfile	Long	[in]	文件句柄
InValidVal	Double	[in]	无效高程值
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

**OutDemDatToBmp 方法**

描述 将指定的高程数据写成 BMP 文件

Visual Basic rtl = GridModel. OutDemDatToBmp (demHeadInfo, varDatPtr, tpHfile, InValidVal)

C++ HRESULT pGridModel->OutDemDatToBmp(IDemInfoStru \*demHeadInfo,  
VARIANT varDatPtr, HFILE tpHfile, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
demHeadInfo	DemInfoStru	[in]	高程数据信息
varDatPtr	Variant	[in]	高程数据,Double 型数组
tpHfile	Long	[in]	BMP 文件句柄
rtl	Boolean	[out]	成功返回 TRUE , 失败返回 FALSE

**SaveDemToSurfGrd 方法**

描述 将规则网数据写到指定的 SURF GRD 格式文件中

Visual Basic rtl = GridModel. SaveDemToSurfGrd (outFile, varDemDptr, frc, nx,ny,InValidVal)

C++ HRESULT pGridModel-> SaveDemToSurfGrd (BSTR outFile,  
VARIANT varDemDptr, ID\_Rect \*frc, long nx, long ny,  
double InValidVal,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
outFile	DemInfoStru	[in]	文件名
varDemDptr	Variant	[in]	高程数据,Double 型数组
frc	Long	[in]	规则网范围



nx	Long	[in]	规则网数据列数
ny	Long	[in]	规则网数据行数
InValidVal	Double	[in]	无效高程值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**InsInvalidZValue 方法**

描述 对规则网高程数据中未知点无效值进行控制加权插值

Visual Basic rtl = GridModel. InsInvalidZValue (DemInfo, pvarZVal, nGrdiRad, nQuadNum, NaN)

C++ HRESULT pGridModel-> InsInvalidZValue (IDemInfoStru \*DemInfo,  
VARIANT \* pvarZVal, short nGrdiRad, short nQuadNum,  
double NaN, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
DemInfo	DemInfoStru	[in]	高程数据信息
pvarZVal	Variant	[in,out]	规则网高程数据
nGrdiRad	Integer	[in]	插值搜索的网格单元数
nQuadNum	Integer	[in]	可插值要求的有效象限数
NaN	Double	[in]	无效高程标志值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**PoisedNokownPnt1 方法**

描述 未知点均衡化

Visual Basic rtl = GridModel. PoisedNokownPnt1 (nx, ny, xdlt, ydlt, pvarZdat, NaN)

C++ HRESULT pGridModel-> PoisedNokownPnt1 (long nx, long ny, double xdlt,  
double ydlt, VARIANT \* pvarZdat, double NaN, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
nx	Long	[in]	格网列数
ny	Long	[in]	格网行数
xdlt	Double	[in]	格网列宽
ydlt	Double	[in]	格网行宽
pvarZdat	Variant	[in]	高程数据,double 型数组
NaN	Double	[in]	未知点数据
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**PoisedNokownPnt 方法**

描述 在规则网高程工作区中对未知点均衡化

Visual Basic    rtl = GridModel. PoisedNokownPnt ()  
 C++            HRESULT pGridModel-> PoisedNokownPnt(VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### EditCurUnKnowZval 方法

描述            编辑当前未知高程标志值

Visual Basic    rtl = GridModel. EditCurUnKnowZval ()  
 C++            HRESULT pGridModel-> EditCurUnKnowZval (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### SetCurUnKnowZval 方法

描述            编辑当前未知高程标志值

Visual Basic    pResult = GridModel. SetCurUnKnowZval (unKownZval)  
 C++            HRESULT pGridModel-> SetCurUnKnowZval(  
                  double unKownZval, double \*pResult);

参数说明：

参数	类型	出口/入口	描述
unKownZval	Double	[in]	未知高程标志值
pResult	Double	[out]	返回旧的未知高程标志值

### GetCurUnKnowZval 方法

描述            获取当前未知高程标志值

Visual Basic    pResult = GridModel. GetCurUnKnowZval ()  
 C++            HRESULT pGridModel-> GetCurUnKnowZval(double \*pResult);

参数说明：

参数	类型	出口/入口	描述
pResult	Double	[out]	未知高程标志值

### SetCurUnKnowFlag 方法

描述            置未知高程标志值有效标志

Visual Basic    rtl = GridModel. SetCurUnKnowFlag (unKnowFlag)  
 C++            HRESULT pGridModel-> SetCurUnKnowFlag (  
                  VARIANT\_BOOL unKnowFlag, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
unKnowFlag	Boolean	[in]	未知高程标志值有效标志
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetCurUnKnowFlag 方法

描述 取未知高程标志值有效标志

Visual Basic pUnknowFlag = GridModel. GetCurUnKnowFlag ()

C++ HRESULT pGridModel-> GetCurUnKnowFlag (VARIANT\_BOOL \*pUnknowFlag);

参数说明：

参数	类型	出口/入口	描述
pUnknowFlag	Boolean	[out]	未知高程标志值有效标志

## 9.19、AppModel 对象

描述 AppModel 对象封装了对应用模型的操作,该对象不能直接被创建,但能被 TinArea 对象内部实例化。可以通过 TinArea 对象的 pAppModel 属性或查询 TinArea 对象的 iAppModel 接口来引用 AppModel 对象。具体使用方法同 TinModel。

### 9.19.1、属性：无

### 9.19.2、方法：

#### EditRcRange 方法

描述 编辑矩形边界范围(弹出对话框)

Visual Basic rtl = AppModel. EditRcRange (fEditRc, title)

C++ HRESULT pAppModel-> EditRcRange (  
ID\_Rect \*\*fEditRc, BSTR title, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fEditRc	D_Rect	[in,out]	矩形边界范围
title	String	[in]	对话框标题
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

#### GetPartGrdByfRc 方法

描述 由给定的 RECT 区域从规则网高程工作区提取局部 GRD 数据

Visual Basic rtl = AppModel. GetPartGrdByfRc (fEditRc, dlgEdtRc)

C++ HRESULT pAppModel-> GetPartGrdByfRc (ID\_Rect \*fGetRc,  
VARIANT\_BOOL dlgEdtRc, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fEditRc	D_Rect	[in,out]	矩形边界范围
dlgEdtRc	Boolean	[in]	是否编辑该矩形边界范围
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### ChangeGridTinDir 方法

描述 改变规则网高程数据方位(弹对话框)

Visual Basic rtl = AppModel. ChangeGridTinDir ()

C++ HRESULT pAppModel-> ChangeGridTinDir (VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### OutDemSectionAnlyDat 方法

描述 绘出剖面基线之剖面线及交点数据

Visual Basic rtl = AppModel. OutDemSectionAnlyDat (  
PtHdInf, varZVal, fdLptr, len, prlai, hSecFile, hCrFile)

C++ HRESULT pAppModel-> OutDemSectionAnlyDat (IDemInfoStru \*PtHdInf,  
VARIANT varZVal, ID\_DotSet \*fdLptr, long len, IWorkArea \*prlai,  
long hSecFile, long hCrFile, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
PtHdInf	DemInfoStru	[in]	高程数据信息
varZVal	Variant	[in]	规则网高程数据
fdLptr	ID_DotSet	[in]	剖面定位线数据
len	Long	[in]	剖面定位线中所含点个数
prlai	WorkArea	[in]	用于求交的线或区工作区
hSecFile	Long	[in]	输出剖面线插值点数据的文件句柄
hCrFile	Long	[in]	输出剖面基线交点数据的文件句柄
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### GetCrossLinByAI 方法

描述 获取线段与线或弧段之交点及交点所在线号

Visual Basic rtl = AppModel. GetCrossLinByAI (plrai, PtSegSet, DotSet, varLPtlNo, nBufSiz)

C++ HRESULT pAppModel-> GetCrossLinByAI (IWorkArea \*plrai,  
ID\_DotSet \*PtSegSet, ID\_DotSet \*\*DotSet, VARIANT \*varLPtlNo,  
short nBufSiz, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
plrai	WorkArea	[in]	线或弧段工作区
PtSegSet	D_DotSet	[in]	线段数据
DotSet	D_DotSet	[out]	输出交点坐标数据(以按距线段起点的远近排序)
varLPtlNo	Variant	[out]	输出交点所在线号数据(以按距线段起点的远近排序), 为 long 型数组
nBufSiz	Integer	[in]	指明 DotSet、varLPtlNo 所含数据个数
Rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

#### TransDemByZScale 方法

描述 变换规则网 DEM 的 Z 向高程值

Visual Basic rtl= AppModel. TransDemByZScale (PtHdInf, pvarZptr, zScal,  
[flagUnKnowZVal=False], [UnKnowZVal])

C++ HRESULT pAppModel-> TransDemByZScale (  
IDemInfoStru \*\*PtHdInf, VARIANT \* pvarZptr, double zScal,  
VARIANT\_BOOL flagUnKnowZVal=VARIANT\_FALSE, double UnKnowZVal=0,  
VARIANT\_BOOL \*rtl)

参数说明：

参数	类型	出口/入口	描述
PtHdInf	DemInfoStru	[in,out]	网格高程数据信息
pvarZptr	Variant	[in,out]	高程数据,Double 型数组,数组大小为 (PtHdInf->nx * PtHdInf->ny)
zScal	Double	[in]	z 向变换比例
flagUnKnowZVal	Boolean	[in]	标志是否有未知高程值,有未知高程值为 True,无未知高程值为 False。缺省为 False
UnKnowZVal	Double	[in]	未知高程值。若 flagUnKnowZVal = False, UnKnowZVal 的值没有意义, 可以输入也可以用缺省值; 若 flagUnKnowZVal = True, UnKnowZVal 的值表示未知高程值
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**TransDemByPrjParam 方法**

描述 投影变换规则网 DEM 的 XY 向坐标

Visual Basic rtl= AppModel. TransDemByPrjParam (PtHdInf, pvPrjTrans)

C++ HRESULT pAppModel-> TransDemByPrjParam (IDemInfoStru \*\*PtHdInf,  
IPrjTrans \*pvPrjTrans,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
PtHdInf	DemInfoStru	[in]	网格高程数据信息
pvPrjTrans	PrjTrans	[in]	投影动态库对象
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**SetTinDispParam 方法**

描述 设置显示参数函数(当重新设置显示参数后,应调用函数 TinLayer::Display()才会生效)

Visual Basic psResult = AppModel. SetTinDispParam (NetDispParam, IsGrd)

C++ HRESULT pAppModel-> SetTinDispParam (ITinNetDispStru \*\*NetDispParam,  
short IsGrd, short \*psResult);

参数说明：

参数	类型	出口/入口	描述
NetDispParam	TinNetDispStru	[out]	高程工作区显示参数, 改参数在组件内赋有缺省值
IsGrd	Integer	[in]	规则格网文件标识符若显示文件为 规则格网, IsGrd 为 1, 否则为 0
psResult	Integer	[out]	返回值: 0=失败, 1=确认, 2=取消

**ProcDemSectionAnly 方法**

描述 剖面线插值处理分析

Visual Basic rtl = AppModel. ProcDemSectionAnly (fdLptr, hWnd, plai, prai, poutlai, poutpai)

C++ HRESULT pAppModel-> ProcDemSectionAnly ( ID\_DotSet \*fdLptr, long hWnd,  
ILinArea \*plai, IRegArea \*prai, ILinArea \*poutlai,  
IPntArea \*poutpai,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
fdLptr	D_DotSet	[in]	剖面定位线数据
hWnd	Long	[in]	剖面定位线内插生成的高程表面数据
plai	LinArea	[in]	传进的线工作区

prai	RegArea	[in]	传进的区工作区
poutlai	LinArea	[in]	剖面线插值处理结果输出线工作区
poutpai	PntArea	[in]	剖面线插值处理结果输出点工作区
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CalDemWvpProc 方法**

描述 规则网高程沟脊提取计算

Visual Basic rtl = AppModel. CalDemWvpProc (PtHdInf, pvarZVal, UnKnowZVal)

C++ HRESULT pAppModel-> CalDemWvpProc ( IDemInfoStru \*PtHdInf,  
VARIANT \* pvarZVal, double UnKnowZVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
PtHdInf	DemInfoStru	[in]	高程数据信息
pvarZVal	Variant	[in,out]	规则网高程数据,Double 型数组
UnKnowZVal	Double	[in]	高程无效值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**CalDemSunMode 方法**

描述 规则网高程表面日照计算

Visual Basic rtl = AppModel. CalDemSunMode (  
PtHdInf, pvarZVal, SunType, fElAzH, fElAzA, UnKnowZVal)

C++ HRESULT pAppModel-> CalDemSunMode ( IDemInfoStru \*PtHdInf,  
VARIANT \* pvarZVal, int SunType, double fElAzH, double fElAzA,  
double UnKnowZVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
PtHdInf	DemInfoStru	[in]	高程数据信息
pvarZVal	Variant	[in,out]	Double 型数组.[in]规则网高程数据, [out]返回为各点的日照因子
SunType	Integer	[in]	0 表示 Horn 反射系数法,1 表示地表日照 度算法
fElAzH	Double	[in]	太阳位置高度角(单位:度,当 SunType=1 时有效)
fElAzA	Double	[in]	太阳位置方位角(单位:度,当 SunType=1 时有效)
UnKnowZVal	Double	[in]	高程无效值
rtl	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

### CalDemSlopeDegreeDir 方法

描述 规则网高程数据坡度、坡向及粗糙度计算处理

Visual Basic rtl = AppModel. CalDemSlopeDegreeDir (PtHdInf, pvarZptr, zScal,  
[flagUnKnowZVal=False],[UnKnowZVal]);

C++ HRESULT pAppModel-> CalDemSlopeDegreeDir (IDemInfoStru \*PtHdInf,  
VARIANT \*pvarZVal, int ProcTyp, VARIANT\_BOOL doAspectGrade,  
VARIANT\_BOOL flagUnKnowZVal=VARIANT\_FALSE, double  
UnKnowZVal=0,  
VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
PtHdInf	DemInfoStru	[in,out]	网格高程数据信息
pvarZptr	Variant	[in,out]	高程数据,Double 型数组, 数组大小为 (PtHdInf->nx * PtHdInf->ny),处理结果仍放入此数组中
ProcTyp	Long	[in]	处理类型(0/1/2/:坡度/坡向/粗糙度)
doAspectGrade	Boolean	[in]	是否对坡向进行分级处理
flagUnKnowZVal	Boolean	[in]	标志是否有未知高程值,有未知高程值为 True,无未知高程值为 False。缺省为 False
UnKnowZVal	Double	[in]	未知高程值。若 flagUnKnowZVal=False ,UnKnowZVal 的值没有意义,可以输入也可以用缺省值;若 flagUnKnowZVal=True , UnKnowZVal 的值表示未知高程值
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

### GetPntDtmParamInGrdByDlg 方法

描述 规则网高程数据坡度、坡向及粗糙度计算处理

Visual Basic rtl= AppModel. GetPntDtmParamInGrdByDlg (demHeadInfo, varValPtr)

C++ HRESULT pAppModel-> GetPntDtmParamInGrdByDlg (  
IDemInfoStru \*demHeadInfo, VARIANT varValPtr, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
demHeadInfo	DemInfoStru	[in]	网格高程数据信息



varValPtr	Variant	[in]	高程数据,Double 型数组,数组大小为 (demHeadInfo ->nx * demHeadInfo ->ny)
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

**ViewPntDtmParamInTinAI 方法**

描述 在规则网高程工作区中计算高程点实体(pos)的坡度、坡向及粗糙度

Visual Basic rtl= AppModel. ViewPntDtmParamInTinAI (pos)

C++ HRESULT pAppModel-> ViewPntDtmParamInTinAI (ID\_Dot \*pos, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
pos	ID_Dot	[in]	高程点实体坐标
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

**GetPntDtmParamInGrd 方法**

描述 获取规则网表面模型数据中指定单点的高程、坡度、坡向

Visual Basic rtl= AppModel. GetPntDtmParamInGrd (PtHdInf, varDat, pNo, dtmParm)

C++ HRESULT pAppModel-> GetPntDtmParamInGrd (IDemInfoStru \*PtHdInf, VARIANT varDat, IL\_Dot \*pNo, ID\_3Dot \*\*dtmParm, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
PtHdInf	DemInfoStru	[in]	网格高程数据信息
varDat	Variant	[in]	高程数据,Double 型数组,数组大小为 (PtHdInf ->nx * PtHdInf ->ny)
pNo	IL_Dot	[in]	指定点索引下标(x,y)
dtmParm	ID_3Dot	[out]	返回值(x/y/z:高程/坡度/坡向)
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

**GridNetSurfaceVpProc 方法**

描述 规则网表面沟脊提取模型(弹出参数编辑对话框)

Visual Basic AppModel. GridNetSurfaceVpProc

C++ HRESULT pAppModel-> GridNetSurfaceVpProc ();

参数说明： 无

**GridNetSurfacevParOut 方法**

描述 规则网表面参数提取模型(弹出参数编辑对话框)

Visual Basic AppModel. GridNetSurfacevParOut  
C++ HRESULT pAppModel-> GridNetSurfacevParOut ();  
参数说明： 无

### GridNetSurfaceSunModel 方法

描述 规则网表面日照处理模型(弹出参数编辑对话框)  
Visual Basic AppModel. GridNetSurfaceSunModel  
C++ HRESULT pAppModel-> GridNetSurfaceSunModel ();  
参数说明： 无

### TinGridNet3dGraphByBuf 方法

描述 给定数据绘制三维立体图(含对话框)  
Visual Basic rtl = AppModel. TinGridNet3dGraphByBuf (PtHdInf, varZVal, plai, ppai, flagUnKnowZVal, UnKnowZVal)  
C++ HRESULT pAppModel-> TinGridNet3dGraphByBuf (IDemInfoStru \*PtHdInf, VARIANT varZVal, ILineArea \*plai, IPntArea \*ppai, VARIANT\_BOOL flagUnKnowZVal, double UnKnowZVal, VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
PtHdInf	DemInfoStru	[in]	网格高程数据信息
varZVal	Variant	[in]	高程数据,Double 型数组,数组大小为(PtHdInf ->nx * PtHdInf ->ny)
plai	LineArea	[in]	线工作区对象(传入时为空的线工作区;方法完成后,该工作区含有线数据,是立体图的一部分)
ppai	PntArea	[in]	点工作区对象(传入时为空的点工作区;方法完成后,该工作区含有点数据,是立体图的一部分)
flagUnKnowZVal	Boolean	[in]	标志是否有未知高程值,有未知高程值为 True,无未知高程值为 False。缺省为 False
UnKnowZVal	Double	[in]	未知高程值。若 flagUnKnowZVal = False , UnKnowZVal 的值没有意义,可以输入也可以用缺省值;若 flagUnKnowZVal = True , UnKnowZVal 的值表示未知高程值
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

**TinGridNet3dEquGraByBuf 方法**

描述 由给定网格高程数据绘制三维等值立体图(含对话框)

Visual Basic rtl= AppModel. TinGridNet3dEquGraByBuf (PtHdInf, varZVal,  
plai, ppai , prai, flagUnKnowZVal, UnKnowZVal)

C++ HRESULT pAppModel-> TinGridNet3dEquGraByBuf (IDemInfoStru \*PtHdInf,  
VARIANT varZVal, ILinArea \*plai, IPntArea \*ppai, IRegArea \*prai,  
VARIANT\_BOOL flagUnKnowZVal, double UnKnowZVal,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
PtHdInf	DemInfoStru	[in]	网格高程数据信息
varZVal	Variant	[in]	高程数据,Double 型数组,数组大小为(PtHdInf ->nx * PtHdInf ->ny)
plai	LinArea	[in]	线工作区对象(传入时为空的线工作区;方法完成后,该工作区含有线数据,是三维等值立体图的一部分)
ppai	PntArea	[in]	点工作区对象(传入时为空的点工作区;方法完成后,该工作区含有点数据,是三维等值立体图的一部分)
prai	RegArea	[in]	区工作区对象(传入时为空的区工作区;方法完成后,该工作区含有区数据,是三维等值立体图的一部分)
flagUnKnowZVal	Boolean	[in]	标志是否有未知高程值,有未知高程值为 True,无未知高程值为 False。缺省为 False
UnKnowZVal	Double	[in]	未知高程值。若 flagUnKnowZVal = False , UnKnowZVal 的值没有意义,可以输入也可以用缺省值;若 flagUnKnowZVal = True , UnKnowZVal 的值表示未知高程值
rtl	Boolean	[out]	成功返回 TRUE,失败返回 FALSE

**TinGridNet3dGraph 方法**

描述 在规则网高程工作区中绘制三维立体图(含对话框)

Visual Basic rtl = AppModel. TinGridNet3dGraph (plai, ppai )

C++ HRESULT pAppModel-> TinGridNet3dGraph (ILinArea \*plai, IPntArea \*ppai,  
VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
plai	LinArea	[in]	线工作区对象(传入时为空的线工作区;方法完成后,该工作区含有线数据,是立体图的一部分)
ppai	PntArea	[in]	点工作区对象(传入时为空的点工作区;方法完成后,该工作区含有点数据,是立体图的一部分)
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

### TinGridNet3dEquGraByBuf 方法

描述 在规则网高程工作区中绘制三维等值立体图

Visual Basic rtl = AppModel.TinGridNet3dEquGraByBuf (plai, ppai, prai)

C++ HRESULT pAppModel-> TinGridNet3dEquGraByBuf (ILinArea \*plai, IPntArea \*ppai, IRegArea \*prai, VARIANT\_BOOL \*rtl)

参数说明：

参数	类型	出口/入口	描述
plai	LinArea	[in]	线工作区对象(传入时为空的线工作区;方法完成后,该工作区含有线数据,是三维等值立体图的一部分)
ppai	PntArea	[in]	点工作区对象(传入时为空的点工作区;方法完成后,该工作区含有点数据,是三维等值立体图的一部分)
prai	RegArea	[in]	区工作区对象(传入时为空的区工作区;方法完成后,该工作区含有区数据,是三维等值立体图的一部分)
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

### ExtractDemAspectReg 方法

描述 提取指定单一范围的坡向区域

Visual Basic rtl= AppModel. ExtractDemAspectReg (prai)

C++ HRESULT pAppModel-> ExtractDemAspectReg ( IRegArea \*prai,VARIANT\_BOOL \*rtl);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

prai	RegArea	[in]	区工作区对象(传入时为空的区工作区;方法完成后,该工作区含有区数据,是坡向区域图)
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**GetDemSeeperAreaGrd 方法**

描述 获取规则网高程数据的积流 GRD 数据 (数据由 PtzVal 带出)

Visual Basic rtl = TinArea. GetDemSeeperAreaGrd (

dHdInf, pvarZVal, dZLimit, falgUnKnowZVal, UnKnowZVal)

C++ HRESULT pTinArea-> GetDemSeeperAreaGrd (IDemInfoStru \*dHdInf,  
VARIANT \*pvarZVal, double dZLimit, VARIANT\_BOOL falgUnKnowZVal,  
double UnKnowZVal, VARIANT\_BOOL \*rtl)

参数说明：

参数	类型	出口/入口	描述
DHdInf	DemInfoStru	[in]	规则网高程数据信息
pvarZVal	Variant	[in,out]	规则网高程数据,Double 型数组,大小为 dHdInf->nx* dHdInf->ny;同时返回结果数据
dZLimit	Double	[in]	积流阈值
falgUnKnowZVal	Boolean	[in]	标志是否有未知高程值,有未知高程值为 True,无未知高程值为 False。缺省为 False
UnKnowZVal	Double	[in]	未知高程值。若 flagUnKnowZVal = False, UnKnowZVal 的值没有意义,可以输入也可以用缺省值;若 flagUnKnowZVal = True, UnKnowZVal 的值表示未知高程值
rtl	Boolean	[out]	成功返回 TRUE, 失败返回 FALSE

**9.20、TinLayer 对象**

描述 TinLayer 对象封装了对 Tin 图层的基本操作。

**9.20.1、属性：**

属性	描述
----	----

LayerType	工作区类型
Enum_Area_Type	
只读	
Rect	工作区范围
D_Rect	
只读	
MinDispRate	最小显示比例
Double	
可读写	
MaxDispRate	最大显示比例
Double	
可读写	
Describe	文件描述
String	
可读写	
Visible	是否可见标志
Boolean	
可读写	
WorkArea	取 WorkArea
WorkArea	
只读	
Tag	附加数据
Long	
可读写	
ModifyFlag	修改标记
Intger	
可读写	
TinArea	取 TinArea
TinArea	
只读	

## 9.20.2、方法：

### Display 方法

描述 显示三角点或三角网的高程数据

Visual Basic AppModel.Display pMydc

C++ HRESULT pAppModel-> Display (IMapGisDC \*pMydc);

参数说明：

参数	类型	出口/入口	描述
pMydc	MapGisDC	[in]	MapGis 显示对象

**Attach 方法**

描述 将 WorkArea 与 Layer 对象绑定

Visual Basic sflg = AppModel.Attach (pIWorkArea)

C++ HRESULT pAppModel-> Attach (IWorkArea \*pIWorkArea, VARIANT\_BOOL \*sflg);

参数说明：

参数	类型	出口/入口	描述
pIWorkArea	WorkArea	[in]	WorkArea 对象
sflg	Boolean	[out]	成功返回 TRUE，失败返回 FALSE

**Detach 方法**

描述 解除 WorkArea 与 Layer 对象的绑定

Visual Basic sflg = AppModel.Detach (pIWorkArea)

C++ HRESULT pAppModel-> Detach (  
IWorkArea \*pIWorkArea, VARIANT\_BOOL \*sflg);

参数说明：

参数	类型	出口/入口	描述
pIWorkArea	WorkArea	[in]	WorkArea 对象

## 第十章 可视化控件说明

本章的示例代码包含在 D:\MAPGIS61\VBDemo\vbW60Area 及 D:\MAPGIS61\VBDemo\Controls 目录下。

### 10.1、EditView 控件

**描述** EditView 控件对象是实现地图显示、操作、编辑（包括点编辑、线编辑、区编辑、数字化、矢量化等）及其他操作功能的控件。

#### 10.1.1、属性：

属性	描述
PopupMenuEnable Integer 可读写	是否显示已有的右键菜单(1/0-显示/不显示)
PopupMenuControl Integer 可读写	右键弹出菜单项控制： -1--不显示菜单 0--全部显示 1--显示窗口操作 2--1+显示点、线等 3--2+显示工具箱，编辑等
CommandFlag Long 可读写	命令操作标记
Map Map 可读写	Map 对象
pMapGisDC MapGisDC 只读	显示设备对象
BackgroundColor Long 可读写	取/设背景色
PntArea WorkArea 可读写	取/设点工作区，控件若有得到此工作区接口，便可以 进行显示、编辑、选择操作。



LinArea	取/设线工作区，控件若有得到此工作区接口，便可以进行显示、编辑、选择操作。
WorkArea	
可读写	
RegArea	取/设区工作区，控件若有得到此工作区接口，便可以进行显示、编辑、选择操作。
WorkArea	
可读写	
NetArea	取/设网工作区，控件若有得到此工作区接口，便可以进行显示、编辑、选择操作。
WorkArea	
可读写	
DspArc	是否弧段可见(True/False-可见/不可见)
BOOL	
可读写	
DspOrg	是否还原显示(True/False-还原显示/不还原显示)
BOOL	
可读写	
DspBigCross	是否显示大十字光标(True/False-显示/不显示)
BOOL	
可读写	
DspCoordPnt	坐标点是否可见(True/False-可见/不可见)
BOOL	
可读写	
DefCurType	缺省的光标类型：
Integer	0—小十字型光标
可读写	1—普通箭头光标
	2—带问号箭头光标
	3—带方向箭头光标
Paralldis	平行线距离
Double	
可读写	
CleanRadius	结点搜索半径
Double	
可读写	
BufferRadius	Buffer分析半径
Double	
可读写	

SmoothRadius	插密光滑半径
Double	
可读写	
LimitRadius	坐标点间最小距离
Double	
可读写	
ReduceRate	抽稀半径
Double	
可读写	
ViewAttElementType	浏览工作区属性的实体类型
Enum_Entity_Type	
只读	
LinkAttSwitch	浏览工作区属性联动标志(True-联动, False-不联动)
BOOL	
可读写	
ErrReportSwitch	错误信息显示开关(True-显示错误信息, False-不显示)
BOOL	
可读写	
SelectSet	选择集合对象
SelectSet	
可读写	

## 10.1.2、方法：

### 一、 通用操作

#### SetExtendTool 方法

描述 设置编辑操作对象。

Visual Basic EditView.SetExtendTool pExtTool

C++ HRESULT EditView -> SetExtendTool (IDispatch\* pExtTool);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

pExtTool	IDispatch	[in]	<p>编辑操作对象</p> <p>必须满足的条件是接口有以下方法和事件：</p> <ol style="list-style-type: none"> <li>1. LButtonDown(nFlags As Integer, x As Integer, y As Integer)</li> <li>2. MouseMove (nFlags As Integer, x As Integer, y As Integer)</li> <li>3. LButtonUp (nFlags As Integer, x As Integer, y As Integer)</li> <li>4. RButtonUp (nFlags As Integer, x As Integer, y As Integer)</li> <li>5. OnKeyDown(nChar As Integer, nFlagsr As Integer)</li> <li>6. DrawGraphics(mdc As MapGisDC)</li> <li>7. DrawBackGround(mdc As MapGisDC)</li> <li>8. Running()</li> <li>9. Reset()</li> <li>10. 事件：Finished()</li> </ol>
----------	-----------	------	---

**AttachAttWorkArea 方法**

描述 设置属性浏览的工作区对象。

Visual Basic EditView.AttachAttWorkArea pWorkArea, [entityVal]

C++ HRESULT EditView->AttachAttWorkArea(IWorkArea\*pWorkArea, [optional]Enum\_Entity\_Type entityVal);

参数说明：

参数	类型	出口/入口	描述
pWorkArea	WorkArea	[in]	属性浏览的工作区
entityVal	Enum_Entity_Type	[in,optional]	实体类型（可选参数）

**GotoElement 方法**

描述 跳转到指定的单个图元。

Visual Basic EditView.GotoElement pWorkArea, IElementNo, entityVal

C++ HRESULT EditView -> GotoElement ( IWorkArea\* pWorkArea, long IElementNo, Enum\_Entity\_Type entityVal);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

pWorkArea	WorkArea	[in]	跳转的工作区
lElementNo	Long	[in]	跳转的图元号
entityVal	Enum_Entity_Type	[in]	跳转的图元类型

**FlashElement 方法**

描述 闪烁指定选择集的图元

Visual Basic EditView.FlashElement pSelectSet

C++ HRESULT EditView -> FlashElement (ISelectSet\* pSelectSet);

参数说明：

参数	类型	出口/入口	描述
pSelectSet	SelectSet	[in]	闪烁的选择集

**StopFlash 方法**

描述 停止所有图元闪烁。

Visual Basic EditView.StopFlash

C++ HRESULT EditView -> StopFlash ();

**RestoreWindow 方法**

描述 复位图形窗口。

Visual Basic EditView.RestoreWindow

C++ HRESULT EditView -> RestoreWindow ();

**SetWinMapRange 方法**

描述 设置图形范围。

Visual Basic EditView.SetWinMapRange Xmin, Ymin, Xmax, Ymax

C++ HRESULT EditView -> SetWinMapRange (  
double Xmin, double Ymin, double Xmax, double Ymax) ;

参数说明：

参数	类型	出口/入口	描述
Xmin	Double	[in]	图形范围的 xmin
Ymin	Double	[in]	图形范围的 ymin
Xmax	Double	[in]	图形范围的 xmax
Ymax	Double	[in]	图形范围的 ymax

**GetWinMapRange 方法**

描述 取图形范围。

Visual Basic EditView.GetWinMapRange (Xmin, Ymin, Xmax, Ymax)

C++ HRESULT EditView -> GetWinMapRange (  
double \*Xmin, double \*Ymin, double \*Xmax, double \*Ymax) ;

参数说明：

参数	类型	出口/入口	描述
Xmin	Double	[out]	图形范围的 xmin
Ymin	Double	[out]	图形范围的 ymin
Xmax	Double	[out]	图形范围的 xmax
Ymax	Double	[out]	图形范围的 ymax

### SetDispParm 方法

描述 设置图形显示参数。

Visual Basic EditView.SetDispParm xorg, yorg, scale

C++ HRESULT EditView -> SetDispParm (  
double xorg, double yorg, double scale);

参数说明：

参数	类型	出口/入口	描述
xorg	Double	[in]	图形显示的原点 x
yorg	Double	[in]	图形显示的原点 y
scale	Double	[in]	图形显示的比例

### GetDispParm 方法

描述 取图形显示参数。

Visual Basic EditView.GetDispParm (xorg, yorg, scale)

C++ HRESULT EditView -> GetDispParm (  
double \*xorg, double \*yorg, double \*scale);

参数说明：

参数	类型	出口/入口	描述
xorg	Double	[out]	图形显示的原点 x
yorg	Double	[out]	图形显示的原点 y
scale	Double	[out]	图形显示的比例

### LpToDp 方法

描述 逻辑坐标转换设备坐标。

Visual Basic EditView.LpToDp (xSource, ySource, xDest, yDest)

C++ HRESULT EditView -> LpToDp (  
double xSource, double ySource, VARIANT\* xDest, VARIANT\* yDest) ;

参数说明：

参数	类型	出口/入口	描述
xSource	Double	[in]	逻辑坐标的 x
ySource	Double	[in]	逻辑坐标的 y
xDest	Double	[out]	设备坐标的 x
yDest	Double	[out]	设备坐标的 y

**LpToWp 方法**

描述 逻辑坐标转换窗口坐标。

Visual Basic EditView.LpToWp (xSource, ySource, xDest, yDest)

C++ HRESULT EditView -> LpToWp (  
double xSource, double ySource, VARIANT\* xDest, VARIANT\* yDest) ;

参数说明：

参数	类型	出口/入口	描述
xSource	Double	[in]	逻辑坐标的 x
ySource	Double	[in]	逻辑坐标的 y
xDest	Double	[out]	窗口坐标的 x
yDest	Double	[out]	窗口坐标的 y

**WpToDp 方法**

描述 窗口坐标转换设备坐标。

Visual Basic EditView.WpToDp (xSource, ySource, xDest, yDest)

C++ HRESULT EditView -> WpToDp (  
double xSource, double ySource, VARIANT\* xDest, VARIANT\* yDest) ;

参数说明：

参数	类型	出口/入口	描述
xSource	Double	[in]	窗口坐标的 x
ySource	Double	[in]	窗口坐标的 y
xDest	Double	[out]	设备坐标的 x
yDest	Double	[out]	设备坐标的 y

**WpToLp 方法**

描述 窗口坐标转换逻辑坐标。

Visual Basic EditView.WpToLp (xSource, ySource, xDest, yDest)

C++ HRESULT EditView -> WpToLp (  
double xSource, double ySource, VARIANT\* xDest, VARIANT\* yDest) ;

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

xSource	Double	[in]	窗口坐标的 x
ySource	Double	[in]	窗口坐标的 y
xDest	Double	[out]	逻辑坐标的 x
yDest	Double	[out]	逻辑坐标的 y

**DpToLp 方法**

描述 设备坐标转换逻辑坐标。

Visual Basic EditView.DpToLp (xSource, ySource, xDest, yDest)

C++ HRESULT EditView -> DpToLp (  
double xSource, double ySource, VARIANT\* xDest, VARIANT\* yDest);

参数说明：

参数	类型	出口/入口	描述
xSource	Double	[in]	设备坐标的 x
ySource	Double	[in]	设备坐标的 y
xDest	Double	[out]	逻辑坐标的 x
yDest	Double	[out]	逻辑坐标的 y

**DpToWp 方法**

描述 设备坐标转换窗口坐标。

Visual Basic EditView.DpToWp (xSource, ySource, xDest, yDest)

C++ HRESULT EditView -> DpToWp (  
double xSource, double ySource, VARIANT\* xDest, VARIANT\* yDest);

参数说明：

参数	类型	出口/入口	描述
xSource	Double	[in]	设备坐标的 x
ySource	Double	[in]	设备坐标的 y
xDest	Double	[out]	窗口坐标的 x
yDest	Double	[out]	窗口坐标的 y

**msgSelectLine 方法**

描述 选择线交互操作，并且得到线实体的结果集。

Visual Basic EditView.msgSelectLine (msg, nFlag, x, y)

C++ short EditView -> msgSelectLine (  
MOUSE\_MESSAGE msg, short nFlag, short x, short y);

参数说明：

参数	类型	出口/入口	描述
----	----	-------	----

msg	MOUSE_MESSAGE	[in]	鼠标消息标志码 L_DOWN: 鼠标左键按下 L_UP: 鼠标左键弹起 R_DOWN: 鼠标右键按下 R_UP: 鼠标右键弹起 MOVE: 鼠标移动
nFlag	Interger	[in]	鼠标的控制码
x	Interger	[in]	窗口坐标的 x
y	Interger	[in]	窗口坐标的 y
ret	Interger	[out,retval]	>0 表示交互操作结束， <=0 表示交互操作在进行时

### msgSelectPnt 方法

**描述** 选择点交互操作，并且得到点实体的结果集。

Visual Basic EditView.msgSelectPnt (msg, nFlag, x, y)

C++ short EditView -> msgSelectPnt (  
MOUSE\_MESSAGE msg, short nFlag, short x, short y);

**参数说明：**

参数	类型	出口/入口	描述
msg	MOUSE_MESSAGE	[in]	鼠标消息标志码 L_DOWN: 鼠标左键按下 L_UP: 鼠标左键弹起 R_DOWN: 鼠标右键按下 R_UP: 鼠标右键弹起 MOVE: 鼠标移动
nFlag	Interger	[in]	鼠标的控制码
x	Interger	[in]	窗口坐标的 x
y	Interger	[in]	窗口坐标的 y
ret	Interger	[out,retval]	>0 表示交互操作结束，<=0 表示交互操作在进行时

### msgSelectReg 方法

**描述** 选择区交互操作，并且得到区实体的结果集。

Visual Basic EditView.msgSelectReg (msg, nFlag, x, y)

C++ short EditView -> msgSelectReg (  
MOUSE\_MESSAGE msg, short nFlag, short x, short y);

**参数说明：**



参数	类型	出口/入口	描述
msg	MOUSE_MESSAGE	[in]	鼠标消息标志码 L_DOWN: 鼠标左键按下 L_UP: 鼠标左键弹起 R_DOWN: 鼠标右键按下 R_UP: 鼠标右键弹起 MOVE: 鼠标移动
nFlag	Interger	[in]	鼠标的控制码
x	Interger	[in]	窗口坐标的 x
y	Interger	[in]	窗口坐标的 y
ret	Interger	[out,retval]	>0 表示交互操作结束, <=0 表示交互操作在进行时

**msgSelectAll 方法**

描述 选择实体交互操作, 并且得到实体的结果集。

Visual Basic EditView.msgSelectAll (msg, nFlag, x, y)

C++ short EditView -> msgSelectAll (  
MOUSE\_MESSAGE msg, short nFlag, short x, short y);

参数说明:

参数	类型	出口/入口	描述
msg	MOUSE_MESSAGE	[in]	鼠标消息标志码 L_DOWN: 鼠标左键按下 L_UP: 鼠标左键弹起 R_DOWN: 鼠标右键按下 R_UP: 鼠标右键弹起 MOVE: 鼠标移动
nFlag	Interger	[in]	鼠标的控制码
x	Interger	[in]	窗口坐标的 x
y	Interger	[in]	窗口坐标的 y
ret	Interger	[out,retval]	>0 表示交互操作结束, <=0 表示交互操作在进行时

**msgFindAReg 方法**

描述 查找一个区实体, 并且得到区实体。

Visual Basic sel = EditView.msgFindAReg (x, y)

C++ ISelectItem \*EditView -> msgFindAReg (short x, short y);

参数说明:

参数	类型	出口/入口	描述
x	Integer	[in]	窗口坐标的 x
y	Integer	[in]	窗口坐标的 y
sel	SelectItem	[out,retval]	返回拾取到的实体号和对的工作区

**msgFindALin 方法**

描述 查找一个线实体，并且得到线实体。

Visual Basic sel = EditView.msgFindALin (x, y)

C++ ISelectItem \*EditView -> msgFindALin (short x, short y);

参数说明：

参数	类型	出口/入口	描述
x	Integer	[in]	窗口坐标的 x
y	Integer	[in]	窗口坐标的 y
sel	SelectItem	[out,retval]	返回拾取到的实体号和对的工作区

**msgFindAPnt 方法**

描述 查找一个点实体，并且得到点实体。

Visual Basic EditView.msgFindAPnt(x, y)

C++ ISelectItem \*EditView -> msgFindAPnt (short x, short y);

参数说明：

参数	类型	出口/入口	描述
x	Integer	[in]	窗口坐标的 x
y	Integer	[in]	窗口坐标的 y
sel	SelectItem	[out,retval]	返回拾取到的实体号和对的工作区

**二、 点编辑操作****MakePoint 方法**

描述 输入点。

Visual Basic EditView.MakePoint

C++ HRESULT EditView -> MakePoint();

**DeletePoint 方法**

描述 删除点。

Visual Basic EditView.DeletePoint

C++ HRESULT EditView -> DeletePoint();

**MovePoint 方法**

描述	移动点。
Visual Basic	EditView.MovePoint
C++	HRESULT EditView -> MovePoint ();

**MovePointByKey 方法**

描述	移动点坐标调整。
Visual Basic	EditView.MovePointByKey
C++	HRESULT EditView -> MovePointByKey ();

**CopyPoint 方法**

描述	复制点。
Visual Basic	EditView.CopyPoint
C++	HRESULT EditView -> CopyPoint ();

**ArrayCopyPoint 方法**

描述	阵列复制点。
Visual Basic	EditView.ArrayCopyPoint
C++	HRESULT EditView -> ArrayCopyPoint ();

**PositionPoint 方法**

描述	定位点。
Visual Basic	EditView.PositionPoint
C++	HRESULT EditView -> PositionPoint ();

**RulePoint 方法**

描述	对齐坐标。
Visual Basic	EditView.RulePoint
C++	HRESULT EditView -> RulePoint ();

**ClipString 方法**

描述	剪断字符串。
Visual Basic	EditView.ClipString
C++	HRESULT EditView -> ClipString ();

**JoinString 方法**

描述	连接字符串。
Visual Basic	EditView.JoinString

C++          HRESULT EditView -> JoinString ();

#### **UpdateImage 方法**

描述          修改图像  
Visual Basic   EditView.UpdateImage  
C++          HRESULT EditView -> UpdateImage ();

#### **UpdateWord 方法**

描述          修改文本  
Visual Basic   EditView.UpdateWord  
C++          HRESULT EditView -> UpdateWord ();

#### **UpdateAngle 方法**

描述          修改角度  
Visual Basic   EditView.UpdateAngle  
C++          HRESULT EditView -> UpdateAngle ();

#### **UpdatePointParm 方法**

描述          修改点参数。  
Visual Basic   EditView.UpdatePointParm  
C++          HRESULT EditView -> UpdatePointParm ();

#### **UpdatePointAtt 方法**

描述          修改点属性。  
Visual Basic   EditView.UpdatePointAtt  
C++          HRESULT EditView -> UpdatePointAtt ();

#### **EditPntAttStru 方法**

描述          编辑点属性结构  
Visual Basic   EditView.EditPntAttStru  
C++          HRESULT EditView -> EditPntAttStru ();

#### **ModPntAttByParm 方法**

描述          根据参数赋属性  
Visual Basic   EditView.ModPntAttByParm  
C++          HRESULT EditView -> ModPntAttByParm ();

**ModPntParmByAtt 方法**

描述	根据属性赋参数
Visual Basic	EditView. ModPntParmByAtt
C++	HRESULT EditView -> ModPntParmByAtt ();

**MadeNoteByAtt 方法**

描述	根据属性标注释
Visual Basic	EditView. MadeNoteByAtt
C++	HRESULT EditView -> MadeNoteByAtt ();

**AddNoteToAtt 方法**

描述	注释赋为属性
Visual Basic	EditView. AddNoteToAtt
C++	HRESULT EditView -> AddNoteToAtt ();

**EditGivePoint 方法**

描述	编辑指定点。
Visual Basic	EditView.EditGivePoint
C++	HRESULT EditView -> EditGivePoint ();

**DefaultPointParm 方法**

描述	缺省点参数。
Visual Basic	EditView.DefaultPointParm
C++	HRESULT EditView -> DefaultPointParm ();

**ReplacePointParm 方法**

描述	替换点参数。
Visual Basic	EditView.ReplacePointParm
C++	HRESULT EditView -> ReplacePointParm ();

**SelectPoint 方法**

描述	选择点参数。
Visual Basic	EditView.SelectPoint
C++	HRESULT EditView -> SelectPoint ();

**SelectPoint 方法**

描述	选择点参数。
Visual Basic	EditView. SelectPoint

---

C++      HRESULT EditView -> SelectPoint ();

### 三、 线编辑操作

#### EditGiveLine 方法

描述          编辑给定线  
Visual Basic   EditView. EditGiveLine  
C++          HRESULT EditView-> EditGiveLine ();

#### MakeLineByKey 方法

描述          键盘输入线  
Visual Basic   EditView. MakeLineByKey  
C++          HRESULT EditView-> MakeLineByKey ();

#### ReplaceLineParm 方法

描述          统改线参数  
Visual Basic   EditView. ReplaceLineParm  
C++          HRESULT EditView-> ReplaceLineParm ();

#### DefaultLineParm 方法

描述          修改缺省线参数  
Visual Basic   EditView. DefaultLineParm  
C++          HRESULT EditView-> DefaultLineParm ();

#### EditLineAttStru 方法

描述          修改线属性结构  
Visual Basic   EditView. EditLineAttStru  
C++          HRESULT EditView-> EditLineAttStru ();

#### MakeLine 方法

描述          输入线  
Visual Basic   EditView. MakeLine  
C++          HRESULT EditView-> MakeLine ();

#### DeleteLine 方法

描述          删除线  
Visual Basic   EditView. DeleteLine  
C++          HRESULT EditView-> DeleteLine ();

**MoveLine 方法**

描述	移动线
Visual Basic	EditView. MoveLine
C++	HRESULT EditView-> MoveLine ();

**MoveGroupLineByKey 方法**

描述	移动线坐标调整
Visual Basic	EditView. MoveGroupLineByKey
C++	HRESULT EditView-> MoveGroupLineByKey ();

**MoveObjectByDist**

描述	推移线
Visual Basic	EditView.MoveObjectByDist
C++	HRESULT EditView-> MoveObjectByDist ();

**CopyLine 方法**

描述	复制线
Visual Basic	EditView. CopyLine
C++	HRESULT EditView-> CopyLine ();

**ArrayCopyLine 方法**

描述	阵列复制线
Visual Basic	EditView. ArrayCopyLine
C++	HRESULT EditView-> ArrayCopyLine ();

**BandLine 方法**

描述	钝化线
Visual Basic	EditView. BandLine
C++	HRESULT EditView-> BandLine ();

**ChainLine 方法**

描述	联接线
Visual Basic	EditView. ChainLine
C++	HRESULT EditView-> ChainLine ();

**LongLine 方法**

描述	延长线
----	-----

---

Visual Basic	EditView. LongLine
C++	HRESULT EditView-> LongLine ();

**ShortLine 方法**

描述	缩短线
Visual Basic	EditView. ShortLine
C++	HRESULT EditView-> ShortLine ();

**NearToLine 方法**

描述	靠近线
Visual Basic	EditView. NearToLine
C++	HRESULT EditView-> NearToLine ();

**AddLineDot 方法**

描述	线上加点
Visual Basic	EditView. AddLineDot
C++	HRESULT EditView-> AddLineDot ();

**DelLineDot 方法**

描述	线上删点
Visual Basic	EditView. DelLineDot
C++	HRESULT EditView-> DelLineDot ();

**MoveLineDot 方法**

描述	鼠标线上移点
Visual Basic	EditView. MoveLineDot
C++	HRESULT EditView-> MoveLineDot ();

**MoveLineDotContinue 方法**

描述	鼠标连续移点
Visual Basic	EditView. MoveLineDotContinue
C++	HRESULT EditView-> MoveLineDotContinue ();

**KeyMoveLineDot 方法**

描述	键盘线上移点
Visual Basic	EditView. KeyMoveLineDot
C++	HRESULT EditView-> KeyMoveLineDot ();



**MakeLineParall 方法**

描述	造平行线
Visual Basic	EditView. MakeLineParall
C++	HRESULT EditView-> MakeLineParall ();

**SmoothLineSegment 方法**

描述	分段光滑线
Visual Basic	EditView. SmoothLineSegment
C++	HRESULT EditView-> SmoothLineSegment ();

**SmoothGroupLine 方法**

描述	光滑线
Visual Basic	EditView. SmoothGroupLine
C++	HRESULT EditView-> SmoothGroupLine ();

**ReduceALine 方法**

描述	抽稀线
Visual Basic	EditView. ReduceALine
C++	HRESULT EditView-> ReduceALine ();

**ChangeLineDir 方法**

描述	改变线方向
Visual Basic	EditView. ChangeLineDir
C++	HRESULT EditView-> ChangeLineDir ();

**EnlargeLine 方法**

描述	放大线
Visual Basic	EditView. EnlargeLine
C++	HRESULT EditView-> EnlargeLine ();

**RotateLine 方法**

描述	旋转线
Visual Basic	EditView. RotateLine
C++	HRESULT EditView-> RotateLine ();

**XMirrorLine 方法**

描述	X 轴镜像
Visual Basic	EditView. XMirrorLine

---

C++          HRESULT EditView-> XMirrorLine ();

**YMirrorLine 方法**

描述          Y 轴镜像

Visual Basic   EditView. YMirrorLine

C++          HRESULT EditView-> YMirrorLine ();

**OMirrorLine 方法**

描述          原点镜像

Visual Basic   EditView. OMirrorLine

C++          HRESULT EditView-> OMirrorLine ();

**UpdateLineParm 方法**

描述          修改线参数

Visual Basic   EditView. UpdateLineParm

C++          HRESULT EditView-> UpdateLineParm ();

**UpdateLineAtt 方法**

描述          修改线属性

Visual Basic   EditView. UpdateLineAtt

C++          HRESULT EditView-> UpdateLineAtt ();

**TakeLineFromArc 方法**

描述          弧工作区提取线

Visual Basic   EditView. TakeLineFromArc

C++          HRESULT EditView-> TakeLineFromArc ();

**ModLinAttByParm 方法**

描述          根据参数赋属性

Visual Basic   EditView. ModLinAttByParm

C++          HRESULT EditView-> ModLinAttByParm ();

**ModLinParmByAtt 方法**

描述          根据属性赋参数

Visual Basic   EditView. ModLinParmByAtt

C++          HRESULT EditView-> ModLinParmByAtt ();

**MakeLineUsePoint 方法**

描述	将已经存在的点连接成线
Visual Basic	EditView. MakeLineUsePoint
C++	HRESULT EditView->MakeLineUsePoint();

**ClipLineHavePoint 方法**

描述	有剪断点
Visual Basic	EditView. ClipLineHavePoint
C++	HRESULT EditView -> ClipLineHavePoint();

**ClipLineNoPoint 方法**

描述	无剪断点。
Visual Basic	EditView. ClipLineNoPoint
C++	HRESULT EditView -> ClipLineNoPoint();

**SelectLine 方法**

描述	选择线
Visual Basic	EditView. SelectLine
C++	HRESULT EditView -> SelectLine ();

## 四、 区编辑操作

**EditGiveRegion 方法**

描述	清除基线
Visual Basic	EditView. EditGiveRegion
C++	HRESULT EditView-> EditGiveRegion ();

**DeleteRegion 方法**

描述	删除区
Visual Basic	EditView. DeleteRegion
C++	HRESULT EditView-> DeleteRegion ();

**TakeArcFromLine 方法**

描述	线工作区提取弧段
Visual Basic	EditView. TakeArcFromLine
C++	HRESULT EditView-> TakeArcFromLine ();

**MakeArcByKey 方法**

描述	键盘输入弧段
Visual Basic	EditView. MakeArcByKey
C++	HRESULT EditView-> MakeArcByKey ();

**MakeArcByKey 方法**

描述	键盘输入弧段
Visual Basic	EditView. MakeArcByKey
C++	HRESULT EditView-> MakeArcByKey ();

**MakeRegion 方法**

描述	输入区
Visual Basic	EditView. MakeRegion
C++	HRESULT EditView-> MakeRegion ();

**MakeSubRegion 方法**

描述	挑子区
Visual Basic	EditView. MakeSubRegion
C++	HRESULT EditView-> MakeSubRegion ();

**UnionRegion 方法**

描述	合并区
Visual Basic	EditView. UnionRegion
C++	HRESULT EditView-> UnionRegion ();

**SplitRegion 方法**

描述	分割 SplitRegion 区
Visual Basic	EditView. SplitRegion
C++	HRESULT EditView-> SplitRegion ();

**NodeCleanLineOrg 方法**

描述	取圆心值
Visual Basic	EditView. NodeCleanLineOrg
C++	HRESULT EditView-> NodeCleanLineOrg ();

**NodeCleanLineAvg 方法**

描述	取平均值
Visual Basic	EditView. NodeCleanLineAvg

---

C++	HRESULT EditView-> NodeCleanLineAvg ();
-----	---

**MakeArc 方法**

描述	输入弧段
Visual Basic	EditView. MakeArc
C++	HRESULT EditView-> MakeArc ();

**AddArcDot 方法**

描述	弧段上加点
Visual Basic	EditView. AddArcDot
C++	HRESULT EditView-> AddArcDot ();

**CheckSelfCross 方法**

描述	检查一个区
Visual Basic	EditView. CheckSelfCross
C++	HRESULT EditView-> CheckSelfCross ();

**CheckAllSelfCross 方法**

描述	检查所有区
Visual Basic	EditView. CheckAllSelfCross
C++	HRESULT EditView-> CheckAllSelfCross ();

**DeleteArcDot 方法**

描述	弧段上删点
Visual Basic	EditView. DeleteArcDot
C++	HRESULT EditView-> DeleteArcDot ();

**MoveArcDot 方法**

描述	鼠标弧段上移点
Visual Basic	EditView. MoveArcDot
C++	HRESULT EditView-> MoveArcDot ();

**MoveArcDotContinue 方法**

描述	鼠标连续移点
Visual Basic	EditView. MoveArcDotContinue
C++	HRESULT EditView-> MoveArcDotContinue ();

**KeyMoveArcDot 方法**

描述	键盘弧段上移点
Visual Basic	EditView. KeyMoveArcDot
C++	HRESULT EditView-> KeyMoveArcDot ();

**DeleteArc 方法**

描述	删除弧段
Visual Basic	EditView. DeleteArc
C++	HRESULT EditView-> DeleteArc ();

**MoveArc 方法**

描述	移动弧段
Visual Basic	EditView. MoveArc
C++	HRESULT EditView-> MoveArc ();

**MoveGroupArcByKey 方法**

描述	移动弧段坐标调整
Visual Basic	EditView. MoveGroupArcByKey
C++	HRESULT EditView-> MoveGroupArcByKey ();

**MoveArcByDist 方法**

描述	推移弧段
Visual Basic	EditView. MoveArcByDist
C++	HRESULT EditView-> MoveArcByDist ();

**ClipArc 方法**

描述	剪断弧段
Visual Basic	EditView. ClipArc
C++	HRESULT EditView-> ClipArc ();

**ChangeArcDir 方法**

描述	弧段改向
Visual Basic	EditView. ChangeArcDir
C++	HRESULT EditView-> ChangeArcDir ();

**LongArc 方法**

描述	延长弧段
Visual Basic	EditView. LongArc

---

C++	HRESULT EditView-> LongArc ();
-----	--------------------------------

**ShortArc 方法**

描述	缩短弧段
Visual Basic	EditView. ShortArc
C++	HRESULT EditView-> ShortArc ();

**NearToArc 方法**

描述	靠近弧段
Visual Basic	EditView. NearToArc
C++	HRESULT EditView-> NearToArc ();

**SmoothArc 方法**

描述	整段光滑
Visual Basic	EditView. SmoothArc
C++	HRESULT EditView-> SmoothArc ();

**SmoothArcSegment 方法**

描述	分段光滑
Visual Basic	EditView. SmoothArcSegment
C++	HRESULT EditView-> SmoothArcSegment ();

**ReduceArc 方法**

描述	弧段抽稀
Visual Basic	EditView. ReduceArc
C++	HRESULT EditView-> ReduceArc ();

**NodeCleanArcOrg 方法**

描述	取圆心值
Visual Basic	EditView. NodeCleanArcOrg
C++	HRESULT EditView-> NodeCleanArcOrg ();

**NodeCleanArcAvg 方法**

描述	取平均值
Visual Basic	EditView. NodeCleanArcAvg
C++	HRESULT EditView-> NodeCleanArcAvg ();

**EnlargeArc 方法**

描述	放大弧段
Visual Basic	EditView. EnlargeArc
C++	HRESULT EditView-> EnlargeArc ();

**RotateArc 方法**

描述	旋转弧段
Visual Basic	EditView. RotateArc
C++	HRESULT EditView-> RotateArc ();

**XMirrorRegion 方法**

描述	X 轴镜像区
Visual Basic	EditView. XMirrorRegion
C++	HRESULT EditView-> XMirrorRegion ();

**OMirrorRegion 方法**

描述	原点镜像区
Visual Basic	EditView. OMirrorRegion
C++	HRESULT EditView-> OMirrorRegion ();

**CopyRegion 方法**

描述	复制区
Visual Basic	EditView. CopyRegion
C++	HRESULT EditView-> CopyRegion ();

**ArrayCopyRegion 方法**

描述	阵列复制区
Visual Basic	EditView. ArrayCopyRegion
C++	HRESULT EditView-> ArrayCopyRegion ();

**CheckSelfCross 方法**

描述	检查一个区
Visual Basic	EditView. CheckSelfCross
C++	HRESULT EditView-> CheckSelfCross ();

**CheckAllSelfCross 方法**

描述	检查所有区
Visual Basic	EditView. CheckAllSelfCross



---

C++ HRESULT EditView-> CheckAllSelfCross ();

**UpdateRegionParm 方法**

描述 修改区参数

Visual Basic EditView. UpdateRegionParm

C++ HRESULT EditView-> UpdateRegionParm ();

**UpdateArcParm 方法**

描述 修改弧段参数

Visual Basic EditView. UpdateArcParm

C++ HRESULT EditView-> UpdateArcParm ();

**UpdateRegionAtt 方法**

描述 修改区属性

Visual Basic EditView. UpdateRegionAtt

C++ HRESULT EditView-> UpdateRegionAtt ();

**UpdateArcAtt 方法**

描述 修改弧段属性

Visual Basic EditView. UpdateArcAtt

C++ HRESULT EditView-> UpdateArcAtt ();

**SetBaseArc 方法**

描述 设置基线

Visual Basic EditView. SetBaseArc

C++ HRESULT EditView-> SetBaseArc ();

**ClearBaseArc 方法**

描述 清除基线

Visual Basic EditView. ClearBaseArc

C++ HRESULT EditView-> ClearBaseArc ();

**DefaultRegParm 方法**

描述 改区缺省参数

Visual Basic EditView. DefaultRegParm

C++ HRESULT EditView-> DefaultRegParm ();

### DefaultRegArcParm 方法

描述	改弧段缺省参数
Visual Basic	EditView. DefaultRegArcParm
C++	HRESULT EditView-> DefaultRegArcParm ();

### ReplaceRegParm 方法

描述	统改区参数
Visual Basic	EditView. ReplaceRegParm
C++	HRESULT EditView-> ReplaceRegParm ();

### ReplaceRegArcParm 方法

描述	统改弧段参数
Visual Basic	EditView. ReplaceRegArcParm
C++	HRESULT EditView-> ReplaceRegArcParm ();

### EditArcAttStru 方法

描述	编辑弧段属性结构
Visual Basic	EditView. EditArcAttStru
C++	HRESULT EditView-> EditArcAttStru ();

### EditRegAttStru 方法

描述	编辑区属性结构
Visual Basic	EditView. EditRegAttStru
C++	HRESULT EditView-> EditRegAttStru ();

### ShowRegionData 方法

描述	查组成区的弧段
Visual Basic	EditView. ShowRegionData
C++	HRESULT EditView-> ShowRegionData ();

### MakeArcOR 方法

描述	圆心造弧
Visual Basic	EditView. MakeArcOR
C++	HRESULT EditView-> MakeArcOR ();

### MakeArc3P 方法

描述	三点造弧
Visual Basic	EditView. MakeArc3P

---

C++            HRESULT EditView-> MakeArc3P ();

**ModRegAttByParm 方法**

描述            根据属性赋参数

Visual Basic    EditView. ModRegAttByParm

C++            HRESULT EditView-> ModRegAttByParm ();

**ModRegParmByAtt 方法**

描述            根据参数赋属性

Visual Basic    EditView. ModRegParmByAtt

C++            HRESULT EditView-> ModRegParmByAtt ();

**SelectRegion 方法**

描述            选择区

Visual Basic    EditView. SelectRegion

C++            HRESULT EditView-> SelectRegion ();

**MakeGraphRegion 方法**

描述            图形造区

Visual Basic    EditView. MakeGraphRegion

C++            HRESULT EditView-> MakeGraphRegion ();

**ShowNodeData 方法**

描述            查组成结点信息

Visual Basic    EditView. ShowNodeData

C++            HRESULT EditView-> ShowNodeData ();

**五、 数字化操作****dgSetParam 方法**

描述            设置数字化参数

Visual Basic    EditView. dgSetParam

C++            HRESULT EditView-> dgSetParam ();

**dgCommonLineByDot 方法**

描述            点方式数字化

Visual Basic    EditView. dgCommonLineByDot

C++            HRESULT EditView-> dgCommonLineByDot ();

**dgCommonLineByRun 方法**

描述	线方式造圆数字化
Visual Basic	EditView. dgCommonLineByRun
C++	HRESULT EditView-> dgCommonLineByRun ();

**dgCrossLine 方法**

描述	正交线数字化
Visual Basic	EditView. dgCrossLine
C++	HRESULT EditView-> dgCrossLine ();

**dgCirByCentreRad 方法**

描述	圆形半径式造圆数字化
Visual Basic	EditView. dgCirByCentreRad
C++	HRESULT EditView-> dgCirByCentreRad ();

**dgCirByTriDot 方法**

描述	三点方式造圆数字化
Visual Basic	EditView. dgCirByTriDot
C++	HRESULT EditView-> dgCirByTriDot ();

**dgArcByCentreRad 方法**

描述	圆形半径式造弧数字化
Visual Basic	EditView. dgArcByCentreRad
C++	HRESULT EditView-> dgArcByCentreRad ();

**dgArcByTriDot 方法**

描述	三点方式造弧数字化
Visual Basic	EditView. dgArcByTriDot
C++	HRESULT EditView-> dgArcByTriDot ();

**dgRectangle 方法**

描述	矩形线数字化
Visual Basic	EditView. dgRectangle
C++	HRESULT EditView-> dgRectangle ();

**dgEllipse 方法**

描述	椭圆线数字化
----	--------

---

Visual Basic	EditView. dgEllipse
C++	HRESULT EditView-> dgEllipse ();

**dgParallelogram 方法**

描述	平行四边形数字化
Visual Basic	EditView. dgParallelogram
C++	HRESULT EditView-> dgParallelogram ();

**dgPntNoteByRun 方法**

描述	字符串数字化
Visual Basic	EditView. dgPntNoteByRun
C++	HRESULT EditView-> dgPntNoteByRun ();

**dgPntSubByRun 方法**

描述	子图数字化
Visual Basic	EditView. dgPntSubByRun
C++	HRESULT EditView-> dgPntSubByRun ();

**dgPntCirByRun 方法**

描述	点圆数字化
Visual Basic	EditView. dgPntCirByRun
C++	HRESULT EditView-> dgPntCirByRun ();

**dgPntArcByRun 方法**

描述	点弧数字化
Visual Basic	EditView. dgPntArcByRun
C++	HRESULT EditView-> dgPntArcByRun ();

**dgProcTabletInitFunc 方法**

描述	初始化数字化仪
Visual Basic	EditView. dgProcTabletInitFunc
C++	HRESULT EditView-> dgProcTabletInitFunc ();

**dgInsDigDevToComPort 方法**

描述	安装数字化仪设备
Visual Basic	EditView. dgInsDigDevToComPort
C++	HRESULT EditView-> dgInsDigDevToComPort ();

## 六、 矢量化操作

### LoadRastFile 方法

描述	装入光栅文件
Visual Basic	EditView. LoadRastFile
C++	HRESULT EditView-> LoadRastFile ();

### SaveRastFile 方法

描述	保存光栅文件
Visual Basic	EditView. SaveRastFile
C++	HRESULT EditView-> SaveRastFile ();

### CloseRastFile 方法

描述	清除光栅文件
Visual Basic	EditView. CloseRastFile
C++	HRESULT EditView-> CloseRastFile ();

### InvertBmp 方法

描述	光栅文件求反
Visual Basic	EditView. InvertBmp
C++	HRESULT EditView-> InvertBmp ();

### AutoRvs 方法

描述	自动矢量化
Visual Basic	EditView. AutoRvs
C++	HRESULT EditView-> AutoRvs ();

### SetRvsAll 方法

描述	全图范围
Visual Basic	EditView. SetRvsAll
C++	HRESULT EditView-> SetRvsAll ();

### SetRvsParm 方法

描述	设置矢量化参数
Visual Basic	EditView. SetRvsParm
C++	HRESULT EditView-> SetRvsParm ();

**SetHeiParm 方法**

描述	设置高程参数
Visual Basic	EditView. SetHeiParm
C++	HRESULT EditView-> SetHeiParm ();

**ManuRvs 方法**

描述	交互式矢量化
Visual Basic	EditView. ManuRvs
C++	HRESULT EditView-> ManuRvs ();

**LoopRvsCenter 方法**

描述	取中心线
Visual Basic	EditView. LoopRvsCenter
C++	HRESULT EditView-> LoopRvsCenter ();

**LoopRvsEdge 方法**

描述	取外边界
Visual Basic	EditView. LoopRvsEdge
C++	HRESULT EditView-> LoopRvsEdge ();

**HeightAlign 方法**

描述	高程自动赋值
Visual Basic	EditView. HeightAlign
C++	HRESULT EditView-> HeightAlign ();

**SetRvsRect 方法**

描述	窗口范围
Visual Basic	EditView. SetRvsRect
C++	HRESULT EditView-> SetRvsRect ();

**UpdateImagePosScale 方法**

描述	设置图像原点参数
Visual Basic	EditView. UpdateImagePosScale
C++	HRESULT EditView-> UpdateImagePosScale ();

**七、 其它操作**

**Topo 方法**

描述	拓扑重建
Visual Basic	EditView. Topo
C++	HRESULT EditView-> Topo ();

**TopoCheck 方法**

描述	拓扑错误检查
Visual Basic	EditView. TopoCheck
C++	HRESULT EditView-> TopoCheck ();

**AutoCrossClipLin 方法**

描述	自动剪断线
Visual Basic	EditView. AutoCrossClipLin
C++	HRESULT EditView-> AutoCrossClipLin ();

**SearchSub 方法**

描述	子区搜索
Visual Basic	EditView. SearchSub
C++	HRESULT EditView-> SearchSub ();

**CoordChange 方法**

描述	键盘输入参数
Visual Basic	EditView. CoordChange
C++	HRESULT EditView-> CoordChange ();

**CoordChangeByMouse 方法**

描述	鼠标定义参数
Visual Basic	EditView. CoordChangeByMouse
C++	HRESULT EditView-> CoordChangeByMouse ();

**LineToArc 方法**

描述	线段转弧
Visual Basic	EditView. LineToArc
C++	HRESULT EditView-> LineToArc ();

**ArcToLine 方法**

描述	弧转线段
Visual Basic	EditView. ArcToLine



---

C++            HRESULT EditView-> ArcToLine ();

**AutoLineClean 方法**

描述            自动线节点平差

Visual Basic    EditView. AutoLineClean

C++            HRESULT EditView-> AutoLineClean ();

**AutoArcClean 方法**

描述            自动弧段节点平差

Visual Basic    EditView. AutoArcClean

C++            HRESULT EditView-> AutoArcClean ();

**MoveGraph 方法**

描述            整块移动

Visual Basic    EditView. MoveGraph

C++            HRESULT EditView-> MoveGraph ();

**SameArcCheck 方法**

描述            重叠弧段检查

Visual Basic    EditView. SameArcCheck

C++            HRESULT EditView-> SameArcCheck ();

**SameLineCheck 方法**

描述            重叠线检查

Visual Basic    EditView. SameLineCheck

C++            HRESULT EditView-> SameLineCheck ();

**MoveGraphByKey 方法**

描述            整块移动坐标调整

Visual Basic    EditView. MoveGraphByKey

C++            HRESULT EditView-> MoveGraphByKey ();

**CopyGraph 方法**

描述            整块复制

Visual Basic    EditView. CopyGraph

C++            HRESULT EditView-> CopyGraph ();

**NearGroupToLine 方法**

描述	线头靠边沿处理
Visual Basic	EditView. NearGroupToLine
C++	HRESULT EditView-> NearGroupToLine ();

**NearGroupToArc 方法**

描述	弧边沿靠近处理
Visual Basic	EditView. NearGroupToArc
C++	HRESULT EditView-> NearGroupToArc ();

**NearGroupToArc 方法**

描述	弧边沿靠近处理
Visual Basic	EditView. NearGroupToArc
C++	HRESULT EditView-> NearGroupToArc ();

**EditSymbolBox 方法**

描述	修改符号编辑框
Visual Basic	EditView. EditSymbolBox
C++	HRESULT EditView-> EditSymbolBox ();

**CombinLabelFile 方法**

描述	Label 与区合并
Visual Basic	EditView. CombinLabelFile
C++	HRESULT EditView-> CombinLabelFile ();

**LineCleanNearDot 方法**

描述	清除线重叠坐标
Visual Basic	EditView. LineCleanNearDot
C++	HRESULT EditView-> LineCleanNearDot ();

**ArcCleanNearDot 方法**

描述	清除弧段重叠坐标
Visual Basic	EditView. ArcCleanNearDot
C++	HRESULT EditView-> ArcCleanNearDot ();

**CleanNouseLine 方法**

描述	清除微短线
Visual Basic	EditView. CleanNouseLine

C++ HRESULT EditView-> CleanNouseLine ();

#### CleanNouseArc 方法

描述 清除微短弧线

Visual Basic EditView. CleanNouseArc

C++ HRESULT EditView-> CleanNouseArc ();

#### LineTopoCheck 方法

描述 线的拓扑错误检查

Visual Basic EditView. LineTopoCheck

C++ HRESULT EditView-> LineTopoCheck ();

#### LineTopoCheck 方法

描述 线的拓扑错误检查

Visual Basic EditView. LineTopoCheck

C++ HRESULT EditView-> LineTopoCheck ();

### 9.1.3、事件：

#### EditOperator 事件

描述 控件编辑操作发生改变

Visual Basic EditView\_EditOperator (long opcode)

参数说明：

参数	类型	出口/入口	描述
opcode	Long	[in]	1-编辑操作状态/0-非编辑操作状态

#### MousePosition 事件

描述 鼠标位置发生改变

Visual Basic EditView\_MousePosition (x\_Pos As double, y\_Pos As double)

参数说明：

参数	类型	出口/入口	描述
x_Pos	Double	[in]	鼠标位置 x
y_Pos	Double	[in]	鼠标位置 y

#### MouseButtonDown 事件

描述 鼠标左键点下

Visual Basic EditView\_MouseLButtonDown (x\_Pos As double, y\_Pos As double)

参数说明：

参数	类型	出口/入口	描述
x_Pos	Double	[in]	鼠标位置 x
y_Pos	Double	[in]	鼠标位置 y

**MouseButtonUp 事件**

描述 鼠标左键放开

Visual Basic EditView\_MouseLButtonUp (x\_Pos As double, y\_Pos As double)

参数说明：

参数	类型	出口/入口	描述
x_Pos	Double	[in]	鼠标位置 x
y_Pos	Double	[in]	鼠标位置 y

**MouseButtonDown 事件**

描述 鼠标右键点下

Visual Basic EditView\_MouseRButtonDown (x\_Pos As double, y\_Pos As double)

参数说明：

参数	类型	出口/入口	描述
x_Pos	Double	[in]	鼠标位置 x
y_Pos	Double	[in]	鼠标位置 y

**LButtonDblClk 事件**

描述 鼠标左键双击

Visual Basic EditView\_LButtonDblClk (x\_Pos As double, y\_Pos As double)

参数说明：

参数	类型	出口/入口	描述
x_Pos	Double	[in]	鼠标位置 x
y_Pos	Double	[in]	鼠标位置 y

**MyDraw 事件**

描述 显示图形内容

Visual Basic EditView\_MyDraw (MpDC As MapGisDC)

参数说明：

参数	类型	出口/入口	描述
MpDC	MapGisDC	[in]	显示设备 MapGisDC 对象

**CurAttElement 事件**

描述 属性联动时选中了某个图元

---

Visual Basic EditView\_CurAttElement (lNo As Long)

参数说明：

参数	类型	出口/入口	描述
lNo	Long	[in]	图元序号

---

## 10.2、MapTreeCtrl 控件

描述 MapTreeCtrl 控件对象是封装了 Map 交互操作功能的控件。

### 10.2.1、属性：无

### 10.2.2、方法：

#### SetMap 方法

描述 设置 Map 对象

Visual Basic MapTreeCtrl.SetMap(newMap)

C++ pMapTreeCtrl->SetMap(newMap)

参数说明：

参数	类型	出口/入口	描述
newMap	Map	[in]	Map 对象

---

#### GetMap 方法

描述 返回 Map 对象

Visual Basic Set curMap = MapTreeCtrl.GetMap()

C++ pMapTreeCtrl->GetMap(&curMap)

#### ReloadTree 方法

描述 重新生成控件内容

Visual Basic MapTreeCtrl.ReloadTree()

C++ pMapTreeCtrl->ReloadTree()

### 10.2.3、事件：

#### UpdateMapTree 事件

描述 Map 对象内容发生改变

Visual Basic MapTreeCtrl\_UpdateMapTree()

### MouseDownClick 事件

描述 鼠标双击事件

Visual Basic MapTreeCtrl\_MouseDbClick(  
clicklayer As MAPGISLAYERCOMLib.MapLayer)

参数说明：

参数	类型	出口/入口	描述
clicklayer	MapLayer	[in]	选中的 MapLayer 对象

### MouseDownClick 事件

描述 鼠标左键单击事件

Visual Basic MapTreeCtrl\_MouseLClick(  
clicklayer As MAPGISLAYERCOMLib.MapLayer)

参数说明：

参数	类型	出口/入口	描述
clicklayer	MapLayer	[in]	选中的 MapLayer 对象

## 10.3、GisAttEdit 控件

描述 GisAttEdit 控件主要用来浏览和编辑 MAPGIS 属性数据。

### 10.3.1、属性：

属性	描述
area	正在编辑的工作区对象
WorkArea	
只读	
rcds	正在编辑的记录集对象
RecordSet	
只读	
EditEnable	可编辑状态（0/1—浏览/编辑）
Integer	
可读写	
GisAttBackColor	背景色
Long	
可读写	
GisAttForeColor	前景色
Long	

可读写	
PopupMenu	弹出菜单标记
Integer	0-不弹出菜单      1-只弹出右键菜单
可读写	2-只弹出左键菜单      3-弹出左、右键菜单
ai	正在编辑的工作区号
Integer	
只读	
type	正在编辑的属性类型
Integer	
只读	

### 10.3.2、方法：

#### AttachArea 方法

描述 设置工作区对象到控件中。此方法将释放控件中原来的工作区对象或记录集对象。

Visual Basic rtn = GisAttEdit.AttachArea(area, type)

C++ short pGisAttEdit->AttachArea(IWorkArea \*area, short type);

参数说明：

参数	类型	出口/入口	描述
area	WorkArea	[in]	浏览、编辑属性的工作区对象
type	Integer	[in]	属性类型
rtn	Integer	[out]	成功返回 1，失败返回 0

#### AttachRcds 方法

描述 设置记录集对象到控件中。此方法将释放控件中原来的工作区对象或记录集对象。

Visual Basic rtn = GisAttEdit.AttachRcds(rcds)

C++ short pGisAttEdit->AttachArea(IRecordSet \*rcds);

参数说明：

参数	类型	出口/入口	描述
rcds	RecordSet	[in]	浏览、编辑属性的记录集对象
rtn	Integer	[out]	成功返回 1，失败返回 0

#### AttachAi 方法

描述 设置工作区到控件中。此方法将释放控件中原来的工作区对象或记录集对象。

Visual Basic rtn = GisAttEdit.AttachAi(ai, type)

C++ short pGisAttEdit->AttachAi(short ai, short type);

参数说明：

参数	类型	出口/入口	描述
ai	Integer	[in]	浏览、编辑属性的工作区号
type	Integer	[in]	属性类型
rtn	Integer	[out]	成功返回 1，失败返回 0

**BrowLstRcds 方法**

描述 浏览、编辑部分属性记录。此方法仅当控件中对应的对象为工作区对象时有效。

Visual Basic rtn = GisAttEdit.BrowLstRcds (lst, type)

C++ short pGisAttEdit->BrowLstRcds (IIDList \*lst, short type);

参数说明：

参数	类型	出口/入口	描述
lst	IDList	[in]	浏览、编辑属性的记录号列表对象
type	Integer	[in]	属性类型
rtn	Integer	[out]	成功返回 1，失败返回 0

**BrowAiLstRcds 方法**

描述 浏览、编辑部分属性记录。此方法与 AttachAi 方法配套使用。

Visual Basic rtn = GisAttEdit. BrowAiLstRcds (lst, type)

C++ short pGisAttEdit->BrowAiLstRcds(long \*lst, short type);

参数说明：

参数	类型	出口/入口	描述
lst	Long	[in]	浏览、编辑属性的记录号列表；lst[0]保存了列表中的记录号总数。
type	Integer	[in]	属性类型
rtn	Integer	[out]	成功返回 1，失败返回 0

**DetachArea 方法**

描述 释放控件中的工作区对象。

Visual Basic GisAttEdit.DetachArea()

C++ void pGisAttEdit->DetachArea();

**DetachRcds 方法**

描述 释放控件中的记录集对象。

Visual Basic GisAttEdit.DetachRcds()

C++ void pGisAttEdit->DetachRcds();



**DetachAi 方法**

描述	释放控件中的工作区。
Visual Basic	GisAttEdit.DetachAi()
C++	void pGisAttEdit->DetachAi();

**RefreshAtt 方法**

描述	刷新控件中的属性内容。当控件对应的内容改变时（直接对控件对应的工作区对象或记录集对象进行操作如：添加、删除记录、改变属性结构等），应调用此方法将属性内容刷新。
Visual Basic	GisAttEdit.RefreshAtt()
C++	void pGisAttEdit->RefreshAtt();

**UpdateAtt 方法**

描述	更新属性内容。将控件对属性的修改保存到工作区或记录集中。
Visual Basic	GisAttEdit.UpdateAtt()
C++	void pGisAttEdit->UpdateAtt();

**GotoAtt 方法**

描述	将控件的当前编辑项，跳转到指定项。
Visual Basic	rtn = GisAttEdit.GotoAtt(rcdNo)
C++	short pGisAttEdit->GotoAtt(long rcdNo);

参数说明：

参数	类型	出口/入口	描述
rcdNo	Long	[in]	跳转的目的记录号；rcdNo=0 时，弹出跳转条件对话框。
rtn	Integer	[out]	成功返回 1，失败返回 0

**HideFields 方法**

描述	屏蔽字段。
Visual Basic	rtn = GisAttEdit.HideFields()
C++	short pGisAttEdit->HideFields();

参数说明：

参数	类型	出口/入口	描述
rtn	Integer	[out]	成功返回 1，失败返回 0

**SortRcds 方法**

描述	记录排序。
----	-------

Visual Basic    rtn = GisAttEdit.SortRcds ()  
 C++            void pGisAttEdit->SortRcds ();

**SortRcdsOnFldi 方法**

描述            按指定的字段排序。

Visual Basic    GisAttEdit.SortRcdsOnFldi (fldi, dir)  
 C++            void pGisAttEdit->SortRcdsOnFldi (VARIANT fldi, short dir);

参数说明：

参数	类型	出口/入口	描述
fldi	Variant	[in]	排序的字段号或字段名
dir	Integer	[in]	0/1 分别对应升序和降序

**SetExtDb 方法**

描述            设置外挂数据库。

Visual Basic    rtn = GisAttEdit.SetExtDb()  
 C++            short pGisAttEdit->SetExtDb();

参数说明：

参数	类型	出口/入口	描述
rtn	Integer	[out]	成功返回 1，失败返回 0

**SetFldiAsDbKey 方法**

描述            设置或取消字段 fldi 为外挂数据库字段。

Visual Basic    rtn = GisAttEdit.SetFldiAsDbKey(fldi, flg)  
 C++            short pGisAttEdit->SetFldiAsDbKey(VARIANT fldi, short flg);

参数说明：

参数	类型	出口/入口	描述
fldi	Variant	[in]	字段号或字段名
flg	Integer	[in]	0/1 分别对应取消和设置
rtn	Integer	[out]	成功返回 1，失败返回 0

**SetFldiExtDb 方法**

描述            修改字段 fldi 的外挂数据库。

Visual Basic    rtn = GisAttEdit.SetFldiExtDb(fldi)  
 C++            short pGisAttEdit->SetFldiExtDb(VARIANT fldi);

参数说明：

参数	类型	出口/入口	描述
fldi	Variant	[in]	字段号或字段名

rtn	Integer	[out]	成功返回 1，失败返回 0
-----	---------	-------	---------------

**RepRcdsFldVal 方法**

描述 统一改字段值。

Visual Basic rtn = GisAttEdit. RepRcdsFldVal()

C++ short pGisAttEdit-> RepRcdsFldVal();

参数说明：

参数	类型	出口/入口	描述
rtn	Integer	[out]	成功返回 1，失败返回 0

**DelCurRcd 方法**

描述 删除当前记录。此方法仅对表工作区或记录集有效。

Visual Basic GisAttEdit.DelCurRcd ()

C++ void pGisAttEdit->DelCurRcd ();

**AppendRcds 方法**

描述 添加记录。此方法仅对表工作区或记录集有效。

Visual Basic GisAttEdit. AppendRcds ()

C++ void pGisAttEdit-> AppendRcds ();

**SetAutoAppendRcdFlg 方法**

描述 设置自动添加记录标志。此方法仅对表工作区或记录集有效。

Visual Basic GisAttEdit. SetAutoAppendRcdFlg(flag)

C++ void pGisAttEdit-> SetAutoAppendRcdFlg(short flg);

参数说明：

参数	类型	出口/入口	描述
flg	Integer	[in]	自动添加记录标志

**10.3.3、事件：****CurRcdFldNo 方法**

描述 当浏览、编辑的记录号或字段号改变时触发此事件。

Visual Basic GisAttEdit.CurRcdFldNo(rcdNo, fldNo)

C++ pGisAttEdit-> CurRcdFldNo(long rcdNo, short fldNo);

参数说明：

参数	类型	出口/入口	描述
rcdNo	Long	[in]	浏览、编辑的当前记录号

fIdNo	Integer	[in]	浏览、编辑的当前字段号
-------	---------	------	-------------

## 10.4、AttStruEdit 控件

**描述** AttStruEdit 控件主要用来浏览和编辑 MAPGIS 属性结构。

### 10.4.1、属性：

属性	描述
area	正在编辑的工作区对象
WorkArea	
只读	
stru	正在编辑的属性结构对象
RecordSet	
只读	
aiStru	正在编辑的属性结构
Variant	说明：必须将得到的 VARIANT 类型属性值的 pIVal 强制转换成 CATT_STRU 类型,以获得属性结构。
只读	
EditEnable	可编辑状态（0/1—浏览/编辑）
Integer	
可读写	
GisAttBackColor	背景色
Long	
可读写	
GisAttForeColor	前景色
Long	
可读写	
DispExtAttStru	显示、隐藏扩展属性结构标志
Integer	
可读写	
PopupMenu	弹出菜单标记
Integer	0-不弹出菜单    1-只弹出右键菜单
可读写	

### 10.4.2、方法：

**AttachArea 方法**

**描述** 设置工作区对象到控件中。此方法将释放控件中原来的工作区对象或属性结构对象。

Visual Basic `rtn = AttStruEdit.AttachArea(area, type)`

C++ `short pAttStruEdit->AttachArea(IWorkArea *area, short type);`

**参数说明：**

参数	类型	出口/入口	描述
area	WorkArea	[in]	浏览、编辑属性结构的工作区对象
type	Integer	[in]	属性类型
rtn	Integer	[out]	成功返回 1，失败返回 0

**AttachStru 方法**

**描述** 设置属性结构对象到控件中。此方法将释放控件中原来的工作区对象或属性结构对象。

Visual Basic `rtn = AttStruEdit.AttachStru(stru)`

C++ `short pAttStruEdit->AttachStru(IRecord_Head *stru);`

**参数说明：**

参数	类型	出口/入口	描述
stru	Record_Head	[in]	浏览、编辑的属性结构对象
rtn	Integer	[out]	成功返回 1，失败返回 0

**AttachAi 方法**

**描述** 设置工作区到控件中。

Visual Basic `rtn = AttStruEdit.AttachAi(ai, type)`

C++ `short pAttStruEdit->AttachAi(short ai, short type);`

**参数说明：**

参数	类型	出口/入口	描述
ai	Integer	[in]	浏览、编辑属性结构的工作区号
type	Integer	[in]	属性类型
rtn	Integer	[out]	成功返回 1，失败返回 0

**AttachAiStru 方法**

**描述** 设置属性结构到控件中。

Visual Basic `rtn = AttStruEdit.AttachAiStru(stru)`

C++ `short pAttStruEdit->AttachAiStru(long *stru);`

**参数说明：**

参数	类型	出口/入口	描述
----	----	-------	----

stru	long *	[in]	浏览、编辑的属性结构
rtn	Integer	[out]	成功返回 1，失败返回 0

**DetachArea 方法**

描述	释放控件中的工作区对象。
Visual Basic	AttStruEdit.DetachArea()
C++	void pAttStruEdit->DetachArea();

**DetachStru 方法**

描述	释放控件中的属性结构对象。
Visual Basic	AttStruEdit.DetachRcds()
C++	void pAttStruEdit->DetachRcds();

**DetachAi 方法**

描述	释放控件中的工作区。
Visual Basic	AttStruEdit.DetachAi()
C++	void pAttStruEdit->DetachAi();

**DetachAiStru 方法**

描述	释放控件中的属性结构。
Visual Basic	AttStruEdit.DetachAiStru()
C++	void pAttStruEdit->DetachAiStru();

**RefreshStru 方法**

描述	刷新控件中的属性结构。当控件对应的属性结构改变时（直接对控件对应的工作区对象或属性结构对象进行操作），应调用此方法进行刷新。
Visual Basic	AttStruEdit.RefreshAtt()
C++	void pAttStruEdit->RefreshAtt();

**UpdateStru 方法**

描述	更新属性结构。将控件对属性结构的修改保存到工作区或属性结构对象中。
Visual Basic	AttStruEdit.UpdateStru()
C++	void pAttStruEdit->UpdateStru();

**DelCurFld 方法**

描述	删除当前字段。
Visual Basic	AttStruEdit.DelCurFld()
C++	void pAttStruEdit->DelCurFld();

**InsertAFld 方法**

描述            插入一个字段。

Visual Basic    AttStruEdit.InsertAFld()

C++             void pAttStruEdit->InsertAFld();

**DispExtFld 方法**

描述            显示扩展字段。

Visual Basic    AttStruEdit.DispExtFld(flag)

C++             void pAttStruEdit->DispExtFld(short flag);

参数说明：

参数	类型	出口/入口	描述
flag	Integer	[in]	显示扩展字段标志

# 附 录 MAPGIS 组件对象索引

## A

Analysis 对象 .....	319
AppModel 对象 .....	592
Arc_Alloc_InfoEdit 控件 .....	364
Arc_Alloc_Info 对象 .....	91
Ask_Db_Rcd 对象 .....	145
AttStruEdit 控件 .....	649

## C

## D

D_3DotSet 对象 .....	43
D_3Dot 对象 .....	43
D_DotSet 对象 .....	37
D_Dot 对象 .....	37
D_RectSet 对象 .....	50
D_Rect 对象 .....	49
D3SecARRAY 对象 .....	534
Date_Stru 对象 .....	52
DBS_FramFileSet 对象 .....	492
DBS_FramFile 对象 .....	491
DBS_FramInfo 对象 .....	489
DBS_GraIndexSet 对象 .....	495
DBS_GraIndex 对象 .....	495
DBSArea 对象 .....	498
DBSLayer 对象 .....	528
DemInfoStru 对象 .....	534
DemInsObj 对象 .....	535
DistPowGridStru 对象 .....	536

## E

EditView 控件 .....	605
ExpObj 对象 .....	301
Ext_Link_Db_Info 对象 .....	146

ExtDbTool 对象 .....	148
--------------------	-----

## F

Field_ExtHead 对象 .....	54
Field_Head 对象 .....	58
Field 对象 .....	65

## G

GeoLin3DSet 对象 .....	124
GeoLin3D 对象 .....	123
GeoLinSet 对象 .....	121
GeoLin 对象 .....	121
GeoNetSet 对象 .....	130
GeoNet 对象 .....	129
GeoPathSet 对象 .....	135
GeoPath 对象 .....	132
GeoPntSet 对象 .....	119
GeoPnt 对象 .....	118
GeoRegSet 对象 .....	127
GeoReg 对象 .....	125
GisAttEdit 控件 .....	643
GisEnv 对象 .....	70
GridModel 对象 .....	577
GroupLayer 对象 .....	427

## I

IDList 对象 .....	138
-----------------	-----

## K

KringGridStru 对象 .....	536
------------------------	-----

## L

L_3DotSet 对象 .....	47
L_3Dot 对象 .....	46
L_DotSet 对象 .....	40
L_Dot 对象 .....	40



LabelFontStru 对象 .....	537	<b>N</b>	
LabelFormatStru 对象 .....	538	Net_InfoEdit 控件 .....	371
LayerOnOffPad 对象 .....	136	Net_Info 对象 .....	95
LayerPad 对象 .....	137	Net_PadEdit 控件 .....	373
Layer 对象 .....	157	Net_Pad 对象 .....	97
LegendItem 对象 .....	442	NetArea 对象 .....	273
Lin_InfoEdit 控件 .....	365	NetAtt 对象 .....	180
Lin_Info 对象 .....	87	NetLayer 对象 .....	436
Lin_PadEdit 控件 .....	367	NetSet 对象 .....	112
Lin_Pad 对象 .....	89	Net 对象 .....	111
Lin_TopEdit 控件 .....	369	Nod_Barrier_Info 对象 .....	100
Lin_Top 对象 .....	90	Nod_Center_InfoEdit 控件 .....	375
Lin3DSet 对象 .....	106	Nod_Center_Info 对象 .....	99
Lin3D 对象 .....	105	Nod_Stop_Info 对象 .....	100
LinArea 对象 .....	237	NodAtt 对象 .....	176
LinAtt 对象 .....	167	NodData 对象 .....	184
LinData 对象 .....	202	<b>P</b>	
LinLayer 对象 .....	434	PathData 对象 .....	211
LinLegendItem 对象 .....	444	PathSet 对象 .....	117
LinSet 对象 .....	104	Path 对象 .....	114
Lin 对象 .....	103	Pnt_Info_ArcEdit 控件 .....	377
LONGList 对象 .....	140	Pnt_Info_Arc 对象 .....	76
<b>M</b>		Pnt_Info_CirEdit 控件 .....	379
Map_Para 对象 .....	71	Pnt_Info_Cir 对象 .....	76
MapGisDC 对象 .....	403	Pnt_Info_ImageEdit 控件 .....	381
MapGis 对象 .....	330	Pnt_Info_Image 对象 .....	77
MapLayer 对象 .....	426	Pnt_Info_NoteEdit 控件 .....	382
MapLegend 对象 .....	446	Pnt_Info_Note 对象 .....	75
MapNorthArrow 对象 .....	451	Pnt_Info_SubEdit 控件 .....	384
MapPrintParam 对象 .....	450	Pnt_Info_Sub 对象 .....	74
MapScaleBar 对象 .....	451	Pnt_Info_TextEdit 控件 .....	386
MapScaleText 对象 .....	451	Pnt_Info_Text 对象 .....	77
MapSurround 对象 .....	451	Pnt_Info 对象 .....	79
MapTreeCtrl 控件 .....	642	Pnt_Pad_ArcEdit 控件 .....	388
Map 对象 .....	437	Pnt_Pad_Arc 对象 .....	83
MsiToVector 对象 .....	485	Pnt_Pad_CirEdit 控件 .....	390
MSYSTEMTime 对象 .....	490		

Pnt_Pad_Cir 对象.....	82	RepFldFlag 对象.....	142
Pnt_Pad_ImageEdit 控件 .....	391	<b>S</b>	
Pnt_Pad_Image 对象.....	83	SearchDataStru 对象.....	539
Pnt_Pad_NoteEdit 控件 .....	393	SelectItem 对象 .....	451
Pnt_Pad_Note 对象 .....	80	SelectSet 对象.....	452
Pnt_Pad_SubEdit 控件 .....	395	SlopeLineStru 对象 .....	539
Pnt_Pad_Sub 对象.....	81	<b>T</b>	
Pnt_Pad_TextEdit 控件 .....	397	TblArea 对象 .....	290
Pnt_Pad_Text 对象.....	84	Tic_DotSet 对象 .....	144
Pnt_Pad 对象.....	85	Tic_Dot 对象 .....	143
PntArea 对象 .....	220	Time_Stru 对象.....	53
PntAtt 对象.....	163	TimeStamp_Stru 对象 .....	53
PntLayer 对象.....	436	TinArea 对象 .....	546
PntLegendItem 对象.....	443	TinLayer 对象.....	602
PntSet 对象.....	101	TinModel 对象.....	567
Pnt 对象.....	101	TinNetDispStru 对象 .....	540
<b>Q</b>		TinNetSet 对象 .....	543
Query 对象 .....	301	TinNet 对象 .....	542
<b>R</b>		TinPntSet 对象.....	541
RadSet 对象.....	147	TinPnt 对象.....	541
Rad 对象.....	146	TraceDEM 对象.....	326
RasterArea 对象 .....	454	Tran_Par 对象.....	70
RasterLayer 对象.....	483	<b>V</b>	
Record_Head 对象 .....	62	VectorLayer 对象.....	433
RecordSet 对象.....	67	VectorToMsi 对象.....	484
Record 对象.....	65	VectWorkArea 对象.....	218
Reg_InfoEdit 控件 .....	399	<b>W</b>	
Reg_Info 对象 .....	92	Win_Org 对象.....	52
Reg_PadEdit 控件 .....	401	WorkArea 对象 .....	214
Reg_Pad 对象.....	94	<b>Z</b>	
RegArea 对象 .....	253	ZLevelStruSet 对象 .....	545
RegAtt 对象.....	171	ZLevelStru 对象 .....	544
RegLayer 对象 .....	436		
RegLegendItem 对象 .....	445		
RegSet 对象.....	109		
Reg 对象.....	107		

