

# **MAPGIS7-IMS 行业版**

## 二次开发函数库篇

# 目 录

目 录.....	- 2 -
A. 服务器端组件函数库.....	- 22 -
A.1 .NET 函数库.....	- 22 -
A.1.1 .NET 类库.....	- 22 -
1. BusCom 类.....	- 22 -
1.1 构造函数.....	- 22 -
1.2 LoadData 方法.....	- 22 -
1.3 GetPassLineByStopName 方法.....	- 23 -
1.4 GetStopListByLineName 方法.....	- 23 -
1.5 GetChangeRlt 方法.....	- 24 -
2. ChineseParse 类.....	- 24 -
2.1 ChineseParse 方法.....	- 24 -
2.2 ParseChinese 方法.....	- 25 -
3. CLayerInformation 类.....	- 25 -
3.1 LayerAlias 属性.....	- 25 -
3.2 LayerBound 属性.....	- 25 -
3.3 LayerName 属性.....	- 26 -
3.4 LayerStatus 属性.....	- 26 -
4. CLayerObj 类.....	- 26 -
4.1 CLayerObj 方法.....	- 26 -
4.2 getLayerAttStruct 方法.....	- 26 -
4.3 getLayerInfo 方法.....	- 27 -
4.4 getLayerType 方法.....	- 27 -
4.5 setLayerAttStruct 方法.....	- 27 -
4.6 setLayerInfo 方法.....	- 27 -
4.7 setLayerType 方法.....	- 28 -
5. CMapInformation 类.....	- 28 -
5.1 Layers 属性.....	- 28 -
5.2 MapAlias 属性.....	- 29 -
5.3 MapName 属性.....	- 29 -
5.4 MapBound 属性.....	- 29 -
6. CMapView 类.....	- 30 -
属性.....	- 30 -
方法.....	- 35 -
7. CMapViewPort 类.....	- 71 -
7.1 CMapViewPort.....	- 71 -
7.2 CMapViewPort.....	- 71 -
7.3 CMapViewPort.....	- 72 -
7.4 ComputerGlobeIndex 方法.....	- 72 -
7.5 GetMapImage 方法.....	- 72 -
7.6 GetMapImage 方法.....	- 73 -

7.7 GetMapImage 方法.....	- 73 -
7.8 GetMapImage 方法.....	- 74 -
7.9 SaveMap 方法.....	- 74 -
7.10 SaveMap 方法.....	- 75 -
7.11 Dispose 方法.....	- 75 -
7.12 SetIndexOffset 方法.....	- 76 -
8. CMpAtt 类.....	- 76 -
8.1 addLayerAttStru 方法.....	- 76 -
8.2 CMpAtt 方法.....	- 76 -
8.3 GetFieldName 方法.....	- 77 -
8.4 GetFieldNumber 方法.....	- 77 -
8.5 GetLayerRecordCount 方法.....	- 77 -
8.6 GetRecordByID 方法.....	- 78 -
8.7 ITotalFNum 属性.....	- 78 -
8.8 sLayerNum 属性.....	- 78 -
9. CNetEdge 类.....	- 78 -
9.1 CNetEdge 方法.....	- 78 -
9.2 CNetEdge 方法.....	- 79 -
9.3 Dots 属性.....	- 79 -
9.4 FieldValus 属性.....	- 79 -
10. CNetInputDot 类.....	- 79 -
10.1 CNetInputDot 方法.....	- 79 -
10.2 CNetInputDot 方法.....	- 80 -
10.3 pDot 属性.....	- 80 -
10.4 positionPercent 属性.....	- 80 -
11. CNetNode 类.....	- 80 -
11.1 CNetNode 方法.....	- 80 -
11.2 CNetNode 方法.....	- 81 -
11.3 CNetNode 方法.....	- 81 -
11.4 FieldValus 属性.....	- 81 -
11.5 Node 属性.....	- 81 -
11.6 TurnAngle 属性.....	- 82 -
12. CNetPath 类.....	- 82 -
12.1 CNetPath 方法.....	- 82 -
12.2 CNetPath 方法.....	- 82 -
12.3 Edges 属性.....	- 82 -
12.4 Nodes 属性.....	- 83 -
13. CNetPathAnaly 类.....	- 83 -
13.1 CNetPathAnaly 方法.....	- 83 -
13.2 OpenPathAnalySvr 方法.....	- 83 -
13.3 ClosePathAnalySvr 方法.....	- 84 -
13.4 PathAnalyze 方法.....	- 84 -
13.4 SearchDriveWay 方法.....	- 84 -
13.5 SearchDriveWay 方法.....	- 85 -

13.6 SetBarrier 方法.....	- 85 -
13.7 SetDscrInfoParam 方法.....	- 85 -
13.8 SetNetWeight 方法.....	- 86 -
14. CPathAnalyzeResult 类.....	- 86 -
14.1 CPathAnalyzeResult 方法.....	- 86 -
14.2 edgeFieldNameArray 属性.....	- 86 -
14.3 inputDots 属性.....	- 86 -
14.4 nodeFieldNameArray 属性.....	- 87 -
14.5 Paths 属性.....	- 87 -
15. CWebMap 类.....	- 87 -
属性.....	- 87 -
方法.....	- 88 -
16. GISDataQuery 类.....	- 91 -
16.1 GISDataQuery 方法.....	- 91 -
16.2 Open 方法.....	- 92 -
16.3 Close 方法.....	- 92 -
16.4 Dispose 方法.....	- 93 -
16.5 GetDataByHitting 方法.....	- 93 -
16.6 GetDataByCircle 方法.....	- 94 -
16.7 GetDataByRectangle 方法.....	- 94 -
16.8 GetDataByPolygon 方法.....	- 95 -
16.9 GetDataByCondition 方法.....	- 95 -
16.10 IsDotInPolygon 方法.....	- 96 -
17. GISSvrGenerator 类.....	- 97 -
17.1 discon 方法.....	- 97 -
17.2 getGisSvr 方法.....	- 97 -
17.3 GISSvrGenerator 方法.....	- 97 -
17.4 sendGisSvr 方法.....	- 98 -
18. IMSOBJArray 类.....	- 98 -
属性.....	- 98 -
方法.....	- 98 -
19. OpCAddLayerInfo 类.....	- 100 -
19.1 DBName 属性.....	- 100 -
19.2 GDBName 属性.....	- 100 -
19.3 LayerIndex 属性.....	- 100 -
19.4 LayerInformation 属性.....	- 100 -
19.5 MapID 属性.....	- 100 -
19.6 PassWord 属性.....	- 101 -
19.7 UserName 属性.....	- 101 -
20. OpCAppendAArcLine 类.....	- 101 -
20.1 LayerIndex 属性.....	- 101 -
20.2 LineColor 属性.....	- 101 -
20.2 LineRateX 属性.....	- 102 -
20.3 LineRateY 属性.....	- 102 -

20.4 LineType 属性.....	- 102 -
20.5 MapID 属性.....	- 102 -
20.6 PenWidth 属性.....	- 103 -
20.7 PropertyNumber 属性.....	- 103 -
20.8 PropertyValues 属性.....	- 103 -
20.9 TheArc 属性.....	- 103 -
21. OpCAppendAPoint 类.....	- 104 -
21.1 LayerIndex 属性.....	- 104 -
21.2 MapID 属性.....	- 104 -
21.3 PicColor 属性.....	- 104 -
21.4 PicType 属性.....	- 104 -
21.5 PicHeight 属性.....	- 105 -
21.6 PicWidth 属性.....	- 105 -
21.7 PropertyNumber 属性.....	- 105 -
21.8 PropertyValues 属性.....	- 105 -
21.9 ThePoint 属性.....	- 106 -
22. OpCAppendARegion 类.....	- 106 -
22.1 LayerIndex 属性.....	- 106 -
22.2 MapID 属性.....	- 106 -
22.3 RegioniPic 属性.....	- 107 -
22.4 RegionColor 属性.....	- 107 -
22.5 PicColor 属性.....	- 107 -
22.6 PicHeight 属性.....	- 107 -
22.7 PicWidth 属性.....	- 108 -
22.8 PropertyNumber 属性.....	- 108 -
22.9 PropertyValues 属性.....	- 108 -
22.10 TheRegion 属性.....	- 108 -
23. OpcAppendOperResult 类.....	- 109 -
24. OpCAttSelectStru 类.....	- 109 -
24.1 AttributeCondition 属性.....	- 109 -
24.2 LayerIndex 属性.....	- 109 -
24.3 MapID 属性.....	- 109 -
24.4 PageIndex 属性.....	- 110 -
24.5 recordCountOfPage 属性.....	- 110 -
25. OpCBasicNetAnalysis 类.....	- 110 -
25.1 LayerIndex 属性.....	- 110 -
25.2 NodeNumber 属性.....	- 110 -
25.3 Radius 属性.....	- 111 -
25.4 Nodes 属性.....	- 111 -
25.5 MapID 属性.....	- 111 -
25.6 barNum 属性.....	- 111 -
25.7 barPoints 属性.....	- 112 -
26. OpCCircleSelectClass 类.....	- 112 -
26.1 Center 属性.....	- 112 -

26.2 LayerIndex 属性.....	- 112 -
26.3 Radius 属性.....	- 112 -
26.4 MapID 属性.....	- 113 -
26.5 PageIndex 属性.....	- 113 -
26.6 recordCountOfPage 属性.....	- 113 -
27. OpCCircleWithAttSelectStru 类.....	- 113 -
27.1 Question 属性.....	- 114 -
28. OpCDeleteFeature 类.....	- 114 -
28.1 DelGeometry 属性.....	- 114 -
28.2 FeatureID 属性.....	- 114 -
28.3 LayerIndex 属性.....	- 114 -
28.4 MapID 属性.....	- 115 -
29. OpCEagleEye 类.....	- 115 -
29.1 EagleRect 属性.....	- 115 -
29.2 Height 属性.....	- 115 -
29.3 ImageFileName 属性.....	- 115 -
29.4 MapID 属性.....	- 116 -
29.5 Rate 属性.....	- 116 -
29.6 Width 属性.....	- 116 -
30. OpCEagleEyeResult 类.....	- 116 -
30.1 ImageFileName 属性.....	- 116 -
30.2 ImageSize 属性.....	- 117 -
31. OpCFeatureGeometry 类.....	- 117 -
31.1 PolygonNumber 属性.....	- 117 -
31.2 ArcNumber 属性.....	- 117 -
31.3 PointNumber 属性.....	- 117 -
31.4 Polygons 属性.....	- 118 -
31.5 ArcLines 属性.....	- 118 -
31.6 Points 属性.....	- 118 -
32. OpCFeatureStyleInfo 类.....	- 118 -
32.1 Color 属性.....	- 118 -
32.2 FeatureType 属性.....	- 119 -
32.3 MarginHeight 属性.....	- 119 -
32.4 MarginWidth 属性.....	- 119 -
32.5 PatternIndex 属性.....	- 119 -
33. OpCGeomInfo 类.....	- 119 -
33.1 LayerIndex 属性.....	- 119 -
33.2 FLAG 属性.....	- 120 -
33.3 MapID 属性.....	- 120 -
34. OpCGetAttStructure 类.....	- 120 -
34.1 LayerIndex 属性.....	- 120 -
34.2 MapID 属性.....	- 121 -
35. OpCGetFeatureInformation 类.....	- 121 -
35.1 FeatureID 属性.....	- 121 -

35.2 InformationType 属性.....	- 121 -
35.3 LayerIndex 属性.....	- 121 -
35.4 MapID 属性.....	- 122 -
36. OpCGetFeatureStyleInfo 类.....	- 122 -
36.1 FeatureID 属性.....	- 122 -
36.2 LayerIndex 属性.....	- 122 -
36.3 MapID 属性.....	- 122 -
37. OpCGetGDBInfo 类.....	- 123 -
37.1 MapID 属性.....	- 123 -
38. OpCGetGDBLayerInfo 类.....	- 123 -
38.1 MapID 属性.....	- 123 -
38.2 DataSource 属性.....	- 123 -
38.3 GDBName 属性.....	- 124 -
38.4 Password.....	- 124 -
38.5 userName.....	- 124 -
39. OpCGetGDBName 类.....	- 124 -
39.1 MapID 属性.....	- 124 -
39.2 DataSource 属性.....	- 125 -
39.3 Password.....	- 125 -
39.4 userName.....	- 125 -
40. OpCGetMap 类.....	- 125 -
40.1 DisplayRectangle 属性.....	- 126 -
40.2 FileType 属性.....	- 126 -
40.3 Height 属性.....	- 126 -
40.4 ImageFileName 属性.....	- 126 -
40.5 LayerCount 属性.....	- 126 -
40.6 LayerInfoTree 属性.....	- 127 -
40.7 MapId 属性.....	- 127 -
40.8 Original 属性.....	- 127 -
40.9 Transparent 属性.....	- 127 -
40.10 Width 属性.....	- 127 -
40.11 bufferShow.....	- 128 -
40.12 dynamicNote.....	- 128 -
40.13 followScale.....	- 128 -
40.14 imageQuality.....	- 128 -
40.15 showCoordinate.....	- 128 -
40.16 showScale.....	- 129 -
41. OpCGetMapLst 类.....	- 129 -
41.1 flag 属性.....	- 129 -
42. OpCGetMapLstResult 类.....	- 129 -
42.1 mapDocList 属性.....	- 129 -
42.2 mapDocNum 属性.....	- 129 -
43. OpCGetMapResult 类.....	- 130 -
43.1 ImageFileName 属性.....	- 130 -

43.2 ImageSize 属性.....	- 130 -
44. OpCGetNewMapInfo 类.....	- 130 -
44.1 MapID 属性.....	- 130 -
45. OpCGetOnePage 类.....	- 130 -
45.1 AttSelect 属性.....	- 131 -
45.2 CircleSelect 属性.....	- 131 -
45.3 CircleWithAttSelect 属性.....	- 131 -
45.4 LayerIndex 属性.....	- 131 -
45.5 LineSelect 属性.....	- 132 -
45.6 PntSelect 属性.....	- 132 -
45.7 PntWithAttSelect 属性.....	- 132 -
45.8 PolygonSelect 属性.....	- 132 -
45.9 PolygonWithAttSelect 属性.....	- 133 -
45.10 RecSelect 属性.....	- 133 -
45.11 RectWithAttSelect 属性.....	- 133 -
45.12 SelectType 属性.....	- 133 -
45.13 StartIndex 属性.....	- 134 -
46. OpCLayerAttStructure 类.....	- 134 -
46.1 FieldNumber 属性.....	- 134 -
46.2 FieldTypes 属性.....	- 134 -
46.3 FieldNames 属性.....	- 134 -
46.4 FieldLength 属性.....	- 135 -
47. OpCLayerInformation 类.....	- 135 -
47.1 ChildLayerCount 属性.....	- 135 -
47.2 ChildLayers 属性.....	- 136 -
47.3 DynamicNote 属性.....	- 136 -
47.4 Editable 属性.....	- 136 -
47.5 FeatureClass 属性.....	- 137 -
47.6 FeatureClassID 属性.....	- 137 -
47.7 FieldName 属性.....	- 137 -
47.8 Font 属性.....	- 137 -
47.9 FontColor 属性.....	- 138 -
47.10 FontSize 属性.....	- 138 -
47.11 LayerBound 属性.....	- 138 -
47.12 LayerIndex 属性.....	- 138 -
47.13 LayerName 属性.....	- 138 -
47.14 LayerState 属性.....	- 139 -
47.15 LayerType 属性.....	- 139 -
47.16 MaxScale 属性.....	- 139 -
47.17 MinScale 属性.....	- 139 -
47.18 NoteMode 属性.....	- 139 -
47.19 ScaleSymbolFlag 属性.....	- 140 -
47.20 Selectable 属性.....	- 140 -
47.21 Valid 属性.....	- 140 -

47.22 Visible 属性.....	- 141 -
48. OpCLayerSelectResult 类.....	- 141 -
48.1 Count 属性.....	- 141 -
48.2 FeatureIDs 属性.....	- 141 -
48.3 FieldNames 属性.....	- 142 -
48.4 FieldNumber 属性.....	- 142 -
48.5 LayerIndex 属性.....	- 142 -
48.6 PropertyValues 属性.....	- 142 -
49. OpCLine 类.....	- 142 -
49.1 PointNumber 属性.....	- 143 -
49.2 ThePoints 属性.....	- 143 -
50. OpCLineSelectClass 类.....	- 143 -
50.1 LayerIndex 属性.....	- 143 -
50.2 LineRadius 属性.....	- 143 -
50.3 SelectLine 属性.....	- 144 -
51. OpCLoadMap 类.....	- 144 -
51.1 MapName 属性.....	- 144 -
52. OpCLoadMapResult 类.....	- 144 -
52.1 LayerInfoTree 属性.....	- 144 -
52.2 LayerNumber 属性.....	- 145 -
52.3 MapBound 属性.....	- 145 -
52.4 MapID 属性.....	- 145 -
53. OpCMultiLine 类.....	- 145 -
53.1 LineNumber 属性.....	- 145 -
53.2 TheLines 属性.....	- 145 -
54. OpCNetAnalysisResult 类.....	- 146 -
54.1 Start 属性.....	- 146 -
54.2 End 属性.....	- 146 -
54.3 InnerNodeNumber 属性.....	- 146 -
54.4 InnerNodes 属性.....	- 147 -
54.5 PathNumber 属性.....	- 147 -
54.6 Paths 属性.....	- 147 -
54.7 SResult 属性.....	- 147 -
54.8 layerSelectResults 属性.....	- 148 -
55. OpCOverlayByIDPntSeq 类.....	- 148 -
55.1 iFeatureNum 属性.....	- 148 -
55.2 pInfo 属性.....	- 148 -
56. OpCPoint3D 类.....	- 148 -
56.1 X 属性.....	- 148 -
56.2 Y 属性.....	- 149 -
56.3 Z 属性.....	- 149 -
57. OpCPointSelectClass 类.....	- 149 -
57.1 Distance 属性.....	- 149 -
57.2 Dot 属性.....	- 149 -

57.3 LayerIndex 属性.....	- 150 -
57.4 MapID 属性.....	- 150 -
57.5 PageIndex 属性.....	- 150 -
57.6 recordCountOfPage 属性.....	- 150 -
58. OpCPointWithAttSelectStru 类.....	- 151 -
58.1 Question 属性.....	- 151 -
59. OpCPolygonSelectClass 类.....	- 151 -
59.1 PointNum 属性.....	- 151 -
59.2 Polygon 属性.....	- 151 -
59.3 LayerIndex 属性.....	- 152 -
59.4 MapID 属性.....	- 152 -
59.5 PageIndex 属性.....	- 152 -
59.6 PolygonNum 属性.....	- 152 -
59.7 recordCountOfPage 属性.....	- 153 -
60. OpCPolygonWithAttSelectStru 类.....	- 153 -
60.1 Question 属性.....	- 153 -
61. OpCPrjTran 类.....	- 153 -
61.1 BZoneType 属性.....	- 153 -
61.2 DAngUnit 属性.....	- 153 -
61.3 DLon 属性.....	- 154 -
61.4 IPointNum 属性.....	- 154 -
61.5 IProjTypeId 属性.....	- 154 -
61.6 ISpheroid 属性.....	- 154 -
61.7 IType 属性.....	- 155 -
61.8 IZone 属性.....	- 155 -
61.9 ThePoint 属性.....	- 155 -
61.10 Unit 属性.....	- 155 -
62. OpCPrjTranRlt 类.....	- 155 -
62.1 PrjPointNum 属性.....	- 155 -
62.2 PrjRltPoint 属性.....	- 156 -
63. OpCRectangleSelectClass 类.....	- 156 -
63.1 LayerIndex 属性.....	- 156 -
63.2 RectangleArea 属性.....	- 156 -
63.3 MapID 属性.....	- 156 -
63.4 PageIndex 属性.....	- 157 -
63.5 recordCountOfPage 属性.....	- 157 -
64. OpCRectWithAttSelectStru 类.....	- 157 -
64.1 Question 属性.....	- 157 -
65. OpCRegion 类.....	- 158 -
65.1 LineNumber 属性.....	- 158 -
65.2 TheLines 属性.....	- 158 -
66. OpCSelectResult 类.....	- 158 -
66.1 LayerCount 属性.....	- 158 -
66.2 TotalCount 属性.....	- 159 -

66.3 LayerSelectResults 属性.....	- 160 -
67. OpCSpatialBuffer 类.....	- 161 -
67.1 LayerIndex 属性.....	- 161 -
67.2 Radius 属性.....	- 161 -
67.3 FeatureClassName 属性.....	- 162 -
67.4 FeatureNumber 属性.....	- 162 -
67.5 FeatureIDS 属性.....	- 162 -
67.6 MapID 属性.....	- 162 -
67.7 OverlayOnCurMap 属性.....	- 163 -
67.8 mapDisplayBound 属性.....	- 163 -
67.9 mapHeight 属性.....	- 163 -
67.10 mapWidth 属性.....	- 163 -
68. OpCSpatialCircleClip 类.....	- 164 -
68.1 LayerIndexLst 属性.....	- 164 -
68.2 LayerIndex 属性.....	- 164 -
68.3 MapID 属性.....	- 164 -
68.4 Radius 属性.....	- 164 -
68.5 FeatureClassName 属性.....	- 165 -
68.6 ClipType 属性.....	- 165 -
68.7 Center 属性.....	- 165 -
68.8 SetNearDis 方法.....	- 165 -
69. OpCSpatialOperResult 类.....	- 166 -
70. OpCSpatialOverlay 类.....	- 166 -
70.1 GDBDataSourceName1 属性.....	- 166 -
70.2 GDBDataSourceName2 属性.....	- 166 -
70.3 GDBName1 属性.....	- 167 -
70.4 GDBName2 属性.....	- 167 -
70.5 FeatureIDLst1 属性.....	- 167 -
70.6 FeatureIDLst2 属性.....	- 167 -
70.7 LayerIndex1 属性.....	- 168 -
70.8 LayerIndex2 属性.....	- 168 -
70.9 OverlayType 属性.....	- 168 -
70.10 OverlayGDBName 属性.....	- 168 -
70.11 MapID 属性.....	- 169 -
70.12 password1 属性.....	- 169 -
70.13 password2 属性.....	- 169 -
70.14 userName1 属性.....	- 169 -
70.15 userName2 属性.....	- 170 -
71. OpCSpatialOverlayByID 类.....	- 170 -
71.1 iNumOfLst1 属性.....	- 170 -
71.2 iNumOfLst2 属性.....	- 170 -
71.3 layerIndex1 属性.....	- 170 -
71.4 layerIndex2 属性.....	- 171 -
71.5 mapID 属性.....	- 171 -

71.6 OverlayType 属性.....	- 171 -
71.7 pFeatureIDLst1 属性.....	- 171 -
71.8 pFeatureIDLst2 属性.....	- 172 -
72. OpCSpatialPolygonClip 类.....	- 172 -
72.1 LayerIndex 属性.....	- 172 -
72.2 LayerIndexLst 属性.....	- 172 -
72.3 ClipType 属性.....	- 172 -
72.4 FeatureClassName 属性.....	- 173 -
72.5 PolygonNum 属性.....	- 173 -
72.6 Points 属性.....	- 173 -
72.7 LpPointNum 属性.....	- 173 -
72.8 MapID 属性.....	- 174 -
72.9 SetNearDis 方法.....	- 174 -
73. OpCTest 类.....	- 174 -
73.1 op 属性.....	- 174 -
74. OpCUniformResult 类.....	- 175 -
74.1 OperCode 属性.....	- 175 -
74.2 Flag 属性.....	- 175 -
74.3 Information 属性.....	- 175 -
75. OpCUpdateFeature 类.....	- 175 -
75.1 FeatureID 属性.....	- 175 -
75.2 LayerIndex 属性.....	- 176 -
75.3 MapID 属性.....	- 176 -
75.4 NewFeature 属性.....	- 176 -
76. OpCUpdateFeatureAttribute 类.....	- 176 -
76.1 LayerIndex 属性.....	- 176 -
76.2 FeatureID 属性.....	- 177 -
76.3 FieldNumber 属性.....	- 177 -
76.4 NewFieldValues 属性.....	- 177 -
76.5 MapID 属性.....	- 177 -
77. OpCUpdateFeatureStyleInfo 类.....	- 178 -
77.1 LayerIndex 属性.....	- 178 -
77.2 FeatureId 属性.....	- 178 -
77.3 MapID 属性.....	- 178 -
78. OpCUpdateLayerInfomation 类.....	- 178 -
78.1 LayerIndex 属性.....	- 178 -
78.2 MapID 属性.....	- 179 -
78.3 UpdatedLayer 属性.....	- 179 -
79. OpCUpdateLayerLst 类.....	- 179 -
79.1 layerInfo 属性.....	- 179 -
79.2 layerNum 属性.....	- 179 -
79.3 mapID 属性.....	- 179 -
80. GridServerInfo 类.....	- 180 -
80.1 HdfInfoArray 属性.....	- 180 -

80.2 ServerIP 属性.....	- 180 -
80.3 ServerPort 属性.....	- 180 -
81. HDFinfo 类.....	- 180 -
81.1 Name 属性.....	- 181 -
81.2 ShowLevel 属性.....	- 181 -
81.3 Xmax 属性.....	- 181 -
81.4 Ymax 属性.....	- 181 -
81.5 Xmin 属性.....	- 181 -
81.5 Ymin 属性.....	- 182 -
82. IMSConfigManager 类.....	- 182 -
82.1 GetDataProvider 方法.....	- 182 -
82.2 GetDBConnectionString 方法.....	- 182 -
82.3 GetGridServerInfo 方法.....	- 182 -
82.4 GetNeedGisMgrInfo 方法.....	- 182 -
82.5 GetVectorServerInfo 方法.....	- 183 -
82.6 IMSConfigManager 方法.....	- 183 -
83. MapDocInfo 类.....	- 183 -
83.1 MapDocInfo 属性.....	- 183 -
83.2 Name 属性.....	- 183 -
83.3 Xmax 属性.....	- 184 -
83.4 Ymax 属性.....	- 184 -
83.5 Xmin 属性.....	- 184 -
83.6 Ymin 属性.....	- 184 -
84. OpCFeatureGeometries 类.....	- 184 -
84.1 iFeatureNum 方法.....	- 184 -
84.1 pInfo 方法.....	- 185 -
85. OpCGetFclsRect 类.....	- 185 -
85.1 ClassToXML 方法.....	- 185 -
85.2 OpCGetFclsRect 构造函数.....	- 185 -
85.3 XMLToClass 方法.....	- 186 -
85.4 FclsName 属性.....	- 186 -
85.5 HDFName 属性.....	- 186 -
85.6 LayerType 属性.....	- 186 -
85.7 NetServiceName 属性.....	- 186 -
85.8 Password 属性.....	- 187 -
85.9 UserName 属性.....	- 187 -
86. OpCGetFclsSel 类.....	- 187 -
86.1 ClassToXML 方法.....	- 187 -
86.2 OpCGetFclsSel 构造函数.....	- 187 -
86.3 XMLToClass 方法.....	- 188 -
86.4 FclsName 属性.....	- 188 -
86.5 HDFName 属性.....	- 188 -
86.6 LayerType 属性.....	- 188 -
86.7 NetServiceName 属性.....	- 189 -

86.8 Password 属性.....	- 189 -
86.9 UserName 属性.....	- 189 -
87. OpCGetSingleFclsGeo 类.....	- 189 -
87.1 ClassToXML 方法.....	- 189 -
87.2 OpCGetSingleFclsGeo 方法.....	- 190 -
87.3 XMLToClass 方法.....	- 190 -
87.4 FeatureID 属性.....	- 190 -
87.5 FeatureNum 属性.....	- 190 -
88. OpCSpatialBufferByGeom 类.....	- 190 -
88.1 ClassToXML 方法.....	- 191 -
88.2 OpCSpatialBufferByGeom 方法.....	- 191 -
88.3 XMLToClass 方法.....	- 191 -
88.4 FeatureGeometry 属性.....	- 191 -
88.5 FeatureNumber 属性.....	- 192 -
88.6 MapID 属性.....	- 192 -
88.7 Radius 属性.....	- 192 -
88.8 RebuildFlag 属性.....	- 192 -
88.9 ResultFeatureClassName 属性.....	- 192 -
89. OpCSpatialDiffer 类.....	- 193 -
89.1 ClassToXML 方法.....	- 193 -
89.2 OpCSpatialDiffer 方法.....	- 193 -
89.3 XMLToClass 方法.....	- 193 -
89.4 FeatureGeometry1 属性.....	- 194 -
89.5 FeatureGeometry2 属性.....	- 194 -
90. OpCSpatialOverlayByGeom 类.....	- 194 -
90.1 ClassToXML 方法.....	- 194 -
90.2 OpCSpatialOverlayByGeom 方法.....	- 194 -
90.3 XMLToClass 方法.....	- 195 -
90.4 FeatureGeometry1 属性.....	- 195 -
90.5 FeatureGeometry2 属性.....	- 195 -
90.6 OverlayType 属性.....	- 195 -
91. VectorServerInfo 类.....	- 195 -
91.1 LogPath 属性.....	- 196 -
91.2 MapDocInfoArray 属性.....	- 196 -
91.3 NeedGisMgr 属性.....	- 196 -
91.4 ServerIP 属性.....	- 196 -
91.5 ServerPort 属性.....	- 196 -
91.6 WriteLog 属性.....	- 197 -
92. MapSetting 类.....	- 197 -
92.1 GetEditableLayersIndex 方法.....	- 197 -
92.2 GetGridLayer 方法.....	- 197 -
92.3 GetGridLayer 方法.....	- 197 -
92.4 GetGridLayer 方法.....	- 198 -
92.5 GetLayerStatus 方法.....	- 198 -

92.6 GetLayerType 方法.....	- 198 -
92.7 GetMapNameConfig 方法.....	- 199 -
92.8 GetNetLayerIndex 方法.....	- 199 -
92.9 GetSelectableLayersIndex 方法.....	- 199 -
92.10 GetVisibleLayersIndex 方法.....	- 200 -
92.11 SetLayerStatus 方法.....	- 200 -
92.12 UpdateMap 方法.....	- 200 -
93. ScriptWriter 类.....	- 201 -
93.1 AddClassData 方法.....	- 201 -
93.2 ClearJSONData 方法.....	- 201 -
93.3 GetJSONArrStr 方法.....	- 201 -
93.4 GetJSONObjStr 方法.....	- 202 -
93.5 ScriptWriter 方法.....	- 202 -
93.6 ScriptWriter 方法.....	- 202 -
93.7 SetClassAttribute 方法.....	- 203 -
93.8 Write 方法.....	- 203 -
93.9 WriteJSONArray 方法.....	- 203 -
A.1.2 .NET 结构体.....	- 204 -
1. ChineseWordUnit 结构体.....	- 204 -
1.1 power 属性.....	- 204 -
1.2 word 属性.....	- 204 -
1.3 ChineseWordUnit 方法.....	- 205 -
2. Dot_2D 结构体.....	- 205 -
2.1 X 属性.....	- 205 -
2.2 Y 属性.....	- 205 -
2.3 Dot_2D 方法.....	- 205 -
3. GDBInfomation 结构体.....	- 206 -
3.1 dataSourceName 属性.....	- 206 -
3.2 HDFName 属性.....	- 206 -
3.3 layerName 属性.....	- 206 -
4. HDFConfig 结构体.....	- 207 -
4.1 GisServerIP 属性.....	- 207 -
4.2 GisServerPort 属性.....	- 207 -
4.3 HDFName 属性.....	- 207 -
4.4 MapRect 属性.....	- 208 -
4.5 StartLevel 属性.....	- 208 -
4.6 DealSize 属性.....	- 208 -
4.7 Degree 属性.....	- 208 -
4.8 ImgWidth 属性.....	- 209 -
4.9 HDFConfig 方法.....	- 209 -
4.10 HDFConfig 方法.....	- 209 -
4.11 HDFConfig 方法.....	- 210 -
4.12 HDFConfig 方法.....	- 210 -
5. MapRectangle 结构体.....	- 211 -

5.1 XMinMap 属性.....	- 211 -
5.2 XMaxMap 属性.....	- 211 -
5.3 YMinMap 属性.....	- 211 -
5.4 YMaxMap 属性.....	- 211 -
5.5 MapRectangle 方法.....	- 212 -
6. OpCPoint2D 结构体.....	- 212 -
6.1 X 属性.....	- 212 -
6.2 Y 属性.....	- 212 -
6.3 OpCPoint2D 方法.....	- 212 -
7. OpCRect 结构体.....	- 213 -
7.1 XMax 属性.....	- 213 -
7.2 XMin 属性.....	- 213 -
7.3 YMax 属性.....	- 213 -
7.4 YMin 属性.....	- 213 -
7.5 OpCRect 方法.....	- 214 -
A.1.3 .NET 枚举.....	- 214 -
1. DatabaseType 枚举.....	- 214 -
2. ImageFormatEnum 枚举.....	- 214 -
3. ImageQuality 枚举.....	- 215 -
4. LayerStatusEnum 枚举.....	- 215 -
5. OutputType 枚举.....	- 215 -
6. QueryTypeEnum 枚举.....	- 215 -
7. FieldTypeEnum 枚举.....	- 216 -
8. LayerTypeEnum 枚举.....	- 216 -
9. LayerTypeEnum 枚举.....	- 217 -
10. SpatialRltDbTypeEnum 枚举.....	- 217 -
11. TopoTypeEnum 枚举.....	- 217 -
A.2 JAVA 函数库.....	- 217 -
A.2.1 JAVA 包.....	- 218 -
1. com.mapgis.ims.see.com 包.....	- 218 -
1.1 CMapViewPort 类.....	- 218 -
1.2 CHDFConfig 类.....	- 221 -
1.3 CMapRectangle 类.....	- 222 -
1.4 CHotSpot 类.....	- 223 -
1.5 CGetVipData 类.....	- 224 -
2. com.mapgis.ims.see.util 包.....	- 225 -
2.1 GISDataQuery 类.....	- 225 -
2.2 CPathQuery 类.....	- 231 -
3.com.mapgis.ims.see.buscom 包.....	- 231 -
3.1 CBusCom 类.....	- 231 -
3.2 CBusData 类.....	- 232 -
4.com.mapgis.ims.pro.function 包.....	- 233 -
4.1 MapViewPort 类.....	- 233 -
4.2 CAddLayerInfo 类.....	- 257 -

4.3 CAppendAArcLine 类.....	- 260 -
4.4 CAppendAPoint 类.....	- 263 -
4.5 CAppendARegion 类.....	- 266 -
4.6 CAttSelectStru 类.....	- 269 -
4.7 CBasicNetAnalysis 类.....	- 270 -
4.8 CCircleSelectClass 类.....	- 272 -
4.9 CCircleWithAttSelectStru 类.....	- 274 -
4.10 CDeleteFeature 类.....	- 276 -
4.11 CDrawMap 类.....	- 278 -
4.12 CFeatureGeometry 类.....	- 278 -
4.13 CFeatureStyleInfo 类.....	- 280 -
4.14 CGeomInfo 类.....	- 282 -
4.15 CGetFeatureInformation 类.....	- 283 -
4.16 CGetFeatureStyleInfo 类.....	- 284 -
4.17 CGetGDBInfo 类.....	- 285 -
4.18 CGetGDBLayerInfo 类.....	- 286 -
4.19 CGetGDBName 类.....	- 287 -
4.20 CGetOnePage 类.....	- 289 -
4.21 CLayerAttStructure 类.....	- 292 -
4.22 CLayerSelectResult 类.....	- 293 -
4.23 CLine 类.....	- 295 -
4.24 CLineSelectClass 类.....	- 296 -
4.25 CMultiLine 类.....	- 298 -
4.26 CNetAnalysisResult 类.....	- 298 -
4.27 CPoint2D 类.....	- 300 -
4.28 CPointSelectClass 类.....	- 301 -
4.29 CPolygonSelectClass 类.....	- 303 -
4.30 CPolygonWithAttSelectStru 类.....	- 305 -
4.31 CRect 类.....	- 308 -
4.32 CRectangleSelectClass 类.....	- 310 -
4.33 CRectWithAttSelectStru 类.....	- 311 -
4.34 CRegion2D 类.....	- 313 -
4.35 CSelectResult 类.....	- 314 -
4.36 CSpatialBuffer 类.....	- 315 -
4.37 CSpatialCircleClip 类.....	- 318 -
4.38 CSpatialOperResult 类.....	- 321 -
4.39 CSpatialOverlay 类.....	- 322 -
4.40 CSpatialPolygonClip 类.....	- 327 -
4.41 CUniformResult 类.....	- 330 -
4.42 CUpdateFeature 类.....	- 331 -
4.43 CUpdateFeatureAttribute 类.....	- 332 -
4.44 CUpdateFeatureStyleInfo 类.....	- 334 -
4.45 CUpdateLayerLst 类.....	- 337 -
5.com.mapgis.ims.pro.chart 包.....	- 338 -

5.1 Chart 类（旧版本使用） .....	338 -
6.com.mappgis.ims.pro.layer 包 .....	339 -
6.1 CMapConfig 类 .....	339 -
6.2 CMapDisplay 类 .....	340 -
6.3 CMapLayerSetting 类 .....	341 -
B.客户端函数库参考 .....	343 -
1. IMSMapType .....	343 -
2. IMSOperType .....	344 -
3. ZDEvent .....	345 -
3.1 addListener .....	345 -
3.2 removeListener .....	345 -
3.3 clearEvent .....	346 -
4. IMSMapConfig .....	346 -
4.1 IMSMapConfig .....	346 -
4.2 this.gridImageUrl .....	347 -
4.3 this.vectImageUrl .....	347 -
4.4 this.getVectCoorPage .....	347 -
5. IMSMaps .....	348 -
5.1 IMSMaps .....	348 -
5.2 setRestoreParam .....	348 -
5.3 setRestoreParam .....	349 -
5.4 restoreMap .....	349 -
5.5 jumpMap .....	349 -
5.6 jumpMapByLev .....	350 -
5.7 pntLocation .....	350 -
5.8 setCurOper .....	351 -
5.9 screenToLogic .....	351 -
5.10 logicToScreen .....	352 -
5.11 addControl .....	352 -
5.12 removeControl .....	353 -
5.13 loadAllCtl .....	353 -
5.14 getCurLev .....	353 -
5.15 refresh .....	354 -
5.16 moveTo .....	354 -
5.17 getClickCoor .....	354 -
5.18 getCurWinLog .....	355 -
5.19 getCenterPnt .....	355 -
5.20 getMapViewSize .....	356 -
5.21 closeGridLayer .....	356 -
5.22 openGridLayer .....	356 -
5.23 openVectMap .....	357 -
5.24 closeVectLayer .....	357 -
5.25 getGridCode .....	357 -
5.26 getGridCodes .....	358 -

5.27 gridCodeConvert.....	- 358 -
5.28 addEventListener.....	- 359 -
5.29 removeEventListener.....	- 359 -
5.30 setRectDrawStyle.....	- 360 -
5.31 setCircleDrawStyle.....	- 360 -
5.32 setLineDrawStyle.....	- 360 -
5.33 setPolygonDrawStyle.....	- 361 -
5.34 clearOper.....	- 361 -
事件.....	- 362 -
6. IMSNavigation.....	- 362 -
6.1 IMSNavigation.....	- 362 -
6.2 setImage.....	- 363 -
6.3 setControlPos.....	- 363 -
6.4 moveSpeed.....	- 363 -
6.5 setMapShade.....	- 364 -
7. IMSMarker.....	- 364 -
7.1 IMSMarker.....	- 364 -
7.2 setMarkerImg.....	- 365 -
7.3 setURL.....	- 365 -
7.4 setStyle.....	- 366 -
7.5 setMarkerTitle.....	- 366 -
7.6 getMarkerPoint.....	- 367 -
8. IMSScale.....	- 367 -
8.1 IMSScale.....	- 367 -
8.2 setUnits.....	- 367 -
8.3 setPos.....	- 368 -
8.4 setFontColor.....	- 368 -
9. IMSLogo.....	- 369 -
9.1 IMSLogo.....	- 369 -
9.2 setlogo.....	- 369 -
9.3 setPos.....	- 369 -
10. IMSSearch.....	- 370 -
10.1 IMSSearch.....	- 370 -
10.2 setSelectByHitting.....	- 370 -
10.3 setSelectByRect.....	- 371 -
10.4 setSelectByCircle.....	- 372 -
10.5 setSelectByPolygon.....	- 372 -
10.6 setSelectByLine.....	- 373 -
11. IMSVIPPnts.....	- 373 -
11.1 IMSVIPPnts.....	- 373 -
11.2 setImgSrc.....	- 374 -
11.3 loadData.....	- 374 -
11.4 showPnts.....	- 375 -
11.5 hidePnts.....	- 375 -

11.6 clearPnts.....	- 376 -
12. IMSMeasure.....	- 376 -
12.1 IMSMeasure.....	- 376 -
12.2 setDistanceUnits.....	- 376 -
12.3 setAreaUnits.....	- 377 -
12.4 setTooltipStyle.....	- 378 -
13. IMSPalette.....	- 378 -
13.1 IMSPalette.....	- 378 -
13.2 setStyle.....	- 378 -
13.3 draw.....	- 379 -
13.4 setStartPnt.....	- 379 -
13.5 drawing.....	- 380 -
13.6 appendPoint.....	- 380 -
13.7 dispose.....	- 381 -
13.8 flash.....	- 381 -
14. IMSEagleMap.....	- 381 -
14.1 IMSEagleMap.....	- 382 -
14.2 eagleJumpingMap.....	- 382 -
14.3 setButtonImage.....	- 383 -
14.4 setBorderColor.....	- 383 -
14.5 setRectBorderColor.....	- 383 -
14.6 setRectBackColor.....	- 384 -
14.7 setViewStatus.....	- 384 -
14.8 isOpen.....	- 385 -
14.9 changeViewStatus.....	- 385 -
15. IMSHotSpots.....	- 385 -
15.1 IMSHotSpots.....	- 386 -
15.2 getHsInfo.....	- 386 -
16. IMSCarMove.....	- 386 -
16.1 IMSCarMove.....	- 387 -
16.2 setLineInfo.....	- 387 -
16.3 drawLine.....	- 387 -
16.4 openNormalNodeDiv.....	- 388 -
16.5 closeNormalNodeDiv.....	- 388 -
16.6 openSpecialNodeDiv.....	- 389 -
16.7 closeSpecialNodeDiv.....	- 389 -
16.8 move.....	- 389 -
16.9 pause.....	- 390 -
16.10 stop.....	- 390 -
16.11 clearVar.....	- 391 -
17. RtMenu.....	- 391 -
17.1 RtMenu.....	- 391 -
17.2 ShowMenu.....	- 392 -
18. 全局方法.....	- 393 -

18.1 String.prototype.trim.....	- 393 -
18.2 Array.prototype.max.....	- 393 -
18.3 Array.prototype.min.....	- 393 -
18.4 StopBubble.....	- 394 -
18.5 StopDefault.....	- 394 -
18.6 StopDivBubble.....	- 394 -
18.7 GetImgSize.....	- 395 -
18.8 CreatDiv.....	- 395 -
18.9 GetAbsolutePos.....	- 396 -
18.10 AJAXRequest.....	- 396 -

# A. 服务器端组件函数库

## A.1 .NET 函数库

**MAPGISIMS.EnterpriseLib** 命名空间应用于 MAPGIS7-IMS 行业版 (Enterprise)，里面包含了公交换乘、网络分析、中文分词、地图视图、地图查询等类，方便用户进行二次开发。

### A.1.1 .NET 类库

#### 1. BusCom 类

根据入库的公交数据进行换乘分析，提供最优换乘方案。

此类与数据库紧密结合，当初始化此类时，首先是将数据库中的数据读取到缓存中，然后再根据相关算法，对用户的输入值进行分析，返回。

#### 1.1 构造函数

```
public BusCom(string strConnection, MAPGISIMS.EnterpriseLib.DatabaseType dbType)
```

**说明：**

初始化带连接字符串的 BusCom 类的新实例。

**参数：**

strConnection 指的是建立到数据库的连接,在 strConnection 字符串中必须指定数据库服务器所在地址(IP/名称)、连接的用户名、密码，以及所要连接的数据库名。

dbType 指的是所要连接的数据库类型。

**返回值：**

无

**备注：**

BusCom 的构造参数

**示例**

```
string strConnection ="data source=127.0.0.1;  
initial catalog=CityPosInfo;user id=sa;password=guest";  
BusCom busObj = new BusCom(strConnection,DatabaseType.SQLServer);
```

#### 1.2 LoadData 方法

```
Public bool LoadData()
```

**说明：**

将数据库所有数据读入缓存中,数据读取完之后就关闭与数据库的连接。

**参数:**

无

**返回值:**

布尔值 返回 true 表示数据读入成功, 返回 false 则失败。

**备注:**

无

**示例:**

```
BusCom busObj = new BusCom(strConnection,DatabaseType.SQLServer);
busObj.LoadData();
```

### 1.3 GetPassLineByStopName 方法

**Public short GetPassLineByStopName(string stopName,out string lineNameLst)**

**说明:**

通过站点名称得到经过该站的所有线路名称。

**参数:**

stopName 表示站点名称。

lineNameLst 表示经过 stopName 站点的所有结果线路名称集合。

**返回值:**

返回-1 代表失败,1 代表成功

**备注:**

在调用该函数前需调用 LoadData 加载数据 。

**示例:**

```
BusCom busObj = new BusCom(strConnection,DatabaseType.SQLServer);
busObj.LoadData();
string linename = "";
short i = busObj.GetPassLineByStopName("鲁巷", out linename);
```

### 1.4 GetStopListByLineName 方法

**Public short GetStopListByLineName (string lineName,out string stopLst)**

**说明:**

通过线路名称得到该条线路上的所有站点名称。

**参数:**

lineName 表示线路名称 (如 709 或 709 路)。

stopLst 表示 stopName 线路上的所有站点名称。

**返回值:**

返回-1 代表失败,1 代表成功

**备注:**

无

**示例:**

```
BusCom busObj = new BusCom(strConnection,DatabaseType.SQLServer);
```

```
busObj.LoadData();  
string stopLst = "";  
short i = busObj.GetStopListByLineName ("709",out stopLst);
```

## 1.5 GetChangeRlt 方法

```
public void GetChangeRlt(string startStop, string endStop, out string xmlStr,  
MAPGISIMS.EnterpriseLib.OutputType opType)
```

**说明:**

获取公交换乘方案。

**参数:**

startSto: p 表示起点车站名。

endStop: 表示终点车站名。

xmlStr: 表示返回的结果 xml 或字符串。

opType: 输出结果类型。xml 表示 XML 格式,str 表示字符串格式。

**返回值:**

无

**备注:**

无

**示例:**

```
BusCom busObj = new BusCom(strConnection,DatabaseType.SQLServer);  
busObj.LoadData();  
string coordinate = "";  
busObj.GetChangeRlt(staPos,endPos,out coordinate,OutputType.str);
```

## 2. ChineseParse 类

中文分词类

### 2.1 ChineseParse 方法

```
public ChineseParse ( string path )
```

**说明:**

初始化具有指定字典文件路径的ChineseParse类的新实例

**参数:**

path: 字典文件路径

**返回值:**

无

**备注:**

ChieseParse 类的构造函数

**示例:**

```
string path = HttpContext.Current.Server.MapPath(".") +
```

```
@\"ChineseDictionary.txt\" ;  
ChineseParse cp = new ChineseParse(path);  
ChineseWordUnit[] arryWords = cp.ParseChinese(key);
```

## 2.2 ParseChinese 方法

**public ChineseWordUnit[] ParseChinese ( string s )**

**说明:**

分析输入的字符串，将其切割成单个的词语。

**参数:**

s: 待切割的字符串

**返回值:**

所切割得到的中文词语数组

**备注:**

得到 ChineseWordUnit 结构体类型的数组

**示例:**

```
string path = HttpContext.Current.Server.MapPath(".") +  
    @\"ChineseDictionary.txt\" ;  
ChineseParse cp = new ChineseParse(path);  
ChineseWordUnit[] arryWords = cp.ParseChinese(key);
```

## 3. CLayerInformation 类

图层状态,图层名,逻辑范围

### 3.1 LayerAlias 属性

**Public string LayerAlias**

**说明**

设置指定图层别名

**备注**

该属性隶属于 CLayerInformation 对象

### 3.2 LayerBound 属性

**Public string LayerBound**

**说明**

设置指定图层逻辑范围

**备注**

该属性隶属于 CLayerInformation 对象

### 3.3 LayerName 属性

**public String LayerName**

说明

设置指定图层名称

备注

该属性隶属于 CLayerInformation 对象

### 3.4 LayerStatus 属性

**Public String LayerStatus**

说明

设置指定图层状态

备注

该属性隶属于 CLayerInformation 对象

## 4. CLayerObj 类

图层对象

### 4.1 CLayerObj 方法

**public CLayerObj()**

说明:

ClayerObj的构造函数

备注:

CLayerObj 类的构造函数

### 4.2 getLayerAttStruct 方法

**public MAPGISIMS.EnterpriseLib.OpCLayerAttStructure getLayerAttStruct()**

说明:

获取图层属性结构。

参数:

无

返回值:

图层的属性结构

备注:

得到 OpCLayerAttStructure 类

### 4.3 getLayerInfo 方法

**public MAPGISIMS.EnterpriseLib.OpCLayerInformation getLayerInfo()**

说明:

获取图层信息。

参数:

无

返回值:

图层信息

备注:

得到 OpCLayerInformation 类

### 4.4 getLayerType 方法

**public int getLayerType()**

说明:

获取图层类型。

参数:

无

返回值:

图层类型

### 4.5 setLayerAttStruct 方法

**public int setLayerAttStruct(MAPGISIMS.EnterpriseLib.OpCLayerAttStructure layerAttStru)**

说明:

设置图层属性结构。

参数:

layerAttStru: 图层属性结构

返回值:

操作是否成功, 大于 0 成功, 小于 0 失败

备注:

隶属于 CLayerObj 类

### 4.6 setLayerInfo 方法

**public int setLayerInfo(MAPGISIMS.EnterpriseLib.OpCLayerInformation layerInformation)**

说明:

设置图层信息。

**参数:**

layerInformation: 图层信息

**返回值:**

操作是否成功, 大于 0 成功, 小于 0 失败

**备注:**

隶属于 CLayerObj 类

## 4.7 setLayerType 方法

**public int setLayerType(int type)**

**说明:**

设置图层类型。

**参数:**

type: 图层类型

**返回值:**

操作是否成功, 大于 0 成功, 小于 0 失败

**备注:**

隶属于 CLayerObj 类

## 5. CMapInformation 类

地图文档对象

### 5.1 Layers 属性

**public MAPGISIMS.EnterpriseLib.CLayerInformation[] Layers**

**说明**

图层信息

**备注**

该属性隶属于 CMapInformation 对象

**示例**

```

//声明函数中返回的地图文档中图层个数
layerNum = -1;
//声明函数的返回值
int sLayerNum = -1, sItem = -1;
//得到地图文档中图层的个数
sLayerNum = viewport.MapObj.GetLayerNum(out layerNum);
mapDocArr[0].Layers = new CLayerInformation[layerNum];
for (int i = 0; i < layerNum; i++)
{
    mapDocArr[0].Layers[i] = new CLayerInformation();
    //依次向集合中添加图层
    rdat.Append(i);
    //依次得到图层对象
    sItem = viewport.MapObj.GetItem(rdat, out pMapLayer);
    if (sItem == 1)//获取图层信息成功
    {
        string layerName = pMapLayer.LayerName;//得到图层名称
        mapDocArr[0].Layers[i].LayerName = layerName;
    }
}

```

## 5.2 MapAlias 属性

### public string MapAlias

说明

地图别名

备注

该属性隶属于 CMapInformation 对象

示例

```

mapDocArr = new CMapInformation[1];
mapDocArr[0] = new CMapInformation();
mapDocArr[0].MapName = (string)Session["MapName"];
mapDocArr[0].MapAlias = (string)Session["MapName"];

```

## 5.3 MapName 属性

### public string MapName

说明

地图名称

备注

该属性隶属于 CMapInformation 对象

示例

```

mapDocArr = new CMapInformation[1];
mapDocArr[0] = new CMapInformation();
mapDocArr[0].MapName = (string)Session["MapName"];
mapDocArr[0].MapAlias = (string)Session["MapName"];

```

## 5.4 MapBound 属性

### public string MapBound

说明

地图逻辑范围

备注

该属性隶属于 CMapInformation 对象

## 6. CMapView 类

地图视图类

### 属性

#### 6.1 AsyncOper 属性

**public bool AsyncOper**

说明

设置与 GIS 服务器的通信是否为异步调用方式

备注

该属性隶属于 CMapView 对象

#### 6.2 EagleBound 属性

**public MAPGISIMS.EnterpriseLib.OpCRect EagleBound**

说明

鹰眼图像逻辑范围

备注

该属性隶属于 CMapView 对象

#### 6.3 EagleImageName 属性

**public string EagleImageName**

说明

鹰眼图像名称

备注

该属性隶属于 CMapView 对象

#### 6.4 GetFileFlag 属性

**public bool GetFileFlag**

说明

空间分析时是否在发分析请求的时候就获取结果文件

备注

该属性隶属于 CMapView 对象

#### 6.5 GisServerAddress 属性

**Public System.Net.IPEndPoint GisServerAddress**

说明

GisAddress

备注

该属性隶属于 CMapView 对象

## 6.6 GisServerManagerAddress 属性

**Public System.Net.IPEndPoint GisServerManagerAddress**

说明

gisMaAddress

备注

该属性隶属于 CMapView 对象

## 6.7 ImageFormat 属性

**public MAPGISIMS.EnterpriseLib.ImageFormatEnum ImageFormat**

说明

读取或设置生成地图图像类型

0-.bmp 1-.gif 2 -.png 3-.jpeg

备注

该属性隶属于 CMapView 对象

示例

```
viewport.ImageFormat = ImageFormatEnum.gif;
string imgFileName = viewport.CreateImageName
(Session.SessionID, viewport.ImageFormat);
byte[] data = viewport.GetBinaryImage(imgFileName);
Response.BinaryWrite(data);
```

## 6.8 ImageHeight 属性

**Public int ImageHeight**

说明

读取或设置地图图像的高

备注

该属性隶属于 CMapView 对象

例子

```
viewport.ImageWidth = width;
viewport.ImageHeight = height;
```

## 6.9 ImageWidth 属性

**Public int ImageWidth**

说明

读取或设置地图图像的宽

备注

该属性隶属于 CMapView 对象

示例

```
viewport.ImageWidth = width;  
viewport.ImageHeight = height;
```

## 6.10 LastSelectResult 属性

**public MAPGISIMS.EnterpriseLib.OpCSelectResult LastSelectResult**

说明

最后一次的查询结果

备注

该属性隶属于 CMapView 对象

示例

```
protected OpCSelectResult selRlt;  
  
selRlt = viewport.LastSelectResult;
```

## 6.11 LayerAttributes 属性

**public MAPGISIMS.EnterpriseLib.CMpAtt LayerAttributes**

说明

获取某一图层属性信息，如属性字段个数、名称等

备注

该属性隶属于 CMapView 对象

示例

```
string[] fieldName = null;  
//取指定图层的属性字段名称  
//layerIndex为指定图层索引号  
selRlt = viewport.LayerAttributes.GetFieldName  
    (layerIndex, out fieldName);
```

## 6.12 MapDisplayBound 属性

**public MAPGISIMS.EnterpriseLib.OpCRect MapDisplayBound**

说明

获取或设置地图的显示范围

备注

该属性隶属于 CMapView 对象

示例

```
//向客户端传送地图当前显示范围  
rc = viewport.MapDisplayBound;
```

## 6.13 MapImageName 属性

**Public string MapImageName**

说明

获取 GIS 服务器端生成的地图图像名

备注

该属性隶属于 CMapView 对象

## 6.14 MapObj 属性

**public MAPGISIMS.EnterpriseLib.CWebMap MapObj**

说明

地图文档类

备注

该属性隶属于 CMapView 对象

示例

```
//得到地图文档中图层的个数  
sLayerNum = viewport.MapObj.GetLayerNum(out layerNum);
```

## 6.15 MapOriginalBound 属性

**public MAPGISIMS.EnterpriseLib.OpCRect MapOriginalBound**

说明

读取地图范围

备注

该属性隶属于 CMapView 对象

示例

```
rc = viewport.MapOriginalBound;
```

## 6.16 OriginalFlag 属性

**Public int OriginalFlag**

说明

读取或设置地图图像是否还原显示

1: 还原显示

0: 不还原显示

备注

该属性隶属于 CMapView 对象

## 6.17 RecordsCountOfPage 属性

**public int RecordsCountOfPage**

说明

查询操作返回的每页的结果数目

备注

该属性隶属于 CMapView 对象

## 6.18 SaveLoadmapResultFlag 属性

**public bool SaveLoadmapResultFlag**

说明

是否保存地图加载结果

备注

该属性隶属于 CMapView 对象

### 6.19 SelectedLayerIndex 属性

**Public long SelectedLayerIndex**

说明

查询的图层索引

备注

该属性隶属于 CMapView 对象

### 6.20 SelectResultPageCount 属性

**Public long SelecteResultPageCount**

说明

获取查询结果总页数

备注

该属性隶属于 CMapView 对象

### 6.21 TransparentFlag 属性

**Public int TransparentFlag**

说明

读取或设置地图图像是否透明显示

0: 不可透明显示

1: 可以透明显示

备注

该属性隶属于 CMapView 对象

### 6.22 ZipFileName 属性

**Public string ViewPort.ZipFileName**

说明

获取压缩文件名称

备注

该属性隶属于 CMapView 对象

### 6.23 SpatialAnalyseRltFname 属性

**public string SpacialAnalyseRltFName**

说明

最后一次空间分析结果要素类名称

备注

该属性隶属于 CMapView 对象

### 6.24 SpatialAnalyseRltHDFName 属性

**public string SpacialAnalyseRltHDFName**

**说明**

最后一次空间分析结果 hdf 文件名称

**备注**

该属性隶属于 CMapView 对象

## 6.25 SpatialOpRltDbConn 属性

**public string SpatialOpRltDbConn**

**说明**

空间分析结果数据库连接字符串

**备注**

该属性隶属于 CMapView 对象

## 6.26 SpatialOpRltDbType 属性

**public MAPGISIMS.EnterpriseLib.SpatialRltDbTypeEnum SpatialOpRltDbType**

**说明**

空间分析结果数据库类型

**备注**

该属性隶属于 CMapView 对象

## 方法

### 6.27 AddLayer 方法

**Public MAPGISIMS.EnterpriseLib.OpCUniformResult AddLayer(MAPGISIMS.EnterpriseLib.OpCAddLayerInfo addLayerInfo)**

**说明**

向地图文档中动态添加图层。

**参数**

addLayerInfo: 要添加的图层信息

**返回值**

操作结果

**备注**

该方法隶属于 CMapView 对象。

### 6.28 AppendOneArc 方法

**public bool AppendOneArc(MAPGISIMS.EnterpriseLib.OpCAppendAArcLine theArc)**

**说明**

向地图文档中添加线要素。

**参数**

theArc: 线要素

**返回值**

成功—true, 失败—false

## 备注

该方法隶属于 CMapView 对象。

## 示例

```
//声明添加线函数输入参数对象
OpCAppendAArcLine theArc = new OpCAppendAArcLine ();

//得到输入参数对象的点对象,注:线由点组成,要把点集数组对象赋给一个线对象
OpCLine LinObj = new OpCLine (); //声明一个线对象
string PointOfPolyline = Request.QueryString["points"];

char[] splitter = new char[] { ',', ';' };
string[] pointsCoor = PointOfPolyline.Split(splitter);
OpCPoint2D[] points2D = new OpCPoint2D[pointsCoor.Length / 2];
for (int i = 0; i < points2D.Length; i++)
{
    points2D[i].X = Convert.ToDouble(pointsCoor[2 * i]);
    points2D[i].Y = Convert.ToDouble(pointsCoor[2 * i + 1]);
}
LinObj.PointNumber = (ushort)points2D.Length; //为线对象赋值
LinObj.ThePoints = points2D;

//线要添加到的图层索引号
theArc.LayerIndex = Convert.ToInt64(Session["strLayerNo"]);
//要添加的线属性字段长度
theArc.PropertyNumber = (ushort)this.Mapgis_AttEdit1.FieldValueArray.Length;
//要添加的点属性字段信息
theArc.PropertyValues = this.Mapgis_AttEdit1.FieldValueArray;
//要添加的线的逻辑坐标集合
theArc.TheArc = LinObj;

//2006-10-31添加:XPY
theArc.LineType = 1822;
theArc.PenWidth = 2.0f;
theArc.LineColor = 277;
theArc.LineRateX = 1.0f;
theArc.LineRateY = 1.0f;
theArc.MapID = viewport.MapObj.loadMapRlt.MapID;
try
{
    AppendFlg = viewport.AppendOneArc(theArc);
}
catch
{
    AppendFlg = false;
}
}
```

## 6.29 AppendOnePoint 方法

**public bool AppendOnePoint(MAPGISIMS.EnterpriseLib.OpCAppendAPoint thePoint)**

### 说明

向地图文档中添加点要素。

### 参数

thePoint: 点要素对象

### 返回值

成功—true, 失败—false

### 备注

该方法隶属于 CMapView 对象。

### 示例

```

//接收AppendElimDialog客户端传来的鼠标点击点的逻辑坐标
OpCPoint2D dotLog = new OpCPoint2D();

//为添加点函数输入参数赋值
OpCAppendAPoint thePoint = new OpCAppendAPoint();
//点要添加到的图层索引号
thePoint.LayerIndex = Convert.ToInt64(Session["strLayerNo"]);
//要添加的点属性字段长度
thePoint.PropertyNumber = (ushort)this.Mapgis_AttEdit1.FieldValueArray.Length;
//要添加的点属性字段信息
thePoint.PropertyValues = this.Mapgis_AttEdit1.FieldValueArray;
//要添加的点的逻辑坐标
dotLog.X = Convert.ToDouble(Request.QueryString["LogicX"]);
dotLog.Y = Convert.ToDouble(Request.QueryString["LogicY"]);
thePoint.ThePoint = dotLog;

thePoint.PicType = 7;
thePoint.PicHeight = 20.0f;
thePoint.PicWidth = 20.0f;
thePoint.PicColor = 11;
thePoint.MapID = viewport.MapObj.loadMapRlt.MapID;
//添加点函数
try
{
    AppendFlg = viewport.AppendOnePoint(thePoint);
}
catch(Exception ex)
{
    string str= ex.Message;
    AppendFlg = false;
}
}

```

### 6.30 AppendOneRegion 方法

**public bool AppendOneRegion(MAPGISIMS.EnterpriseLib.OpCAppendARegion theRegion**

**说明**

向地图文档中添加区要素。

**参数**

theRegion: 区要素

**返回值**

成功—true, 失败—false

**备注**

该方法隶属于 CMapView 对象。

**示例**

```

OpCAppendRegion theRegion = new OpCAppendRegion();
OpCRegion reg2D = new OpCRegion();
OpCLine[] line2D = new OpCLine[1];
line2D[0] = new OpCLine();

string PointOfPolygon = Request.QueryString["points"];

char[] splitterPolygon = new char[] { ',', ' ', ' ' };
string[] pointsCoorPolygon = PointOfPolygon.Split(splitterPolygon);
OpCPoint2D[] points2DPolygon = new OpCPoint2D[pointsCoorPolygon.Length / 2];
for (int i = 0; i < points2DPolygon.Length; i++)
{
    points2DPolygon[i].X = Convert.ToDouble(pointsCoorPolygon[2 * i]);
    points2DPolygon[i].Y = Convert.ToDouble(pointsCoorPolygon[2 * i + 1]);
}

line2D[0].ThePoints = points2DPolygon;
line2D[0].PointNumber = (ushort)points2DPolygon.Length;

reg2D.TheLines = line2D;
reg2D.LineNumber = 1;

theRegion.LayerIndex = Convert.ToInt64(Session["strLayerNo"]);
theRegion.PropertyValues = this.Mappgis_AttEdit1.FieldValueArray;
theRegion.PropertyNumber = (ushort)this.Mappgis_AttEdit1.FieldValueArray.Length;
theRegion.TheRegion = reg2D;

theRegion.RegioniPic = 1443;
theRegion.RegionColor = 411;
theRegion.PicHeight = 5.0f;
theRegion.PicWidth = 5.0f;
theRegion.PicColor = 6;

theRegion.MapID = viewport.MapObj.loadMapRlt.MapID;

try
{
    AppendFlg = viewport.AppendOneRegion(theRegion);
}
catch
{
    AppendFlg = false;
}

```

### 6.31 CheckNullObj 方法

**public static void CheckNullObj(object paraObj)**

说明

检查一个对象是否为 null，是 null 抛出异常。

参数

paraObj: 要检查的对象

备注

该方法隶属于 CMapView 对象，对象为空的时候抛出 System.ArgumentNullException 异常。

### 6.32 CheckSocketConnetion 方法

**Public bool CheckSocketConnetion**

说明

Check conn。

返回值

布尔值

备注

该方法隶属于 CMapView 对象。

### 6.33 CMapView 方法

#### **Public CMapView()**

说明

地图视图类的构造函数。

备注

该方法隶属于 CMapView 对象。

示例

```
CMapView viewport;  
  
viewport = new CMapView();
```

### 6.34 CreateImageName 方法

#### **public string CreateImageName(string SessionID, MAPGISIMS.EnterpriseLib. ImageFormatEnum fileType)**

说明

生成唯一的图像的文件名

参数

SessionID: 初始化文件名的字符串。如 Session.SessionID.可为 null

fileType: ImageFormatEnum 值之一

返回值

文件名。如: "32432432423432007070132432.gif"

备注

该方法隶属于 CMapView 对象。

示例

```
viewport.ImageFormat = ImageFormatEnum.gif;  
string imgFileName = viewport.CreateImageName  
(Session.SessionID, viewport.ImageFormat);  
byte[] data = viewport.GetBinaryImage(imgFileName);  
Response.BinaryWrite(data);
```

### 6.35 DeleteLayer 方法

#### **public MAPGISIMS.EnterpriseLib.OpCUniformResult DeleteLayer(MAPGISIMS. EnterpriseLib.OpCGeomInfo geomInfo)**

说明

删除图层

参数

geomInfo: 图层信息

返回值

操作结果

**备注**

该方法隶属于 CMapView 对象。

**6.36 DeleteOneFeature 方法**

**public bool DeleteOneFeature(MAPGISIMS.EnterpriseLib.OpCDeleteFeature theFeature)**

**说明**

删除要素

**参数**

theFeature: 删除要素对象

**返回值**

成功—true, 失败—false

**备注**

该方法隶属于 CMapView 对象。

**示例**

```
OpCDeleteFeature theFeature = new OpCDeleteFeature();

bool delFeature = false;
theFeature.FeatureID = featureID;
theFeature.LayerIndex = layerIndex;
theFeature.DelGeometry = 1;

theFeature.MapID = viewport.MapObj.LoadMapRlt.MapID;
try
{
    delFeature = viewport.DeleteOneFeature(theFeature);
}
}
```

**6.37 GetBinaryImage 方法**

**Public byte[] GetBinaryImage(string imgFileName);**

**说明**

获取图片, 返回图片字节流。

**参数**

imgFileName: 图片名称

**返回值**

字节流

**备注**

该方法隶属于 CMapView 对象。

**示例**

```
viewport.ImageFormat = ImageFormatEnum.gif;
string imgFileName = viewport.CreateImageName
(Session.SessionID, viewport.ImageFormat);
byte[] data = viewport.GetBinaryImage(imgFileName);
Response.BinaryWrite(data);
```

**6.38 GetEagleImage 方法**

**public short GetEagleImage(ushort imageWidth, ushort imageHeight, double**

**zoomRateIn, string imageFileName, out double xOrg, out double yOrg, out double zoomRateOut)**

**说明**

获取鹰眼图像。

**参数**

imageWidth: 鹰眼图像宽

imageHeight: 鹰眼图像高

zoomRateIn: 鹰眼图像放大比例

imageFileName: 鹰眼图像文件名

xOrg: 鹰眼显示原点 X 坐标

yOrg: 鹰眼显示原点 Y 坐标

zoomRateOut: 鹰眼显示比例

**返回值**

成功—1, 失败— -1

**备注**

该方法隶属于 CMapView 对象。

### 6.39 GetFeatureGeometry 方法

**public MAPGISIMS.EnterpriseLib.OpCFeatureGeometry GetFeatureGeometry (MAPGISIMS.EnterpriseLib.OpCGetFeatureInformation featureToGet)**

**说明**

获取要素几何信息

**参数**

featureToGet: 获取要素信息类

**返回值**

OpCFeatureGeometry 类

**备注**

该方法隶属于 CMapView 对象。

**示例**

```
//获取查询结果的几何信息
OpCFeatureGeometry featureGeometry;
OpCGetFeatureInformation featureToGet = new OpCGetFeatureInformation();
featureToGet.FeatureID = featureID;
featureToGet.LayerIndex = LayerIndex;
featureToGet.InformationType = 0;
try
{
    featureGeometry = viewport.GetFeatureGeometry(featureToGet);
    if (featureGeometry == null)
    {
        throw new Exception("alert('获取结果失败! 请重新查询!')");
    }
}
}
```

### 6.40 GetFeaturelayerAttributeStructure 方法

**public MAPGISIMS.EnterpriseLib.OpCLayerAttStructure GetFeaturelayerAttributeStructure(long layerIndex)**

**说明**

得到指定要素图层的属性结构

**参数**

layerIndex: 图层索引

**返回值**

成功-true, 失败-false

**备注**

该方法隶属于 CMapView 对象。

**6.41 GetFeaturelayerAttributeStructure 方法**

```
public MAPGISIMS.EnterpriseLib.OpCLayerAttStructure  
GetFeaturelayerAttributeStructure(long layerIndex, long MapID)
```

**说明**

得到指定要素图层的属性结构

**参数**

layerIndex: 图层索引

MapID: 地图 ID

**返回值**

成功-true, 失败-false

**备注**

该方法隶属于 CMapView 对象。

**示例**

```
OpCLayerAttStructure regLayAttStr;  
long MapID = viewport.MapObj.LoadMapRlt.MapID;  
regLayAttStr = viewport.GetFeaturelayerAttributeStructure(  
    Convert.ToInt64(Session["strConditionLayerNo"]), MapID);  
int fldNum = regLayAttStr.FieldNumber;  
string[] fldName = regLayAttStr.FieldNames;  
byte[] fldType = regLayAttStr.FieldTypes;  
this.Mapgis_ConditionGenerator1.FieldNameArray = fldName;
```

**6.42 GetFeatureStyleInfo 方法**

```
public MAPGISIMS.EnterpriseLib.OpCFeatureStyleInfo GetFeatureStyleInfo  
(MAPGISIMS.EnterpriseLib.OpCGetFeatureStyleInfo getPatter)
```

**说明**

获取要素样式信息, 如颜色, 高宽等

**参数**

getPatter: 获取要素样式信息类

**返回值**

OpCFeatureStyleInfo 类。

**备注**

该方法隶属于 CMapView 对象。

**6.43 GetGDBInfo 方法**

```
public MAPGISIMS.EnterpriseLib.OpCUniformResult GetGDBInfo
```

**(MAPGISIMS.EnterpriseLib.OpCGetGDBInfo GDBInfo)****说明**

获取 GDB 数据源名称。

**参数**

GDBInfo: GDB 数据源信息

**返回值**

OpCUniformResult 类

**备注**

该方法隶属于 CMapView 对象。

**示例**

```
OpCGetGDBInfo GDBInfo = new OpCGetGDBInfo ();
OpCUniformResult uniformRlt = null;
string[] GDBDataSourceName;
GDBInfo.MapID = viewport.MapObj.loadMapRlt.MapID;
try
{
    uniformRlt = viewport.GetGDBInfo(GDBInfo);
}
```

**6.44 GetGDBLayerInfo 方法****public MAPGISIMS.EnterpriseLib.OpCUniformResult GetGDBLayerInfo  
(MAPGISIMS.EnterpriseLib.OpCGetGDBLayerInfo GDBLayerInfo)****说明**

获取 GDB 信息。

**参数**

GDBLayerInfo: GDB 信息

**返回值**

OpCUniformResult 类

**备注**

该方法隶属于 CMapView 对象。

**示例**

```
OpCGetGDBLayerInfo GDBLayerInfo = new OpCGetGDBLayerInfo ();
OpCUniformResult uniformRlt = new OpCUniformResult ();
GDBLayerInfo.GDBName = gdbName;
GDBLayerInfo.MapID = viewport.MapObj.loadMapRlt.MapID;
GDBLayerInfo.DataSource = gdbServerName;
try
{
    uniformRlt = viewport.GetGDBLayerInfo(GDBLayerInfo);
}
```

**6.45 GetGDBName 方法****public MAPGISIMS.EnterpriseLib.OpCUniformResult GetGDBName(MAPGISIMS.  
EnterpriseLib.OpCGetGDBName GDBName)****说明**

获取 GDB 中数据库名称。

**参数**

GDBName: 数据库名称

返回值

OpCUniformResult 类

备注

该方法隶属于 CMapView 对象。

示例

```
OpCGetGDBName GDBName = new OpCGetGDBName();
OpCUniformResult uniformRlt = new OpCUniformResult();
string[] GDBNameStr;
GDBName.MapID = viewport.MapObj.loadMapRlt.MapID;
GDBName.DataSource = GDBDataSourceName;
GDBName.userName = user;
GDBName.password = pwds;
try
{
    uniformRlt = viewport.GetGDBName(GDBName);
}
```

## 6.46 GetGeomInfo 方法

**public MAPGISIMS.EnterpriseLib.OpCUniformResult GetGeomInfo(MAPGISIMS.EnterpriseLib.OpCGeomInfo geomInfo)**

说明

根据图层索引号得到图层信息。

参数

geomInfo:

返回值

图层信息

备注

该方法隶属于 CMapView 对象。

示例

```
OpCGeomInfo geomInfo = new OpCGeomInfo();
OpCUniformResult uniformRlt = new OpCUniformResult();
op = Request.QueryString["op"];
geomInfo.LayerIndex = selectlayerIndex;
geomInfo.MapID = viewport.MapObj.loadMapRlt.MapID;
geomInfo.FLAG = 0;
try
{
    uniformRlt = viewport.GetGeomInfo(geomInfo); //获取图层类型(点、线、区)
}
```

## 6.47 GetImageFromRar 方法

**public static void GetImageFromRar(string exePath, string rarPath, string destImagePath)**

说明

从压缩包获取图像文件到指定目录

参数

exePath: 解压程序所在路径 如: d: \rar.exe

rarPath: 压缩文件所在路径 如: d: \ims.rar

destImagePath: 释放文件目标目录 如 d: \

返回值

图层信息

**备注**

该方法隶属于 CMapView 对象。

## 6.48 GetMapDocList 方法

**public string[] GetMapDocList()**

**说明**

从 Gis 服务器获取地图文档列表

**返回值**

地图文档列表

**备注**

该方法隶属于 CMapView 对象。

## 6.49 GetMapImage 方法

**Public** short GetMapImage(string imgFileName);

**说明**

生成地图图像

**参数**

imgFileName: 图像文件名称

**返回值**

是否成功

**备注**

该方法隶属于 CMapView 对象。

## 6.50 GetMapLayerObj 方法

**public int getMapLayerObj(string strLayerNo, out MAPGISIMS.EnterpriseLib.**

**OpCLayerInformation mapLayerObj)**

**说明**

获取指定图层索引的图层信息

**参数**

strLayerNo: 图层索引, 子图层: "0,0"

mapLayerObj: 得到的图层信息

**返回值**

是否成功

**备注**

该方法隶属于 CMapView 对象。

**示例**

```
int fieldNum = layerSelectRlt[i].FieldNumber;
int selectResultCount = layerSelectRlt[i].Count;
string layerName;
int sRlt =viewport.getMapLayerObj(layerSelectRlt[i].LayerIndex.ToString(), out playerInfo);
if(sRlt<1)
Response.Write("<script language=javascript>alert('获取图层信息失败')</script>");
layerName = playerInfo.LayerName;
```

## 6.51 GetMapObj 方法

**public MAPGISIMS.EnterpriseLib.CWebMap getMapObj()**

说明

获取地图文档

参数

无

返回值

地图文档

备注

该方法隶属于 CMapView 对象。

## 6.52 GetNewMapInfo 方法

**Public short GetNewMapInfo(long mapID);**

说明

重新获取地图文档信息。

参数

mapID: 地图信息

返回值

大于 0 成功, 小于 0 失败

备注

该方法隶属于 CMapView 对象。

## 6.53 GetOnePageResult 方法

**public void GetOnePageResult(int nPage, MAPGISIMS.EnterpriseLib.  
OpCGetOnePage page)**

说明

获取查询结果的指定页。

参数

nPage: 页码

page: 查询对象

备注

该方法隶属于 CMapView 对象。

示例

```
page = (OpCGetOnePage)Session["page"];  
viewport.GetOnePageResult(currentPage, page);
```

## 6.54 getloadflag 方法

**Public bool getreloadflag()**

说明

获取重新连接标志

参数

无

返回值

布尔值

备注

该方法隶属于 CMapView 对象。

## 6.55 GetSpatialAnalysisResult 方法

**public void GetSpatialAnalysisResult(MAPGISIMS.EnterpriseLib.  
OpCSpatialOperResult result)**

说明

得到空间分析的结果文件

参数

result: 空间分析结果

备注

该方法隶属于 CMapView 对象。

示例

```
//声明叠加分析对象
OpCSpatialOverlay overlay = new OpCSpatialOverlay();
OpCSpatialOperResult SpatialOperRlt = null;

//为叠加分析的两个图层赋值
overlay.LayerIndex1 = overlay.LayerIndex2 = 0;
overlay.GDBDataSourceName1 = "MapgisLocal";
//overlay.GDBName1 = "ChinaTestNew.HDF";
overlay.GDBName1 = "WuhanTest.HDF";
overlay.GDBDataSourceName2 = "";
overlay.FeatureIDSLst2 = "";
overlay.FeatureIDSLst1 = layerName1 + "," + layerName2;
//叠加分析的类型，如并、差、交等
overlay.OverlayType = Convert.ToInt16(operation);
//叠加分析后生成的文件名
overlay.OverlayGDBName = fclsName;
overlay.MapID = viewport.MapObj.loadMapRlt.MapID;
//viewport.MapObj.loadMapRlt;
//叠加分析的返回结果对象
try
{
    SpatialOperRlt = viewport.SpatialAnalysisOverlay(overlay);
}
catch
{
    SpatialOperRlt = null;
}

if (SpatialOperRlt == null)
{
    flg.Value = "-1";
    Response.Write("<script language=javascript>alert('叠加分析失败!')</script>");
    return;
}
else
{
    //得到叠加分析后结果对象
    viewport.GetSpatialAnalysisResult(SpatialOperRlt);
    //通过结果对象取叠加分析最后文件所在URL地址
    fileName.Value = viewport.ZipFileName;
    flg.Value = "1";
}
}
```

## 6.56 InitWebServer 方法

**public static void InitWebServer(bool needGisMgr, string[] ipList, string[] portList, string schemaFilePath)**

说明

重载的初始化函数

参数

needGisMgr: 是否需要GIS服务器管理器

ipList: 服务器IP列表

portList: 服务器端口列表

schemaFilePath: 样式表文件目录

备注

带参数的初始化函数。该方法隶属于 CMapView 对象。

## 6.57 InitWebServer 方法

**Public static void InitWebServer(string imageFolder);**

说明

初始化 Web 服务器

参数

imageFolder: 生成图像在 Web 服务器端的存放目录

备注

该方法隶属于 CMapView 对象。

## 6.58 LoadMap 方法

**Public int LoadMap(string mapDocName);**

说明

加载地图文档。

参数

mapDocName: 地图文档名称

备注

该方法隶属于 CMapView 对象。

示例

```
//获取地图显示高宽
viewport.ImageWidth = width;
viewport.ImageHeight = height;
try
{
    //加载地图文档
    string mapName = Session["MapName"].ToString();
    if (mapName == null)
    {
        Response.Write("地图文档不能为空。");
        Response.End();
    }
    viewport.LoadMap(mapName);
    Session["viewport"] = viewport;
}
```

## 6.59 LogicalToWindow 方法

**public short LogicalToWindow(MAPGISIMS.EnterpriseLib.OpCPoint2D logicalPoint, ref MAPGISIMS.EnterpriseLib.OpCPoint2D windowPoint)**

说明

逻辑坐标转换窗口坐标

参数

logicalPoint: 逻辑坐标

windowPoint: 窗口坐标

返回值

大于 0 成功, 小于 0 失败

备注

该方法隶属于 CMapView 对象。

## 6.60 Move 方法

**public short Move(int xFrmPos, int yFrmPos, int xToPos, int yToPos)**

说明

地图的移动操作。

参数

xFrmPos: 移动起点X窗口坐标

yFrmPos: 移动起点Y窗口坐标

xToPos: 移动终点X窗口坐标

yToPos: 移动终点 Y 窗口坐标

备注

该方法隶属于 CMapView 对象。

## 6.61 NetAnalysis 方法

**public MAPGISIMS.EnterpriseLib.OpCNetAnalysisResult NetAnalysis(MAPGISIMS.EnterpriseLib.OpCBasicNetAnalysis net)**

说明

网络分析

参数

net: 网络分析对象

返回值

网络分析结果

备注

该方法隶属于 CMapView 对象。

示例

```

OpCBasicNetAnalysis net = new OpCBasicNetAnalysis();
OpCNetAnalysisResult netRlt = null;
net.NodeNumber = points2D.Length;
net.barNum=barPntNum;
net.barPoints = barpoints;
net.Nodes = points2D;
net.LayerIndex = Convert.ToInt64(Session["strLayerNo"]);
net.Radius = radius;
net.MapID = viewport.MapObj.loadMapRlt.MapID;
try
{
    netRlt = viewport.NetAnalysis(net);
}

```

## 6.62 PrjTran 方法

**public MAPGISIMS.EnterpriseLib.OpCPrjTranRlt PrjTran(MAPGISIMS.EnterpriseLib.OpCPrjTran prj)**

说明

投影变换

参数:

prj: 投影类

返回值

投影结果

备注

该方法隶属于 CMapView 对象。

## 6.63 Restore 方法

**Public short Restore()**

说明

复位操作

返回值

成功—1, 失败—-1

备注

该方法隶属于 CMapView 对象。

## 6.64 SelectByCircle 方法

**public short SelectByCircle(MAPGISIMS.EnterpriseLib.OpCCircleSelectClass selectCircle)**

说明

画圆查询

参数

selectCircle: 画圆查询对象

返回值

成功—1, 失败—-1

备注

该方法隶属于 CMapView 对象。

## 示例

```

xmin = Convert.ToDouble(Request.QueryString["LogicX"]);
ymin = Convert.ToDouble(Request.QueryString["LogicY"]);
radius = Convert.ToDouble(Request.QueryString["Radius"]);
if (radius == 0.0)
{
    Response.Write("<script language=javascript>"+
        "alert(' 画圆操作失败：半径为0。请重新操作！ ')</script>");
    return;
}
p2d = new OpCPoint2D(xmin, ymin);
OpCCircleSelectClass circleSelectObj = new OpCCircleSelectClass();
circleSelectObj.Center = p2d;
circleSelectObj.Radius = radius;
circleSelectObj.MapID = viewport.MapObj.loadMapRlt.MapID;
if (Session["singleLayerQuery"] == null)
    circleSelectObj.LayerIndex = -1;
else
{
    if (Session["strLayerNo"] == null)
    {
        Response.Write("<script language=javascript>alert(' 请点击正确的图层名并提交更改！ ')</script>");
        return;
    }
    circleSelectObj.LayerIndex = Convert.ToInt64(Session["strLayerNo"]);
}
circleSelectObj.PageIndex = 0;
try
{
    viewport.SelectByCircle(circleSelectObj);
    page = new OpCGetOnePage();
    QueryTypeEnum querytype = QueryTypeEnum.SelectByCircle;
    page.CircleSelect = circleSelectObj;
    page.SelectType = querytype;
    Session["page"] = page;
    Session["viewport"] = viewport;
    ShowAttView();
}

```

## 6.65 SelectByCondition 方法

**public short SelectByCondition(MAPGISIMS.EnterpriseLib.OpCAttSelectStru selectString)**

说明

条件查询

参数

selectString: 条件查询对象

返回值

成功—1, 失败—-1

备注

该方法隶属于 CMapView 对象。

## 6.66 SelectByCondition 方法

**public short SelectByCondition(MAPGISIMS.EnterpriseLib.OpCCircleWithAttSelectStru circleAndAtt)**

说明

圆条件查询

参数

circleAndAtt: 圆条件查询对象

返回值

成功—1，失败— -1

#### 备注

该方法隶属于 CMapView 对象。

### 6.67 SelectByCondition 方法

**public short SelectByCondition(MAPGISIMS.EnterpriseLib.  
OpCPointWithAttSelectStru pntAndAtt)**

#### 说明

点击条件查询

#### 参数

pntAndAtt: 点击条件查询对象

#### 返回值

成功—1，失败— -1

#### 备注

该方法隶属于 CMapView 对象。

### 6.68 SelectByCondition 方法

**public short SelectByCondition(MAPGISIMS.EnterpriseLib.  
OpCPolygonWithAttSelectStru polygonAndAtt)**

#### 说明

多边形条件查询

#### 参数

polygonAndAtt: 多边形条件查询对象

#### 返回值

成功—1，失败— -1

#### 备注

该方法隶属于 CMapView 对象。

### 6.69 SelectByCondition 方法

**public short SelectByCondition(MAPGISIMS.EnterpriseLib.  
OpCRectWithAttSelectStru rectAndAtt)**

#### 说明

矩形条件查询

#### 参数

rectAndAtt: 矩形条件查询对象

#### 返回值

成功—1，失败— -1

#### 备注

该方法隶属于 CMapView 对象。

#### 示例

```

xmin = Convert.ToDouble(Request.QueryString["LogicMinX"]);
ymin = Convert.ToDouble(Request.QueryString["LogicMinY"]);
xmax = Convert.ToDouble(Request.QueryString["LogicMaxX"]);
ymax = Convert.ToDouble(Request.QueryString["LogicMaxY"]);
condition = Request.QueryString["condition"];
if (condition == "")
{
    Response.Write("<script language=javascript>"+
        "alert('请输入查询条件!');window.close();</script>");
    return;
}
rc = new OpCRect();
rc.XMin = xmin;
rc.YMin = ymin;
rc.XMax = xmax;
rc.YMax = ymax;
OpCRectWithAttSelectStru rectAndAtt = new OpCRectWithAttSelectStru();
rectAndAtt.MapID = viewport.MapObj.loadMapRlt.MapID;
rectAndAtt.RectangleArea = rc;
rectAndAtt.Question = condition;
if (Session["singleLayerQuery"] == null)
    rectAndAtt.LayerIndex = -1;
else
{
    if (Session["strLayerNo"] == null)
    {
        Response.Write("<script language=javascript>alert('请点击正确的图层名并提交更改!')</script>");
        return;
    }
    rectAndAtt.LayerIndex = Convert.ToInt64(Session["strLayerNo"]);
}
try
{
    viewport.SelectByCondition(rectAndAtt);
    page = new OpCGetOnePage();
    QueryTypeEnum querytype = QueryTypeEnum.SelectByRectCon;
    page.RectWithAttSelect = rectAndAtt;
    page.SelectType = querytype;
    Session["page"] = page;
    Session["viewport"] = viewport;
    ShowAttView();
}

```

## 6.70 SelectByHitting 方法

**public short SelectByHitting(MAPGISIMS.EnterpriseLib.OpCPointSelectClass selectPoint);**

说明

点击查询

参数

selectPoint: 点击查询对象

返回值

成功—1, 失败— -1

备注

该方法隶属于 CMapView 对象。

示例

```

//点击点的坐标
xmin = Convert.ToDouble(Request.QueryString["LogicX"]);
ymin = Convert.ToDouble(Request.QueryString["LogicY"]);
//误差半径
distance = Convert.ToDouble(Request.QueryString["BufX"]);
p2d = new OpCPoint2D(xmin, ymin);
OpCPointSelectClass pointSelectObj = new OpCPointSelectClass();
pointSelectObj.Dot = p2d;
pointSelectObj.Distance = distance;
pointSelectObj.MapID = viewport.MapObj.loadMapRlt.MapID;
if (Session["singleLayerQuery"] == null)
    pointSelectObj.LayerIndex = -1;
else
{
    if (Session["strLayerNo"] == null)
    {
        Response.Write("<script language=javascript>alert('请点击正确的图层名并提交更改!')</script>");
        return;
    }
    pointSelectObj.LayerIndex = Convert.ToInt64(Session["strLayerNo"]);
}
try
{
    viewport.SelectByHitting(pointSelectObj);
    //保存上次查询对象
    page = new OpCGetOnePage();
    QueryTypeEnum querytype = QueryTypeEnum.SelectByHitting;
    page.PntSelect = pointSelectObj;
    page.SelectType = querytype;
    Session["page"] = page;
    Session["viewport"] = viewport;
    ShowAttView();
}

```

## 6.71 SelectByLine 方法

**public short SelectByLine(MAPGISIMS.EnterpriseLib.OpCLineSelectClass selectLine)**

说明

线查询

参数

selectLine: 线查询对象

返回值

成功—1, 失败— -1

备注

该方法隶属于 CMapView 对象。

示例

```

string pointsOfPolyline = Request.QueryString["Points"];
splitor = new char[] { ',', ' ' };
pointsCoor = pointsOfPolyline.Split(splitor);
points2D = new OpCPoint2D[pointsCoor.Length / 2];
for (i = 0; i < points2D.Length; i++)
{
    points2D[i].X = Convert.ToDouble(pointsCoor[2 * i]);
    points2D[i].Y = Convert.ToDouble(pointsCoor[2 * i + 1]);
}
OpCLine arcLine2D = new OpCLine();
arcLine2D.ThePoints = points2D;
arcLine2D.PointNumber = points2D.Length;
OpCLineSelectClass lineSelectObj = new OpCLineSelectClass();
lineSelectObj.SelectLine = arcLine2D;
if (Session["singleLayerQuery"] == null)
    lineSelectObj.LayerIndex = -1;
else
{
    if (Session["strLayerNo"] == null)
    {
        Response.Write("<script language=javascript>alert('请点击正确的图层名并提交更改!')</script>");
        return;
    }
    lineSelectObj.LayerIndex = Convert.ToInt64(Session["strLayerNo"]);
}
try
{
    viewport.SelectByLine(lineSelectObj);
    page = new OpCGetOnePage();
    QueryTypeEnum querytype = QueryTypeEnum.SelectByLine;
    page.LineSelect = lineSelectObj;
    page.SelectType = querytype;
    Session["page"] = page;
    Session["viewport"] = viewport;
    ShowAttView();
}
}

```

## 6.72 SelectByPolygon 方法

**public short SelectByPolygon(MAPGISIMS.EnterpriseLib.OpCPolygonSelectClass selectPolygon)**

说明

多边形查询

参数

selectPolygon: 多边形查询对象

返回值

成功—1, 失败— -1

备注

该方法隶属于 CMapView 对象。

示例

```

string PointOfPolygon = Request.QueryString["Points"];

char[] splitter = new char[] { ',', ' ' };
string[] pointsCoor = PointOfPolygon.Split(splitter);
OpCPoint2D[] points2D = new OpCPoint2D[pointsCoor.Length / 2];
for (int i = 0; i < points2D.Length; i++)
{
    points2D[i].X = Convert.ToDouble(pointsCoor[2 * i]);
    points2D[i].Y = Convert.ToDouble(pointsCoor[2 * i + 1]);
}

OpCPolygonSelectClass polygonSelectObj = new OpCPolygonSelectClass();
polygonSelectObj.Polygon = points2D;
polygonSelectObj.PointNumber = (ushort)points2D.Length;
polygonSelectObj.MapID = viewport.MapObj.loadMapRlt.MapID;
polygonSelectObj.LayerIndex = -1;
try
{
    //调用点击查询函数,输入参数为上面赋值的拉框查询对象
    viewport.SelectByPolygon(polygonSelectObj);
    Session["viewport"] = viewport;
    ShowAttView();
}

```

### 6.7.3 SelectByRect 方法

**public short SelectByRect(MAPGISIMS.EnterpriseLib.OpCRectangleSelectClass selectRectangle)**

说明

拉框查询

参数

selectRectangle: 拉框查询对象

返回值

成功—1, 失败—-1

备注

该方法隶属于 CMapView 对象。

示例

```

OpCRect rc = new OpCRect();
rc.XMin = xmin;
rc.YMin = ymin;
rc.XMax = xmax;
rc.YMax = ymax;

//声明拉框查询对象
OpCRectangleSelectClass rectSelectObj = new OpCRectangleSelectClass();
rectSelectObj.RectangleArea = rc;
rectSelectObj.MapID = viewport.MapObj.loadMapRlt.MapID;
rectSelectObj.LayerIndex = -1;

try
{
    //调用点击查询函数,输入参数为上面赋值的拉框查询对象
    viewport.SelectByRect(rectSelectObj);
    Session["viewport"] = viewport;
    ShowAttView();
}

```

## 6.74 SetImageDisplayParam 方法

**public void SetImageDisplayParam(bool originalFlag, bool showScale, bool showCoordinate, bool followScale, bool bufferShow, bool dynamicNote, MAPGISIMS.EnterpriseLib.ImageQuality imageQuality)**

说明

设置图像生成参数

参数

originalFlag: 是否还原显示

showScale: 是否标尺

showCoordinate: 是否显示坐标点

followScale: 是否跟随比例显示

bufferShow: 是否缓冲显示

imageQuality: 图像质量 1低,2中,3高

dynamicNote: 是否显示动态标注

备注

该方法隶属于 CMapView 对象。

示例

```
bool originalFlag = this.origin.Checked;
bool showScale = this.scale.Checked;
bool showCoordinate = this.Coordinate.Checked;
bool followScale = this.follow.Checked;
bool bufferShow = this.buff.Checked;
bool dynamicNote = this.dynamic.Checked;
ImageQuality imageQuality=ImageQuality.middle;
switch(this.quality.SelectedIndex)
{
    case 0:
        imageQuality=ImageQuality.low;
        break;
    case 1:
        imageQuality=ImageQuality.middle;
        break;
    case 2:
        imageQuality=ImageQuality.high;
        break;
}
if (Session["viewport"] == null)
{
    Response.Write("<script language=javascript>alert('+
    ''地图文档装载失败或请求服务超时!');</script>");
    return;
}
viewport = (CMapView)Session["viewport"];
viewport.SetImageDisplayParam(originalFlag, showScale, showCoordinate,
    followScale, bufferShow, dynamicNote, imageQuality);
```

## 6.75 setreloadflag 方法

**Public void setreloadflag(bool reloadflag)**

说明

设置重新连接标志。

参数

reloadflag: 布尔值

**备注**

该方法隶属于 CMapView 对象。

**6.76 SpatialAnalysisBuffer 方法**

**public MAPGISIMS.EnterpriseLib.OpCSpatialOperResult SpatialAnalysisBuffer  
(MAPGISIMS.EnterpriseLib.OpCSpatialBuffer buffer)**

**说明**

缓冲区分析

**参数**

buffer: 缓冲区分析对象

**返回值**

缓冲区分析结果

**备注**

该方法隶属于 CMapView 对象。

**示例**

```
//声明缓冲区分析对象、返回值描述对象
OpCSpatialBuffer buffer = new OpCSpatialBuffer ();
Random rand = new Random ();
//缓冲区分析后生成文件的名称
buffer.FeatureClassName = "buffer" + Convert.ToString(rand.NextDouble () * 1000);
//要进行缓冲区分析的要素实体号
buffer.FeatureIDS = featureIDS;
//要进行缓冲区分析的要素个数
buffer.FeatureNumber = Convert.ToInt16(featureIDS.Length);
//缓冲区分析半径
buffer.Radius = radius;
//要进行缓冲区分析的图层索引号
buffer.LayerIndex = layerSelectRlt[selLayerIndex].LayerIndex;
buffer.MapID = viewport.MapObj.loadMapRlt.MapID;
//调用缓冲区分析函数，并通过返回值验证分析是否成功
OpCSpatialOperResult SpatialOperRlt = null;
string flg;
try
{
    SpatialOperRlt = viewport.SpatialAnalysisBuffer (buffer);
}
```

**6.77 SpatialAnalysisCircleClip 方法**

**public MAPGISIMS.EnterpriseLib.OpCSpatialOperResult  
SpatialAnalysisCircleClip(MAPGISIMS.EnterpriseLib.OpCSpatialCircleClip  
circleClip)**

**说明**

圆裁剪

**参数**

circleClip: 圆裁剪对象

**返回值**

圆裁剪结果

**备注**

该方法隶属于 CMapView 对象。

**示例**

```

//声明圆裁剪对象、返回值对象
OpCSpatialCircleClip circleClip = new OpCSpatialCircleClip();

double xmin = Convert.ToDouble(Request.QueryString["LogicX"]);
double ymin = Convert.ToDouble(Request.QueryString["LogicY"]);
double radius = Convert.ToDouble(Request.QueryString["Radius"]);
string layerIndexLst = (string)Session["layerIndexLst"];
if (Session["layerIndexLst"]==null||Session["layerIndexLst"] == "")
{
    Response.Write("<script language=javascript>alert('请选择要素图层!');window.close();</script>");
    return;
}
if (Session["viewport"] == null)
{
    Response.Write("地图文档装载失败或请求服务超时");
    return;
}
viewport = (CMapView)Session["viewport"];

OpCSpatialOperResult spatialOperRlt = null;
//fClassDescription description = null;
//接收客户端传来的圆心、半径及裁剪半径
OpCPoint2D dotCenter = new OpCPoint2D();
dotCenter.X = xmin;
dotCenter.Y = ymin;
circleClip.Center = dotCenter;
circleClip.Radius = radius;
circleClip.LayerIndexLst = layerIndexLst;
circleClip.MapID = viewport.MapObj.loadMapRlt.MapID;
//裁剪类型
Random rand = new Random();
circleClip.ClipType = 1;
circleClip.FeatureClassName = "CircleClip" + Convert.ToString(rand.NextDouble() * 1000);
try
{
    spatialOperRlt = viewport.SpatialAnalysisCircleClip(circleClip);
}
catch
{
    spatialOperRlt = null;
}

```

## 6.78 SpatialAnalysisOverlay 方法

**public MAPGISIMS.EnterpriseLib.OpCSpatialOperResult SpatialAnalysisOverlay  
(MAPGISIMS.EnterpriseLib.OpCSpatialOverlay overlay)**

说明

空间叠加分析

参数

Overlay: 叠加分析对象

返回值

叠加分析结果

备注

该方法隶属于 CMapView 对象。

示例

```

//声明叠加分析对象
OpCSpatialOverlay overlay = new OpCSpatialOverlay();
OpCSpatialOperResult SpatialOperRlt = null;

//为叠加分析的两个图层赋值
overlay.LayerIndex1 = overlay.LayerIndex2 = 0;
overlay.GDBDataSourceName1 = "MapgisLocal";
//overlay.GDBName1 = "ChinaTestNew.HDF";
overlay.GDBName1 = "WuhanTest.HDF";
overlay.GDBDataSourceName2 = "";
overlay.FeatureIDSLst2 = "";
overlay.FeatureIDSLst1 = layerName1 + "," + layerName2;
//叠加分析的类型，如并、差、交等
overlay.OverlayType = Convert.ToInt16(operation);
//叠加分析后生成的文件名
overlay.OverlayGDBName = fclsName;
overlay.MapID = viewport.MapObj.loadMapRlt.MapID;
//viewport.MapObj.loadMapRlt;
//叠加分析的返回结果对象
try
{
    SpatialOperRlt = viewport.SpatialAnalysisOverlay(overlay);
}
catch
{
    SpatialOperRlt = null;
}

```

## 6.79 SpatialAnalysisPolygonClip 方法

**public** **MAPGISIMS.EnterpriseLib.OpCSpatialOperResult**  
**SpatialAnalysisPolygonClip(MAPGISIMS.EnterpriseLib.OpCSpatialPolygonClip**  
**polygonClip)**

说明

多边形裁剪

参数

polygonClip: 多边形裁剪对象

返回值

多边形裁剪结果

备注

该方法隶属于 CMapView 对象。

示例

```

//定义多边形裁剪对象和返回值对象
OpCSpatialPolygonClip polygonClip = new OpCSpatialPolygonClip();
//要素类查询
if (Session["viewport"] == null)
{
    Response.Write("地图文档装载失败或请求服务超时");
    return;
}
viewport = (CMapView)Session["viewport"];
OpCSpatialOperResult SpatialOperRlt = null;
Random rand = new Random();
string layerIndexLst = (string)Session["layerIndexLst"];
if (Session["layerIndexLst"] == "")
{
    Response.Write("<script language=javascript>alert('请选择要素图层!');window.close();</script>");
    return;
}
string PointOfPolygon = Request.QueryString["Points"];

char[] splitter = new char[] { ',', ' ' };
string[] pointsCoor = PointOfPolygon.Split(splitter);
OpCPoint2D[] points2D = new OpCPoint2D[pointsCoor.Length / 2];
for (int i = 0; i < points2D.Length; i++)
{
    points2D[i].X = Convert.ToDouble(pointsCoor[2 * i]);
    points2D[i].Y = Convert.ToDouble(pointsCoor[2 * i + 1]);
}
polygonClip.PointNumber = (short) points2D.Length;
polygonClip.Points = points2D;
polygonClip.ClipType = 1;
polygonClip.LayerIndexLst = layerIndexLst;
polygonClip.MapID = viewport.MapObj.loadMapRlt.MapID;
polygonClip.FeatureClassName = "polygonClip" + Convert.ToString(rand.NextDouble() * 1000);
try
{
    SpatialOperRlt = viewport.SpatialAnalysisPolygonClip(polygonClip);
}
catch
{
    SpatialOperRlt = null;
}

```

## 6.80 SpatialOverlayByID 方法

**public MAPGISIMS.EnterpriseLib.OpCOverlayByIDPntSeq SpatialOverlayByID  
(MAPGISIMS.EnterpriseLib.OpCSpatialOverlayByID featureIDs)**

说明

要素叠加分析

参数

featureIDs: 需要叠加的要素

返回值

要素叠加分析结果(点序列)

备注

该方法隶属于 CMapView 对象。

## 6.81 UpdateFeature 方法

**public bool UpdateFeature(MAPGISIMS.EnterpriseLib.OpCUpdateFeatureAttribute  
newAttribute)**

说明

更新要素属性

**参数**

newAttribute: 更新要素对象

**返回值**

成功—true,失败—false

**备注**

该方法隶属于 ViewPort 对象。

**示例**

```
//定义更新要素类对象、返回值对象
OpCUpdateFeatureAttribute newAttribute = new OpCUpdateFeatureAttribute();

bool updateRltFlag = false;
//为新要素对象赋值
newAttribute.FieldNumber = (ushort)fldArr.Length;//字段个数
newAttribute.NewFieldValues = valArr;//字段值
newAttribute.LayerIndex = layerIndex;//要素所在图层
newAttribute.FeatureID = featureID;//要素ID号
//调用要素更新函数
updateRltFlag = viewport.UpdateFeature(newAttribute);
//根据返回值对象判断要素文件修改成功与否
if (updateRltFlag == false)
{
    Response.Write("<script language=javascript>alert('要素类文件修改失败!')</script>");
}
else
{
    Response.Write("<script language=javascript>alert('要素类文件修改成功!');window.close();</script>");
}
}
```

**6.82 UpdateFeatureGeomtry 方法**

**public bool UpdateFeatureGeomtry(MAPGISIMS.EnterpriseLib.OpCUpdateFeature newFeatureGeomtry)**

**说明**

更新要素几何信息

**参数**

newFeatureGeomtry: 更新要素对象

**返回值**

成功—true,失败—false

**备注**

该方法隶属于 ViewPort 对象。

**示例**

```
featureToUpdate.NewFeature = featureGeometry;
bool succ = viewport.UpdateFeatureGeomtry(featureToUpdate);
```

**6.83 UpdateFeatureStyleInfo 方法**

**public MAPGISIMS.EnterpriseLib.OpCUniformResult UpdateFeatureStyleInfo (MAPGISIMS.EnterpriseLib.OpCUpdateFeatureStyleInfo updatePatter)**

**说明**

更新要素样式，如颜色，高宽等

**参数**

updatePatter: 更新要素样式类

返回值

OpCUniformResult 类

备注

该方法隶属于 ViewPort 对象

## 6.84 UpdateLayer 方法

**public short UpdateLayer(MAPGISIMS.EnterpriseLib.OpCLayerInformation  
layerObj)**

说明

更新图层

参数

layerObj: 更新图层对象

返回值

成功-true, 失败-false

备注

该方法隶属于 ViewPort 对象

示例

```

OpCLayerInformation mapLayerObj = null;
getMapLayerObj(layerNo, out mapLayerObj);
//layerType = mapLayerObj.LayerType;
switch (Convert.ToInt32(LsArr[i]))
{
    case 0:
        mapLayerObj.Visible = 0;
        mapLayerObj.Selectable = 0;
        mapLayerObj.Editable = 0;
        mapDocArr[0].Layers[i].LayerStatus = Mapgis71WebControl.MCC_LayerStatus.Close;
        break;
    case 1:
        mapLayerObj.Visible = 1;
        mapLayerObj.Selectable = 0;
        mapLayerObj.Editable = 0;
        //if(layerType==3)
        layerIndexLst += i + ",";
        mapDocArr[0].Layers[i].LayerStatus = Mapgis71WebControl.MCC_LayerStatus.Visible;
        break;
    case 2:
        mapLayerObj.Visible = 1;
        mapLayerObj.Selectable = 1;
        mapLayerObj.Editable = 0;
        mapDocArr[0].Layers[i].LayerStatus = Mapgis71WebControl.MCC_LayerStatus.Selectable;
        layerIndexLst += i + ",";
        layerSelectCount++;
        Session["strConditonLayerNo"] = i;
        break;
    case 3:
        mapLayerObj.Visible = 1;
        mapLayerObj.Selectable = 1;
        mapLayerObj.Editable = 1;
        Session["strLayerNo"] = i; //图层索引号
        Session["strConditonLayerNo"] = i;
        layerIndexLst += i + ",";
        Session["searchType"] = mapLayerObj.LayerType.ToString(); //图层类型
        mapDocArr[0].Layers[i].LayerStatus = Mapgis71WebControl.MCC_LayerStatus.Editable;
        layerSelectCount++;
        break;
    default:
        break;
}
try
{
    viewport.UpdateLayer(mapLayerObj);
}

```

## 6.85 UpdateLayerLst 方法

**public short UpdateLayerLst(MAPGISIMS.EnterpriseLib.OpCUpdateLayerLst layerObj)**

**说明**

更新多个图层列表。

**参数**

layerObj: 更新图层对象

**返回值**

成功—true, 失败—false

**备注**

该方法隶属于 CMapView 对象。

**示例**

```

if (layerIndexLst.Length > 0)
{
    layerIndexLst = layerIndexLst.Substring(0, layerIndexLst.Length - 1);
}
Session["layerIndexLst"] = layerIndexLst;
try
{
    viewport.UpdateLayerLst(maplayer);
}

```

## 6.86 WindowToLogical 方法

**public void WindowToLogical(MAPGISIMS.EnterpriseLib.OpCPoint2D windowPoint, ref MAPGISIMS.EnterpriseLib.OpCPoint2D logicalPoint)**

说明

窗口坐标转逻辑坐标。

参数

windowPoint: 窗口坐标

logicalPoint: 逻辑坐标

备注

该方法隶属于 ViewPort 对象。

## 6.87 ZoomIn 方法

**public short ZoomIn(int xmin, int ymin, int xmax, int ymax)**

说明

地图的拉框放大操作。

参数

xmin: 矩形框xmin窗口坐标

ymin: 矩形框ymin窗口坐标

xmax: 矩形框xmax窗口坐标

ymax: 矩形框 ymax 窗口坐标

返回值

成功-1,失败--1

备注

该方法隶属于 ViewPort 对象。

示例

```

//输入参数为客户端传来的矩形框窗口坐标
viewport.ZoomIn(xmin,ymin,xmax,ymax);

```

## 6.88 ZoomOut 方法

**public short ZoomOut(int xmin, int ymin, int xmax, int ymax)**

说明

地图的拉框缩小操作。

参数

xmin: 矩形框xmin窗口坐标

ymin: 矩形框ymin窗口坐标

xmax: 矩形框xmax窗口坐标

ymax: 矩形框 ymax 窗口坐标

**返回值**

成功—1,失败—-1

**备注**

该方法隶属于 ViewPort 对象。

**示例**

```
//输入参数为客户端传来的矩形框窗口坐标
viewport.ZoomOut(xmin,ymin,xmax,ymax);
```

## 6.89 FeatureClassSelect 方法

**public MAPGISIMS.EnterpriseLib.OpCSelectResult**

**FeatureClassSelect(MAPGISIMS.EnterpriseLib.OpCGetFclsSel spacialRltSelct)**

**说明**

单要素类查询。

**参数**

spacialRltSelct: 要素类查询对象

**返回值**

要素类查询结果

**备注**

该方法隶属于 CMapView 对象。

**示例**

```
OpCSelectResult selectRlt = viewport.FeatureClassSelect(fclsSel);
long[] fIds = selectRlt.LayerSelectResults[0].FeatureIDs;
```

## 6.90 GetFclsRect 方法

**public MAPGISIMS.EnterpriseLib.OpCRect**

**GetFclsRect(MAPGISIMS.EnterpriseLib.OpCGetFclsRect getFcls)**

**说明**

获取某个要素类的逻辑范围。

**参数**

getFcls: 获取要素类范围对象

**返回值**

要素类逻辑范围

**备注**

该方法隶属于 CMapView 对象。

## 6.91 GetFeatureClassImg 方法

**public byte[] GetFeatureClassImg(string imgFileName, string spacialAnlyRltHDFName, string spacialAnlyRltfName, string netSvrName, string username, string password, MAPGISIMS.EnterpriseLib.LayerTypeEnum layerType)**

**说明**

根据图层类型获取单图层图像，返回图片字节流。

**参数**

imgFileName: 图片名称  
 spacialAnlyRltHDFName: HDF 名称  
 spacialAnlyRltfName: 要素类名称  
 netSvrName: 网络服务名(默认为 MapGisLocal)  
 username: 用户名(默认为空)  
 password: 密码(默认为空)  
 layerType: 图层类型 (要素类、简单要素类)

**备注**

该方法隶属于 CMapView 对象。

**示例**

```
Response.Clear();
Response.ContentType = "image/gif";
byte[] buffer = viewport.GetFeatureClassImg(imageName, hdf, fsc, netServer, "", "", viewport.LastAnalyseLayerType);
Response.BinaryWrite(buffer);
```

**6.92 GetFeatureClassImg 方法**

**public byte[] GetFeatureClassImg(string imgFileName, string spacialAnlyRltHDFName, string spacialAnlyRltfName, string netSvrName, string username, string password)**

**说明**

获取要素类图像，返回图片字节流。

**参数**

imgFileName: 图片名称  
 spacialAnlyRltHDFName: HDF 名称  
 spacialAnlyRltfName: 要素类名称  
 netSvrName: 网络服务名(默认为 MapGisLocal)  
 username: 用户名(默认为空)  
 password: 密码(默认为空)

**备注**

该方法隶属于 CMapView 对象。

**示例**

```
Response.Clear();
Response.ContentType = "image/gif";
byte[] buffer = viewport.GetFeatureClassImg(imageName, hdf, fsc, netServer, "", "", viewport.LastAnalyseLayerType);
Response.BinaryWrite(buffer);
```

**6.93 GetImageDisplayParam 方法**

**public void GetImageDisplayParam(out bool originalFlag, out bool showScale, out bool showCoordinate, out bool followScale, out bool bufferShow, out bool dynamicNote, out MAPGISIMS.EnterpriseLib.ImageQuality imageQuality)**

**说明**

获取图像显示参数。

**参数**

originalFlag: 是否还原显示  
 showScale: 是否标尺  
 showCoordinate: 是否显示坐标点  
 followScale: 是否跟随比例显示  
 bufferShow: 是否缓冲显示  
 imageQuality: 图像质量 1 低,2 中,3 高  
 dynamicNote: 是否显示动态标注

**备注**

该方法隶属于 CMapView 对象。

**示例**

```
viewport = (CMapView)Session["viewport"];
viewport.GetImageDisplayParam(out originalFlag, out showScale, out showCoordinate, out followScale,
    out bufferShow, out dynamicNote, out imageQuality);
```

**6.94 GetSingleFclsGeometry 方法**

**public MAPGISIMS.EnterpriseLib.OpCFeatureGeometries  
 GetSingleFclsGeometry(MAPGISIMS.EnterpriseLib.OpCGetSingleFclsGeo  
 getFeatureGeo)**

**说明**

获取单要素类某个要素空间信息。

**参数**

getFeatureGeo: 获取某个要素类中某个要素空间信息类

**备注**

该方法隶属于 CMapView 对象。

**示例**

```
try
{
    featureGeometry = viewport.GetSingleFclsGeometry(geo).pInfo[0];
}
```

**6.95 GetZipFile 方法**

**public string GetZipFile(MAPGISIMS.EnterpriseLib.OpCGetZipFile zip)**

**说明**

获取包含指定 HDF 和要素类的压缩文件。

**参数**

zip: 获取包含指定 HDF 和要素类压缩文件的参数对象

**备注**

该方法隶属于 CMapView 对象。

**6.96 SetAsyncTimeout 方法**

**public static void SetAsyncTimeout(int timeout)**

**说明**

设置异步操作时等待时间。

**参数**

timeout: 等待时间（毫秒）

**备注**

该方法隶属于 CMapView 对象。

### 6.97 SetImageFolder 方法

**public static void SetImageFolder(string imageFolder)**

**说明**

设置图片生成文件夹。

**参数**

imageFolder: 文件夹路径

**备注**

该方法隶属于 CMapView 对象。

### 6.98 SetLogPath 方法

**public static void SetLogPath(string logPath)**

**说明**

设置日志文件生成路径。

**参数**

logPath: 日志文件生成路径

**备注**

该方法隶属于 CMapView 对象。

### 6.99 SetSyncTimeout 方法

**public void SetSyncTimeout(int sendTimeout, int receiveTimeout)**

**说明**

设置同步操作时发送和接收数据的超时时间。

**参数**

sendTimeout: 发送数据超时时间（毫秒）

receiveTimeout: 接收数据超时时间（毫秒）

**备注**

该方法隶属于 CMapView 对象。

### 6.100 SetWriteLogFlag 方法

**public static void SetWriteLogFlag(bool writeLogFlag)**

**说明**

设置是否生成日志文件。

**参数**

writeLogFlag: 是否生成日志

**备注**

该方法隶属于 CMapView 对象。

### 6.101 SpatialAnalysisBufferByGeom 方法

**public MAPGISIMS.EnterpriseLib.OpCSpatialOperResult  
SpatialAnalysisBufferByGeom(MAPGISIMS.EnterpriseLib.OpCSpatialBufferByGeom  
bufferGeo)**

说明

缓冲区分析,分析结果生成的图像大小和范围默认都是当前地图的范围和大小。

参数

bufferGeo: 缓冲区分析对象

返回值

缓冲区分析结果

备注

该方法隶属于 CMapView 对象。

示例

```
try
{
    viewport.SpatialAnalysisBufferByGeom(buffByGeom);
}
```

### 6.102 SpatialAnlyFeatureOverlay 方法

**public MAPGISIMS.EnterpriseLib.OpCSpatialOperResult  
SpatialAnlyFeatureOverlay(MAPGISIMS.EnterpriseLib.OpCSpatialOverlayFCIsAnd  
Geom overlay)**

说明

要素与要素类叠加分析,分析结果生成的图像大小和范围默认都是当前地图的范围和大小。

参数

overlay: 叠加分析对象

返回值

查询分析结果

备注

该方法隶属于 CMapView 对象。

示例

```
try{
    viewport.SpatialAnlyFeatureOverlay(fcIsWithGeom);
}catch(Exception exx){
    ErrorHandler.EndWithMsg("分析失败。");
    return;
}
```

### 6.103 SpatialDiffer 方法

**public MAPGISIMS.EnterpriseLib.TopoTypeEnum  
SpatialDiffer(MAPGISIMS.EnterpriseLib.OpCSpatialDiffer overlay)**

说明

判断两个要素之间的拓扑关系,分析结果生成的图像大小和范围默认都是当前地图的范围和大小。

**参数**

overlay: 叠加分析对象

**返回值**

叠加分析结果

**备注**

该方法隶属于 CMapView 对象。

**示例**

```
protected void AddTopo2 ()
{
    CMapView viewport = (CMapView)Session["viewport"];
    OpCSpatialDiffer sd = (OpCSpatialDiffer)Session["Differ"];
    sd.FeatureGeometry2 = GetGeom(viewport);
    Session["Differ"] = sd;
}
```

**6.104 SpatialOverlayByGeom 方法**

```
public MAPGISIMS.EnterpriseLib.OpCOverlayByIDPntSeq
SpatialOverlayByGeom(MAPGISIMS.EnterpriseLib.OpCSpatialOverlayByGeom
featureGeom)
```

**说明**

要素叠加分析。

**参数**

featureGeom: 需要叠加的要素空间信息

**返回值**

要素叠加分析结果（点序列）

**备注**

该方法隶属于 CMapView 对象。

**7. CMapViewPort 类**

地图视图类。主要用来取图与保存图片。

**7.1 CMapViewPort**

```
public CMapViewPort()
```

**说明:**

初始化CMapViewPort类的新实例。

**参数:**

无

**返回值:**

无

## 7.2 CMapViewPort

### **public CMapViewPort (string ipaddress ,int port )**

**说明:**

初始化具有GIS服务器IP地址和端口号的CMapViewPort类的新实例

**参数:**

ipaddress: GIS服务器IP地址

port: GIS 服务器端口号

**返回值:**

无

**备注:**

带参数构造函数

**示例:**

```
CMapViewPort viewport = new
```

```
CMapViewPort(gisConfig[No].GisServerIP, gisConfig[No].GisServerPort);
```

## 7.3 CMapViewPort

### **public CMapViewPort(string ipaddress, int port, int bufferSize)**

**说明:**

初始化具有GIS服务器IP地址、端口号，图像缓存的CMapViewPort类的新实例

**参数:**

ipaddress: GIS服务器IP地址

port: GIS服务器端口号

bufferSize: 图像缓存(MB)

**返回值:**

无

**备注:**

带参数构造函数

## 7.4 ComputerGlobeIndex 方法

### **public void ComputerGlobeIndex(double dgree)**

**说明:**

计算按全球经纬度划分的网格索引

**参数:**

dgree: 每网格对应的经度度数

**返回值:**

无

**备注:**

隶属于 CmapViewPort 对象

## 7.5 GetMapImage 方法

**public byte[] GetMapImage(MAPGISIMS.EnterpriseLib.HDFConfig gisConfig, int rowNo, int lineNo, int level, bool isGlobe)**

**说明:**

根据HDF地图库设置, 获取指定参数下的地图图片

**参数:**

gisConfig: HDF地图库配置

rowNo: 请求图片所在的行号

lineNo: 请求图片所在的列号

level: 请求图片所在的级数

isGlobe: 是否为经纬度切分HDF地图库

**返回值:**

返回地图图片字节数组

## 7.6 GetMapImage 方法

**public byte[] GetMapImage(string HDFName, int iStartLvl, int rowNo, int lineNo, int level, int dealSize)**

**说明:**

获取指定参数下的地图图片(推荐用此方法)

**参数:**

HDFName: 文件名

iStartLvl: HDF库存储图片开始的级数,一般为0

rowNo: 请求图片所在的行号

lineNo: 请求图片所在的列号

level: 请求图片所在的级数

dealSize: 地图缓冲处理的大小

**返回值:**

返回地图图片字节数组

## 7.7 GetMapImage 方法

**public byte[] GetMapImage(string HDFName, int iStartLvl, int rowNo, int lineNo, int level, int dealSize, int imgWidth )**

**说明:**

获取指定参数下的地图图片

**参数:**

HDFName: 文件名

iStartLvl: HDF库存储图片开始的级数,一般为0

rowNo: 请求图片所在的行号

lineNo: 请求图片所在的列号

level: 请求图片所在的级数  
dealSize: 地图缓冲处理的大小  
imgWidth: 单个地图图片的大小  
返回值:  
返回地图图片字节数组

## 7.8 GetMapImage 方法

**public byte[] GetMapImage(string HDFName, int iStartLvl, int rowNo, int lineNo, int level)**

**说明:**

获取指定参数下的地图图片

**参数:**

HDFName: HDF文件名

iStartLvl: HDF库存储图片开始的级数

rowNo: 请求图片所在的行号

lineNo: 请求图片所在的列号

level: 请求图片所在的级数

**返回值:**

返回地图图片字节数组

**示例:**

```
public byte[] SaveMap(HDFConfig[] gisConfig, MapRectangle rect, int
mapLevel, ImageFormat fileFormat)
{
.....
CMapViewPort viewport = new
CMapViewPort(gisConfig[No].GisServerIP, gisConfig[No].GisServerPort);
byte[] buffer = null;
buffer = viewport.GetMapImage
(gisConfig[No].HDFName, gisConfig[No].StartLevel, row, cell, mapLevel);
.....
}
```

## 7.9 SaveMap 方法

**public void SaveMap(MAPGISIMS.EnterpriseLib.HDFConfig[] gisConfig, MAPGISIMS.EnterpriseLib.MapRectangle rect, int mapLevel, System.Drawing.Imaging.ImageFormat fileFormat, string savePath)**

**说明:**

保存指定逻辑范围内的地图为文件

**参数:**

gisConfig: HDF相关配置

rect: 指定的逻辑范围

mapLevel: 地图级数

fileFormat: 图片保存格式

savePath: 指定的保存路径

返回值: 无

备注: 示例中设置了HDF信息, 并且通过SaveMap方法保存地图到指定位置

示例: HDFConfig[] config = new HDFConfig[1];

```
config[0].GisServerIP = "127.0.0.1";
```

```
config[0].GisServerPort = 5142;
```

```
config[0].HDFName = "whmap.HDF";
```

```
config[0].MapRect = logicCoordinate;
```

```
string strpath="D: \\\\"
```

```
//保存地图的范围
```

```
MapRectangle windowCoordinate = new MapRectangle(startX, startY, endX, endY);
```

```
CMapViewPort image = new CMapViewPort();
```

```
//得到地图数据流
```

```
Response.BinaryWrite(image.SaveMap(config,windowCoordinate,level,System.Drawing.Imaging.ImageFormat.Jpeg,strpath));
```

## 7.10 SaveMap 方法

**public byte[] SaveMap(MAPGISIMS.EnterpriseLib.HDFConfig[] gisConfig, MAPGISIMS.EnterpriseLib.MapRectangle rect, int mapLevel, System.Drawing.Imaging.ImageFormat fileFormat)**

说明:

保存指定逻辑范围内的地图

参数:

gisConfig: HDF相关配置

rect: 指定的逻辑范围

mapLevel: 地图级数

fileFormat: 图片保存格式

返回值:

返回图片的字节数组

示例:

```
HDFConfig[] config = new HDFConfig[1];
```

```
config[0].GisServerIP = "127.0.0.1";
```

```
config[0].GisServerPort = 5142;
```

```
config[0].HDFName = "whmap.HDF";
```

```
config[0].MapRect = logicCoordinate;
```

```
//保存地图的范围
```

```
MapRectangle windowCoordinate = new MapRectangle(startX, startY, endX, endY);
```

```
CMapViewPort image = new CMapViewPort();
```

```
//得到地图数据流
```

```
Response.BinaryWrite(image.SaveMap(config,windowCoordinate,level,System.Drawing.Ima
```

```
ging.ImageFormat.Jpeg));
```

## 7.11 Dispose 方法

### **Public void Dispose()**

**说明:**

释放CmapViewPort所占用的资源。

**参数:**

无

**返回值:**

无

## 7.12 SetIndexOffset 方法

### **public void SetIndexOffset(long offset)**

**说明**

取图时的索引偏移量，老版HDF的偏移量为2，新版的为0，默认为2

**参数**

offset: 索引偏移量

**返回值**

无

## 8. CMpAtt 类

图层属性结构类

### 8.1 addLayerAttStru 方法

#### **public int addLayerAttStru(MAPGISIMS.EnterpriseLib.OpCSelectResult selectRlt)**

**说明:**

往图层属性结构对象里添加图层属性结构，如果已经有了则不添加。

**参数:**

selectRlt: 查询结果

**返回值:**

成功返回 1，失败返回小于 0 的值

**备注**

隶属于 CMpAtt 对象

## 8.2 CMpAtt 方法

**public CMpAtt(int layerCount)**

说明:

构造函数

参数:

layerCount: 地图文档图层数目

返回值:

无

备注:

构造函数

## 8.3 GetFieldName 方法

**public short GetFieldName(int LayerIndex, out string[] FldName)**

说明:

返回图层的属性字段

参数:

LayerIndex: 图层索引

FldName: 属性字段名

返回值:

大于 0 成功, 小于 0 失败

备注:

隶属于 CmapViewPort 对象

## 8.4 GetFieldNumber 方法

**public short GetFieldNumber(int LayerIndex, out int sFldNum)**

说明:

返回某个图层属性字段个数

参数:

LayerIndex: 图层索引

sFldNum: 属性字段个数

返回值:

大于 0 成功, 小于 0 失败

备注:

隶属于 CmapViewPort 对象

## 8.5 GetLayerRecordCount 方法

**public short GetLayerRecordCount(int LayerIndex, out long lCount)**

**说明:**

获取某个图层查询结果记录个数

**参数:**

LayerIndex: 图层索引

ICount: 图层查询结果记录个数

**返回值:**

大于 0 成功, 小于 0 失败

**备注:**

隶属于 CmapViewPort 对象

## 8.6 GetRecordByID 方法

**public short GetRecordByID(short sLayerIndex, short index, short fldId, out object valObj)**

**参数:**

sLayerIndex:

index:

fldId:

valObj:

**备注:**

隶属于 CmapViewPort 对象

## 8.7 ITotalFNum 属性

**public long ITotalFNum**

**说明:**

记录个数

**备注:**

隶属于 CmapViewPort 对象

## 8.8 sLayerNum 属性

**public long sLayerNum**

**说明:**

图层个数

**备注:**

隶属于 CmapViewPort 对象

## 9. CNetEdge 类

网络边线类

## 9.1 CNetEdge 方法

**public CNetEdge()**

说明:

网络边线构造函数

备注

隶属于 CMpAtt 对象

## 9.2 CNetEdge 方法

**public CNetEdge(MAPGISIMS.EnterpriseLib.Dot\_2D[ ] dotsData, string[ ] fldVals)**

说明

带参数的网络边线构造函数

参数

dotsData: 组成线的结点数组

fldVals: 网络边线属性列表

备注

隶属于 CMpAtt 对象

## 9.3 Dots 属性

**public MAPGISIMS.EnterpriseLib.Dot\_2D[ ] Dots**

说明

组成线的结点数组

备注

隶属于 CMpAtt 对象

## 9.4 FieldValus 属性

**public string[ ] FieldValus**

说明

网络边线属性列表

备注

隶属于 CMpAtt 对象

## 10. CNetInputDot 类

输入点信息。

## 10.1 CNetInputDot 方法

**public CNetInputDot ()**

说明:

输入点信息构造函数

备注

隶属于 CNetInputDot 对象

## 10.2 CNetInputDot 方法

**public CNetInputDot(MAPGISIMS.EnterpriseLib.Dot\_2D pDot, double posPercent)**

说明

网络分析中的有效输入点

参数

pDot: 线上点

posPercent: 垂点在边线上的位置百分比

备注

隶属于 CNetInputDot 对象

## 10.3 pDot 属性

**public MAPGISIMS.EnterpriseLib.Dot\_2D pDot**

说明

输入点在边线上的垂点坐标

备注

隶属于 CNetInputDot 对象

## 10.4 positionPercent 属性

**public double positionPercent**

说明

垂点在边线上的位置百分比

备注

隶属于 CNetInputDot 对象

## 11. CNetNode 类

网络结点。

## 11.1 CNetNode 方法

### **public CNetNode ()**

说明:

网络结点的构造函数

备注

隶属于 CNetNode 对象

## 11.2 CNetNode 方法

### **public CNetNode(MAPGISIMS.EnterpriseLib.Dot\_2D nodeData, string[] fldVals)**

说明

带参数的网络结点构造函数

参数

nodeData: 网络结点

fldVals: 网络结点的属性列表

备注

隶属于 CNetNode 对象

## 11.3 CNetNode 方法

### **public CNetNode(MAPGISIMS.EnterpriseLib.Dot\_2D nodeData, string[] fldVals, double turnAngle)**

说明

带参数的网络结点构造函数

参数

nodeData: 网络结点

fldVals: 网络结点的属性列表

turnAngle: 与该结点相连边线的夹角

备注

隶属于 CNetNode 对象

## 11.4 FieldValus 属性

### **public string[] FieldValus**

说明

网络结点的属性列表

备注

隶属于 CNetNode 对象

## 11.5 Node 属性

**public MAPGISIMS.EnterpriseLib.Dot\_2D Node**

说明

网络结点

备注

隶属于 CNetNode 对象

## 11.6 TurnAngle 属性

**public double TurnAngle**

说明

与该结点相连的边线夹角

值域为[-180,180]

[-180,0]右转, [0,180]左转

180 度直行(看实际情况, 度数小可以近似的认为是直行), 0 度折返

备注

隶属于 CNetNode 对象

## 12. CNetPath 类

网络路径。

### 12.1 CNetPath 方法

**public CNetPath()**

说明:

网络路径构造函数

备注

隶属于 CNetPath 对象

### 12.2 CNetPath 方法

**public CNetPath(MAPGISIMS.EnterpriseLib.CNetEdge[ ] edgesData,  
MAPGISIMS.EnterpriseLib.CNetNode[ ] nodesData)**

说明

网络路径

参数

edgesData: 弧段数组

nodesData: 结点数组

备注

隶属于 CNetPath 对象

## 12.3 Edges 属性

**public MAPGISIMS.EnterpriseLib.CNetEdge[ ] Edges**

说明

弧段数组

备注

隶属于 CNetPath 对象

## 12.4 Nodes 属性

**public MAPGISIMS.EnterpriseLib.CNetNode[ ] Nodes**

说明

结点数组

备注

隶属于 CNetPath 对象

## 13. CNetPathAnaly 类

路径分析类

### 13.1 CNetPathAnaly 方法

**public CNetPathAnaly(string gdbSvrName, string gdbSvrUser, string gdbSvrPwds, string gdbName, string fdsName, bool isJW)**

说明:

网络分析构造函数

参数:

gdbSvrName: GDB服务器名称

gdbSvrUser: GDB服务器用户名

gdbSvrPwds: GDB服务器密码

gdbName: GDB名称

fdsName: 要素数据集名称

isJW: 是否经纬度坐标

返回值:

无

备注

隶属于 CNetPathAnaly 对象

## 13.2 OpenPathAnalySvr 方法

**public void OpenPathAnalySvr()**

说明:

打开 MapGIS 服务连接

参数:

无

备注

隶属于 CNetPathAnaly 对象

## 13.3 ClosePathAnalySvr 方法

**public void ClosePathAnalySvr()**

说明:

关闭 MapGIS 服务连接

参数:

无

备注

隶属于 CNetPathAnaly 对象

## 13.4 PathAnalyze 方法

**public MAPGISIMS.EnterpriseLib.CPathAnalyzeResult PathAnalyze  
(MAPGISIMS.EnterpriseLib.Dot\_2D[] inputDotArr, string netName, double  
nearDistance)**

说明

路径分析

参数

inputDotArr: 输入点的坐标序列

netName: 网文件名称

nearDistance: 输入点的捕捉(查找)半径, 与地图坐标系相关, 一般为10个屏幕象素所代表的逻辑长度

返回值

路径分析结果

备注

隶属于CNetPathAnaly对象

## 13.4 SearchDriveWay 方法

**public string SearchDriveWay(MAPGISIMS.EnterpriseLib.Dot\_2D[] dotArr, int netId,  
double maxDistance, int backFlag)**

**说明:**

同一城市路径分析,返回坐标和描述信息

**参数:**

dotArr: 路径分析所经过的点

netId: 网文件ID

maxDistance: 点到线上的投影半径。如未找到路径,请重新调整投影半径

backFlag: 返回数据的选择,0表示全部,1表示只返回坐标,2表示只返回描述信息

**返回值:**

返回x, y坐标串和描述信息字符串

**备注**

隶属于 CNetPathAnaly 对象

## 13.5 SearchDriveWay 方法

```
public string SearchDriveWay(MAPGISIMS.EnterpriseLib.Dot_2D[] dotArr, string netName, double maxDistance, int backFlag)
```

**说明:**

同一城市路径分析,返回坐标和描述信息

**参数:**

dotArr: 路径分析所经过的点

netName: 网文件名称

maxDistance: 点到线上的投影半径。如未找到路径,请重新调整投影半径

backFlag: 返回数据的选择,0表示全部,1表示只返回坐标,2表示只返回描述信息

**返回值:**

返回 x, y 坐标串和描述信息字符串

**备注**

隶属于 CNetPathAnaly 对象

## 13.6 SetBarrier 方法

```
public void SetBarrier(MAPGISIMS.EnterpriseLib.Dot_2D[] barrierDotArr)
```

**说明:**

设置障碍

**参数:**

barrierDotArr: 障碍点系列

**返回值:**

无

**备注**

隶属于 CNetPathAnaly 对象

## 13.7 SetDscrInfoParam 方法

**public void SetDscrInfoParam(int edgeNameIndex, int edgeLengthIndex, int nodeNameIndex, int conversionRate)**

**说明:**

设置路径分析描述中所需的相关参数

**参数:**

edgeNameIndex: 弧段名称字段在弧段属性中的序号

edgeLengthIndex: 弧段长度字段在弧段属性中的序号

nodeNameIndex: 结点名称字段在结点属性中的序号

conversionRate: 长度换算比率。即1弧段长度等于多少米

**返回值:**

无

**备注**

隶属于 CNetPathAnaly 对象

## 13.8 SetNetWeight 方法

**public void SetNetWeight(string[] netWeightArr)**

**说明:**

设置路络权

**参数:**

netWeightArr: 网络权值数组

**返回值:**

无

**备注**

隶属于 CNetPathAnaly 对象

## 14. CPathAnalyzeResult 类

网络分析结果类

### 14.1 CPathAnalyzeResult 方法

**public CPathAnalyzeResult ()**

**说明:**

网络分析结果类构造函数

**备注**

隶属于CPathAnalyzeResult 对象

## 14.2 edgeFieldNameArray 属性

**public string[ ] edgeFieldNameArray**

说明:

边线字段名数组

备注

隶属于 CPathAnalyzeResult 对象

## 14.3 inputDots 属性

**public MAPGISIMS.EnterpriseLib.CNetInputDot[ ] inputDots**

说明:

输入点的信息

备注

隶属于 CPathAnalyzeResult 对象

## 14.4 nodeFieldNameArray 属性

**public string[ ] nodeFieldNameArray**

说明:

结点字段名数组

备注

隶属于 CPathAnalyzeResult 对象

## 14.5 Paths 属性

**public MAPGISIMS.EnterpriseLib.CNetPath[ ] Paths**

说明:

路径信息

备注

隶属于 CPathAnalyzeResult 对象

## 15. CWebMap 类

地图文档类

## 属性

### 15.1 LoadMapRlt 属性

**public MAPGISIMS.EnterpriseLib.OpCLoadMapResult loadMapRlt**

说明

地图加载结果

备注

该属性隶属于 CWebMap 对象

示例

```
OpCCircleSelectClass circleSelectObj = new OpCCircleSelectClass ();  
circleSelectObj.Center = p2d;  
circleSelectObj.Radius = radius;  
circleSelectObj.MapID = viewport.MapObj.LoadMapRlt.MapID;
```

### 15.2 mapCount 属性

**Public int mapCount**

说明

获取地图的图层个数

备注

该属性隶属于 CWebMap 对象

### 15.3 MapRect 属性

**public MAPGISIMS.EnterpriseLib.OpCRect MapRect**

说明

地图文档的地图范围

备注

该属性隶属于 CWebMap 对象

### 15.4 m\_layerObj 属性

**public MAPGISIMS.EnterpriseLib.CLayerObj[] m\_layerObj**

说明

图层对象

备注

该属性隶属于 CWebMap 对象

### 15.5 m\_LoadMapRlt 属性

**public MAPGISIMS.EnterpriseLib.OpCLoadMapResult m\_LoadMapRlt**

说明

地图返回结果

备注

该属性隶属于 CWebMap 对象

## 方法

### 15.6 CWebMap 方法

**public CWebMap(MAPGISIMS.EnterpriseLib.OpCLoadMapResult loadMapRlt)**

说明

构造函数。

备注

该方法隶属于 CWebMap 对象。

### 15.7 GetItem 方法

**public short GetItem(MAPGISIMS.EnterpriseLib.IMSOBJArray rdat, out MAPGISIMS.**

**EnterpriseLib.OpCLayerInformation pLayer)**

说明

得到图层对象。

参数

rdat: 数组对象

pLayer: 图层对象

返回值

大于 0 成功, 小于 0 失败

备注

该方法隶属于 CWebMap 对象。

示例

```
OpCLayerInformation pMapLayer = null;
IMSOBJArray rdat = new IMSOBJArray();
//声明函数中返回的地图文档中图层个数
layerNum = -1;
//声明函数的返回值
int sLayerNum = -1, sItem = -1;
//得到地图文档中图层的个数
sLayerNum = viewport.MapObj.GetLayerNum(out layerNum);
mapDocArr[0].Layers = new CLayerInformation[layerNum];
for (int i = 0; i < layerNum; i++)
{
    mapDocArr[0].Layers[i] = new CLayerInformation();
    //依次向集合中添加图层
    rdat.Append(i);
    //依次得到图层对象
    sItem = viewport.MapObj.GetItem(rdat, out pMapLayer);
}
```

### 15.8 getLayerAttStruct 方法

**public MAPGISIMS.EnterpriseLib.OpCLayerAttStructure getLayerAttStruct(long layerIndex)**

说明

获取图层的属性结构。

参数

layerIndex: 图层索引

返回值

OpCLayerAttStructure 类

备注

该方法隶属于 CWebMap 对象。

## 15.9 getLayerInformation 方法

**public MAPGISIMS.EnterpriseLib.OpCLayerInformation getLayerInformation(long layerIndex)**

说明

获取图层信息。

参数

layerIndex: 图层索引

返回值

OpCLayerInformation 类

备注

该方法隶属于 CWebMap 对象。

示例

```
for (long i = 0; i < layerNum; i++)
{
    layerNo = i.ToString();
    OpCLayerInformation mapLayerObj = null;
    mapLayerObj=viewport.MapObj.getLayerInformation(i);
    switch (Convert.ToInt32(LayerStatus[i]))
    {
        case 0://关闭
            mapLayerObj.Visible = 0;
            mapLayerObj.Selectable = 0;
            mapLayerObj.Editable = 0;
            visibleOrClosedLayerCount++;
            break;
    }
}
```

## 15.10 GetLayerNum 方法

**Public Short GetLayerNum(out int layerNum);**

说明

获取当前地图文档中图层个数。

参数

LayerNum: 图层数目

返回值

大于 0 成功, 小于 0 失败

备注

该方法隶属于 CWebMap 对象。

示例

```
//声明函数中返回的地图文档中图层个数
layerNum = -1;
//声明函数的返回值
int sLayerNum = -1, sItem = -1;
//得到地图文档中图层的个数
sLayerNum = viewport.MapObj.GetLayerNum(out layerNum);
```

### 15.11 getLayerType 方法

**public Int mapObj.getLayerType(long layerIndex);**

说明

获取图层类型。

参数

layerIndex: 图层索引

返回值

大于 0 成功, 小于 0 失败

备注

该方法隶属于 CWebMap 对象。

### 15.12 getLayerType 方法

**public MAPGISIMS.EnterpriseLib.LayerTypeEnum GetLayerType(int layerIndex);**

说明

获取图层类型 (要素类、简单要素类)。

参数

layerIndex: 图层索引

备注

该方法隶属于 CWebMap 对象。

### 15.13 setLayerAttStru 方法

**public int setLayerAttStru(MAPGISIMS.EnterpriseLib.OpCLayerAttStructure layerAttStru, long layerIndex)**

说明

添加图层属性结构。

参数

layerAttStru: 图层属性结构

layerIndex: 图层索引

返回值

大于 0 成功, 小于 0 失败

备注

该方法隶属于 CWebMap 对象。

### 15.14 setLayerType 方法

**public int setLayerType(int layerType, long layerIndex)**

说明

为指定图层设置图层类型。

参数

layerType: 图层类型

layerIndex: 图层索引

返回值

大于 0 成功, 小于 0 失败

**备注**

该方法隶属于 CWebMap 对象。

## 16. GISDataQuery 类

数据查询类，此类主要提供图形查询相关的方法。该类中的查询方法可以支持 SQL Server 和 Oracle 两种数据库类型的图形查询，通过设置参数 DatabaseType 判断数据库类型。

### 16.1 GISDataQuery 方法

**public GISDataQuery(MAPGISIMS.EnterpriseLib.DatabaseType dbType)****说明：**

GISDataQuery 的构造函数

**参数：**

dbType: 指定数据查询所使用的数据库类型

**返回值：**

无

**备注：**

隶属于GISDataQuery对象

**示例：**

```
GISDataQuery dq = new GISDataQuery(DatabaseType.SQLServer);  
dq.Open(ConfigurationSettings.AppSettings["CityConStr"]);
```

### 16.2 Open 方法

**public void Open ( string connectionString )****说明：**

打开数据库连接

**参数：**

connectionString: 数据库连接字符串

**返回值：**

无

**备注：**

示例代码设置了数据库类型为 SQLServer，根据设置链接数据库

**示例：**

```
GISDataQuery dq = new GISDataQuery(DatabaseType.SQLServer);  
dq.Open(ConfigurationSettings.AppSettings["CityConStr"]);
```

## 16.3 Close 方法

### **public void Close()**

说明:

关闭数据库连接

参数:

无

返回值:

无

备注:

查询完毕后关闭数据库连接

示例:

```
GISDataQuery dq = new GISDataQuery(DatabaseType.SQLServer);
dq.Open(ConfigurationSettings.AppSettings["CityConStr"])
.....
dq.Close();
dq.Dispose();
```

## 16.4 Dispose 方法

### **public void Dispose()**

说明:

释放数据库资源

参数:

无

返回值:

无

备注:

释放数据库资源

示例:

```
GISDataQuery dq = new GISDataQuery(DatabaseType.SQLServer);
dq.Open(ConfigurationSettings.AppSettings["CityConStr"])
.....
dq.Close();
dq.Dispose();
```

## 16.5 GetDataByHitting 方法

### **public System.Data.DataSet GetDataByHitting ( string strQuery,double x , double y , double offset )**

说明:

返回点击查询结果集

**参数:**

strQuery : 查询语句

x: x坐标

y: y坐标

offset: 点的偏移半径

**返回值:**

DataSet数据集

**备注:**

用于点击查询

**示例:** GISDataQuery dq = new GISDataQuery(DatabaseType.SQLServer);

dq.Open(ConfigurationSettings.AppSettings["CityConStr"]);

DataSet ds=new DataSet();

ds=dq.GetDataByHitting(strCmd,x,y,offset);

dq.Close();

dq.Dispose();

## 16.6 GetDataByCircle 方法

**public System.Data.DataSet GetDataByCircle ( string strQuery , double x , double y , double radius )****说明:**

返回圆查询结果集

**参数:**

strQuery : 查询语句

x: 圆心x坐标

y: 圆心y坐标

radius: 半径

**返回值:**

DataSet数据集

**备注:**

将圆查询结果返回

**示例:**

GISDataQuery dq = new GISDataQuery(DatabaseType.SQLServer);

dq.Open(ConfigurationSettings.AppSettings["CityConStr"]);

DataSet ds=new DataSet();

string strResultXml = "";

string strQuery = "SELECT PlaceName as N,X,Y FROM c027 WHERE GridCode in("+ gridcode +)";

ds = dq.GetDataByCircle(strQuery, Convert.ToDouble(paramArray[0]),

Convert.ToDouble(paramArray[1]), Convert.ToDouble(paramArray[2]));

strResultXml = ds.GetXml();

## 16.7 GetDataByRectangle 方法

**public System.Data.DataSet GetDataByRectangle ( string strQuery, double startX , double startY , double endX , double endY )**

**说明:**

返回矩形拉框查询结果集

**参数:**

strQuery: 查询语句

startX: 起点x坐标

startY: 起点y坐标

endX: 终点x坐标

endY: 终点y坐标

**返回值:**

DataSet数据集

**备注:**

用于矩形拉框，将结果集返回

**示例:**

```
GISDataQuery dq = new GISDataQuery(DatabaseType.SQLServer);
dq.Open(ConfigurationSettings.AppSettings["CityConStr"]);
DataSet ds=new DataSet();
string strResultXml = "";
string strQuery = "SELECT PlaceName as N,X,Y FROM c027 WHERE GridCode in("+
gridcode +")";
ds      =      dq.GetDataByRectangle(strQuery,      Convert.ToDouble(paramArray[0]),
Convert.ToDouble(paramArray[1]),      Convert.ToDouble(paramArray[2]),
Convert.ToDouble(paramArray[3]));
strResultXml = ds.GetXml();
```

## 16.8 GetDataByPolygon 方法

**public System.Data.DataSet GetDataByPolygon ( string strQuery, double[] Xarry , double[] Yarry )**

**说明:**

返回多边形查询结果集

**参数:**

strQuery : 查询语句

Xarry: x点坐标集

Yarry: y点坐标集

**返回值:**

DataSet数据集

**备注:**

用于多边形拉框，将结果集返回

**示例:**

```

GISDataQuery dq = new GISDataQuery(DatabaseType.SQLServer);
dq.Open(ConfigurationSettings.AppSettings["CityConStr"]);
DataSet ds=new DataSet();
string strResultXml = "";
string strQuery = "SELECT PlaceName as N,X,Y FROM c027 WHERE GridCode in("+
gridcode +")";
ds = dq. GetDataByPolygon (strQuery, paramArray[0], paramArray[1]);
strResultXml = ds.GetXml();

```

## 16.9 GetDataByCondition 方法

**public System.Data.DataSet GetDataByCondition (string strQuery)**

**说明:**

返回条件查询结果集

**参数:**

strQuery: 查询语句

**返回值:**

DataSet 数据集

**备注:**

根据关键词来查询

**示例:**

```

GISDataQuery dq = new GISDataQuery(DatabaseType.SQLServer);//定义为 SQL 数据库
dq.Open(ConfigurationSettings.AppSettings["CityConStr"]);
string strCmd = "SELECT PlaceName as N,X,Y from c027 WHERE ";
for (int i = 0; i < arryWords.Length; i++)
{
    if (arryWords[i].word != null)
        strCmd += " PlaceName like '%" + arryWords[i].word + "%' AND";
}
string strQuery = strCmd.Substring(0, strCmd.Length - 3);
string strResultXml = dq.GetDataByCondition(strQuery).GetXml();

```

## 16.10 IsDotInPolygon 方法

**public static bool IsDotInPolygon ( double[] PQXarray , double[] PQYarray , double XDot , double YDot )**

**说明:**

判断点是否在多边形中

**参数:**

PQXarray: 多边形x坐标集合

PQYarray: 多边形y坐标集合

XDot: 待定点x坐标

YDot: 待定点y坐标

**返回值:**

True, 点在多边形中; False, 点不在多边形中

**备注:**

用于多边形查询时结果集中的数据判断

**示例:**

```
string strCmd = strQuery;
SqlDataAdapter da = new SqlDataAdapter(strCmd, _con);
DataSet ds = new DataSet();
da.Fill(ds, "P");
foreach (DataRow row in ds.Tables["P"].Rows)
{
    double x = Convert.ToDouble(row["X"]);
    double y = Convert.ToDouble(row["Y"]);
    if(!checkDotInPolygon(Xarry, Yarry, x, y))
        row.Delete();
}
ds.AcceptChanges();
da.Dispose();
return ds;
```

## 17. GISSvrGenerator 类

访问GIS服务管理器以获取一个当前连接数最小的工作正常的GIS服务器的IP和服务端口或向其发送当前使用的GIS服务器IP地址和端口号

### 17.1 discon 方法

**public void discon()****说明:**

检测SOCKET

**参数:**

无

**返回值:**

无

**备注:**

隶属于GISSvrGenerator对象

### 17.2 getGisSvr 方法

**public System.Net.IPEndPoint getGisSvr(System.Net.IPEndPoint severAddress)****说明:**

根据传入参数(GIS 服务管理器 IP 地址和端口)向 GIS 服务管理器发送请求, 返回一个当

前具有最小连接数的 GIS 服务器的 IP 和服务端口

**参数:**

severAddress: GIS 服务管理器 IP 地址和端口

**返回值:**

当前具有最小连接数的 GIS 服务器的 IP 和服务端口

**备注:**

隶属于GISSvrGenerator对象

## 17.3 GISSvrGenerator 方法

### **public GISSvrGenerator ()**

**说明:**

构造函数

**参数:**

无

**返回值:**

无

**备注:**

隶属于GISSvrGenerator对象

## 17.4 sendGisSvr 方法

### **public bool sendGisSvr(System.Net.IPEndPoint GISServerMa, System.Net.IPEndPoint GISserverAdd)**

**说明:**

向 GIS 服务管理器发送当前连接的 GIS 服务器的 IP 和端口

**参数:**

GISServerMa: GIS 服务管理器地址

GISserverAdd: GIS 服务器地址

**返回值:**

布尔值

**备注:**

隶属于GISSvrGenerator对象

## 18. IMSOBJArray 类

对象数组类

## 属性

### 18.1 Count 属性

**Public int Count**

说明

获取记录个数

备注

该属性隶属于 IMSOBJArray 对象

## 方法

### 18.2 Append 方法

**Public int Append(Object obj)**

说明

添加记录。

参数

obj: 对象

返回值

大于 0 成功, 小于 0 失败

备注

该方法隶属于 IMSOBJArray 对象。

### 18.3 GetItem 方法

**public short GetItem(int index, out object pObj)**

说明

获取指定索引处的元素。

参数

Index: 元素索引(从零开始)

pObj: 返回对象

返回值

大于 0 成功, 小于 0 失败

备注

该方法隶属于 IMSOBJArray 对象。

### 18.4 IMSOBJArray 方法

**public IMSOBJArray ()**

说明

构造函数。

参数

无

返回值

无

**备注**

该方法隶属于 IMSOBJArray 对象。

## 18.5 RemoveAll 方法

**Public void RemoveAll()**

**说明**

移除所有元素。

**备注**

该方法隶属于 IMSOBJArray 对象。

## 18.6 RemoveAt 方法

**Public void RemoveAt(int index);**

**说明**

移除指定索引号的对象。

**参数**

index: 元素索引

**返回值**

无

**备注**

该方法隶属于 IMSOBJArray 对象。

## 19. OpCAddLayerInfo 类

添加图层信息;

### 19.1 DBName 属性

**Public String DBName**

**说明**

数据库名称

**备注**

该属性隶属于 OpCAddLayerInfo 对象

### 19.2 GDBName 属性

**Public String GDBName**

**说明**

GDB 名

**备注**

该属性隶属于 OpCAddLayerInfo 对象

### 19.3 LayerIndex 属性

**Public long LayerIndex**

说明

图层索引

备注

该属性隶属于 OpCAddLayerInfo 对象

### 19.4 LayerInformation 属性

**public MAPGISIMS.EnterpriseLib.OpCLayerInformation LayerInformation**

说明

图层信息

备注

该属性隶属于 OpCAddLayerInfo 对象

### 19.5 MapID 属性

**Public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCAddLayerInfo 对象

### 19.6 PassWord 属性

**Public string PassWord**

说明

密码

备注

该属性隶属于 OpCAddLayerInfo 对象

### 19.7 UserName 属性

**Public string UserName**

说明

用户名

备注

该属性隶属于 OpCAddLayerInfo 对象

## 20. OpCAppendAArcLine 类

添加线类

### 20.1 LayerIndex 属性

#### **Public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCAppendAArcLine 对象

示例

见 **CMapView** 类 **AppendOneArc** 方法。

### 20.2 LineColor 属性

#### **Public int LineColor**

说明

线颜色

备注

该属性隶属于 OpCAppendAArcLine 对象

示例

见 **CMapView** 类 **AppendOneArc** 方法。

### 20.2 LineRateX 属性

#### **Public float LineRateX**

说明

线比例 X 系数

备注

该属性隶属于 OpCAppendAArcLine 对象

示例

见 **CMapView** 类 **AppendOneArc** 方法。

### 20.3 LineRateY 属性

#### **Public float LineRateY**

说明

线比例 Y 系数

备注

该属性隶属于 OpCAppendAArcLine 对象

示例

见 **CMapView** 类 **AppendOneArc** 方法。

## 20.4 LineType 属性

**Public int LineType**

说明

线的型号

备注

该属性隶属于 OpCAppendAArcLine 对象

示例

见 **CMapView** 类 **AppendOneArc** 方法。

## 20.5 MapID 属性

**Public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCAppendAArcLine 对象

示例

见 **CMapView** 类 **AppendOneArc** 方法。

## 20.6 PenWidth 属性

**Public float PenWidth**

说明

外部笔宽

备注

该属性隶属于 OpCAppendAArcLine 对象

示例

见 **CMapView** 类 **AppendOneArc** 方法。

## 20.7 PropertyNumber 属性

**Public int PropertyNumber**

说明

图层属性字段个数

备注

该属性隶属于 OpCAppendAArcLine 对象

示例

见 **CMapView** 类 **AppendOneArc** 方法。

## 20.8 PropertyValues 属性

**Public string[] PropertyValues**

说明

图层属性值

备注

该属性隶属于 OpCAppendAArcLine 对象

示例

见 **CMapView** 类 **AppendOneArc** 方法。

## 20.9 TheArc 属性

**public MAPGISIMS.EnterpriseLib.OpCLine TheArc**

说明

线对象

备注

该属性隶属于 OpCAppendAArcLine 对象

示例

见 **CMapView** 类 **AppendOneArc** 方法。

## 21. OpCAppendAPoint 类

添加点类

### 21.1 LayerIndex 属性

**Public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCAppendAPoint 对象

示例

见 **CMapView** 类 **AppendOnePoint** 方法。

### 21.2 MapID 属性

**Public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCAppendAPoint 对象

示例

见 **CMapView** 类 **AppendOnePoint** 方法。

### 21.3 PicColor 属性

**Public int PicColor**

说明

子图颜色

备注

该属性隶属于 OpCAppendAPoint 对象

示例

见 **CMapView** 类 **AppendOnePoint** 方法。

### 21.4 PicType 属性

**Public int PicType**

说明

子图号

备注

该属性隶属于 OpCAppendAPoint 对象

示例

见 **CMapView** 类 **AppendOnePoint** 方法。

### 21.5 PicHeight 属性

**Public float PicHeight**

说明

子图高度

备注

该属性隶属于 OpCAppendAPoint 对象

示例

见 **CMapView** 类 **AppendOnePoint** 方法。

### 21.6 PicWidth 属性

**Public float PicWidth**

说明

子图宽度

**备注**

该属性隶属于 OpCAppendAPoint 对象

**示例**

见 **CMapView** 类 **AppendOnePoint** 方法。

## 21.7 PropertyNumber 属性

**Public int PropertyNumber**

**说明**

图层属性字段个数

**备注**

该属性隶属于 OpCAppendAPoint 对象

**示例**

见 **CMapView** 类 **AppendOnePoint** 方法。

## 21.8 PropertyValues 属性

**Public string[] PropertyValues**

**说明**

图层属性字段值，

**备注**

该属性隶属于 OpCAppendAPoint 对象

**示例**

见 **CMapView** 类 **AppendOnePoint** 方法。

## 21.9 ThePoint 属性

**public MAPGISIMS.EnterpriseLib.OpCPoint2D ThePoint**

**说明**

点对象

**备注**

该属性隶属于 OpCAppendAPoint 对象

**示例**

见 **CMapView** 类 **AppendOnePoint** 方法。

## 22. OpCAppendARegion 类

添加区类

## 22.1 LayerIndex 属性

### **Public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCAppendARegion 对象

示例

见 **CMapView** 类 **AppendOneRegion** 方法。

## 22.2 MapID 属性

### **Public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCAppendARegion 对象

示例

见 **CMapView** 类 **AppendOneRegion** 方法。

## 22.3 RegioniPic 属性

### **Public int RegioniPic**

说明

区填充图案

备注

该属性隶属于 OpCAppendAPoint 对象

示例

见 **CMapView** 类 **AppendOneRegion** 方法。

## 22.4 RegionColor 属性

### **Public int RegionColor**

说明

区填充颜色

备注

该属性隶属于 OpCAppendAPoint 对象

示例

见 **CMapView** 类 **AppendOneRegion** 方法。

## 22.5 PicColor 属性

### **Public int PicColor**

说明

图案颜色

备注

该属性隶属于 OpCAppendAPoint 对象

示例

见 **CMapView** 类 **AppendOneRegion** 方法。

## 22.6 PicHeight 属性

### **Public float PicHeight**

说明

图案高度

备注

该属性隶属于 OpCAppendAPoint 对象

示例

见 **CMapView** 类 **AppendOneRegion** 方法。

## 22.7 PicWidth 属性

### **Public float PicWidth**

说明

图案宽度

备注

该属性隶属于 OpCAppendAPoint 对象

示例

见 **CMapView** 类 **AppendOneRegion** 方法。

## 22.8 PropertyNumber 属性

### **Public int PropertyNumber**

说明

属性字段个数

备注

该属性隶属于 OpCAppendARegion 对象

示例

见 **CMapView** 类 **AppendOneRegion** 方法。

## 22.9 PropertyValues 属性

### **Public String[] PropertyValues**

#### 说明

设置要添加区要素的属性字段值，数组个数由 PropertyNumber 属性决定

#### 备注

该属性隶属于 OpCAppendARegion 对象

#### 示例

见 CMapView 类 AppendOneRegion 方法。

## 22.10 TheRegion 属性

### **public MAPGISIMS.EnterpriseLib.OpCRegion TheRegion**

#### 说明

区上线对象数组

#### 备注

该属性隶属于 OpCAppendARegion 对象

#### 示例

见 CMapView 类 AppendOneRegion 方法。

## 23. OpcAppendOperResult 类

添加点、线、区要素返回值类

## 24. OpCAttSelectStru 类

属性条件查询类

### 24.1 AttributeCondition 属性

#### **Public string AttributeCondition**

#### 说明

条件语句

#### 备注

该属性隶属于 OpCAttSelectStru 对象

#### 示例

见 CMapView 类 SelectByCondition 方法。

## 24.2 LayerIndex 属性

### **Public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCAttSelectStru 对象

示例

见 **CMapView** 类 **SelectByCondition** 方法。

## 24.3 MapID 属性

### **Public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCAttSelectStru 对象

示例

见 **CMapView** 类 **SelectByCondition** 方法。

## 24.4 PageIndex 属性

### **Public long PageIndex**

说明

起始记录的索引

备注

该属性隶属于 OpCAttSelectStru 对象

示例

见 **CMapView** 类 **SelectByCondition** 方法。

## 24.5 recordCountOfPage 属性

### **Public int recordCountOfPage**

说明

查询操作返回的每页的记录条数

备注

该属性隶属于 OpCAttSelectStru 对象

示例

见 **CMapView** 类 **SelectByCondition** 方法。

## 25. OpCBasicNetAnalysis 类

网络分析类

### 25.1 LayerIndex 属性

**Public long LayerIndex**

说明

图层索引

备注

该属性隶属于 OpCBasicNetAnalysis 对象

示例

见 CMapView 类 NetAnalysis 方法。

### 25.2 NodeNumber 属性

**Public int NodeNumber**

说明

节点个数

备注

该属性隶属于 OpCBasicNetAnalysis 对象

示例

见 CMapView 类 NetAnalysis 方法。

### 25.3 Radius 属性

**Public double Radius**

说明

半径

备注

该属性隶属于 OpCBasicNetAnalysis 对象

示例

见 CMapView 类 NetAnalysis 方法。

### 25.4 Nodes 属性

**public MAPGISIMS.EnterpriseLib.OpCPoint2D[] Nodes**

说明

节点数组

备注

该属性隶属于 OpCBasicNetAnalysis 对象

示例

见 **CMapView** 类 **NetAnalysis** 方法。

## 25.5 MapID 属性

**public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCBasicNetAnalysis 对象

示例

见 **CMapView** 类 **NetAnalysis** 方法。

## 25.6 barNum 属性

**public long barNum**

说明

障碍点数目

备注

该属性隶属于 OpCBasicNetAnalysis 对象

示例

见 **CMapView** 类 **NetAnalysis** 方法。

## 25.7 barPoints 属性

**public MAPGISIMS.EnterpriseLib.OpCPoint2D[] barPoints**

说明

障碍点列表

备注

该属性隶属于 OpCBasicNetAnalysis 对象

示例

见 **CMapView** 类 **NetAnalysis** 方法。

## 26. OpCCircleSelectClass 类

圆查询类

## 26.1 Center 属性

**public MAPGISIMS.EnterpriseLib.OpCPoint2D Center**

说明

圆心坐标

备注

该属性隶属于 OpCCircleSelectClass 对象

示例

见 **CMapView** 类 **SelectByCircle** 方法。

## 26.2 LayerIndex 属性

**public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCCircleSelectClass 对象

示例

见 **CMapView** 类 **SelectByCircle** 方法。

## 26.3 Radius 属性

**public double Radius**

说明

圆查询半径

备注

该属性隶属于 OpCCircleSelectClass 对象

示例

见 **CMapView** 类 **SelectByCircle** 方法。

## 26.4 MapID 属性

**public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCCircleSelectClass 对象

示例

见 **CMapView** 类 **SelectByCircle** 方法。

## 26.5 PageIndex 属性

**public long PageIndex**

说明

查询结果当前页的第一条记录的索引

备注

该属性隶属于 OpCCircleSelectClass 对象

示例

见 CMapView 类 SelectByCircle 方法。

## 26.6 recordCountOfPage 属性

**Public int recordCountOfPage**

说明

查询操作返回的每页的记录条数

备注

该属性隶属于 OpCCircleSelectClass 对象

## 27. OpCCircleWithAttSelectStru 类

圆条件查询类

### 27.1 Question 属性

**Public string Question**

说明

查询条件

备注

该属性隶属于 OpCCircleWithAttSelectStru 对象

示例

见 CMapView 类 SelectByCondition 方法。

## 28. OpCDeleteFeature 类

删除点、线、区要素类

### 28.1 DelGeometry 属性

**Public int DelGeometry**

说明

删除类型

备注

该属性隶属于 OpCDeleteFeature 对象

示例

见 **CMapView** 类 **DeleteOneFeature** 方法。

## 28.2 FeatureID 属性

**Public long FeatureID**

说明

要素号

备注

该属性隶属于 OpCDeleteFeature 对象

示例

见 **CMapView** 类 **DeleteOneFeature** 方法。

## 28.3 LayerIndex 属性

**Public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCDeleteFeature 对象

示例

见 **CMapView** 类 **DeleteOneFeature** 方法。

## 28.4 MapID 属性

**Public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCDeleteFeature 对象

示例

见 **CMapView** 类 **DeleteOneFeature** 方法。

## 29. OpCEagleEye 类

鹰眼类

## 29.1 EagleRect 属性

**public MAPGISIMS.EnterpriseLib.OpCRect EagleRect**

说明

设置鹰眼图像的显示范围

备注

该属性隶属于 OpCEagleEye 对象

## 29.2 Height 属性

**public ushort Height**

说明

设置鹰眼图像高

备注

该属性隶属于 OpCEagleEye 对象

## 29.3 ImageFileName 属性

**public string ImageFileName**

说明

设置要生成的鹰眼图像文件名

备注

该属性隶属于 OpCEagleEye 对象

## 29.4 MapID 属性

**public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCEagleEye 对象

## 29.5 Rate 属性

**Public double Rate**

说明

设置鹰眼图像与主图的显示比例

备注

该属性隶属于 OpCEagleEye 对象

## 29.6 Width 属性

**public ushort Width**

说明

设置鹰眼图像宽

备注

该属性隶属于 OpCEagleEye 对象

## 30. OpCEagleEyeResult 类

获取鹰眼图像类

### 30.1 ImageFileName 属性

**Public string ImageFileName**

说明

获取生成的鹰眼图像名称

备注

该属性隶属于 OpCEagleEyeResult 对象

### 30.2 ImageSize 属性

**Public int ImageSize**

说明

图像大小

备注

该属性隶属于 OpCEagleEyeResult 对象

## 31. OpCFeatureGeometry 类

要素几何信息类

### 31.1 PolygonNumber 属性

**Public int PolygonNumber**

说明

要素含有多边形个数

备注

该属性隶属于 OpCFeatureGeometry 对象

示例

见 **CMapView** 类 **GetFeatureGeometry** 方法。

## 31.2 ArcNumber 属性

**Public int ArcNumber**

说明

要素含有弧线个数

备注

该属性隶属于 OpCFeatureGeometry 对象

示例

见 **CMapView** 类 **GetFeatureGeometry** 方法。

## 31.3 PointNumber 属性

**Public int PointNumber**

说明

要素含有点个数

备注

该属性隶属于 OpCFeatureGeometry 对象

示例

见 **CMapView** 类 **GetFeatureGeometry** 方法。

## 31.4 Polygons 属性

**public MAPGISIMS.EnterpriseLib.OpCRegion[ ] Polygons**

说明

多边形对象数组

备注

该属性隶属于 OpCFeatureGeometry 对象

## 31.5 ArcLines 属性

**public MAPGISIMS.EnterpriseLib.OpCMultiLine[ ] ArcLines**

说明

复杂线数组

备注

该属性隶属于 OpCFeatureGeometry 对象

## 31.6 Points 属性

**public MAPGISIMS.EnterpriseLib.OpCPoint2D[] Points**

说明

点数组

备注

该属性隶属于 OpCFeatureGeometry 对象

示例

见 CMapView 类 GetFeatureGeometry 方法。

## 32. OpCFeatureStyleInfo 类

要素样式属性类

### 32.1 Color 属性

**public int Color**

说明

颜色 (点: 子图颜色; 线: 线颜色; 区: 填充颜色)

备注

该属性隶属于 OpCFeatureStyleInfo 对象

### 32.2 FeatureType 属性

**public int FeatureType**

说明

要素类型

备注

该属性隶属于 OpCFeatureStyleInfo 对象

### 32.3 MarginHeight 属性

**public float MarginHeight**

说明

高 (点: 子图高; 线: 设置无效; 区: 填充图案高)

备注

该属性隶属于 OpCFeatureStyleInfo 对象

## 32.4 MarginWidth 属性

### **public float MarginWidth**

说明

宽 (点: 子图宽; 线: 外部笔宽; 区: 填充图案宽)

备注

该属性隶属于 OpCFeatureStyleInfo 对象

## 32.5 PatternIndex 属性

### **public int PatternIndex**

说明

图案号 (点: 子图号; 线: 线型号; 区: 填充图案号)

备注

该属性隶属于 OpCFeatureStyleInfo 对象

## 33. OpCGeomInfo 类

添加、删除、更新要素统一返回值类

### 33.1 LayerIndex 属性

#### **Public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCGeomInfo 对象

示例

见 CMapView 类 GetGeomInfo 方法。

### 33.2 FLAG 属性

#### **Public Short FLAG**

说明

操作标记

备注

该属性隶属于 OpCGeomInfo 对象

示例

见 CMapView 类 GetGeomInfo 方法。

### 33.3 MapID 属性

**Public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCGeomInfo 对象

示例

见 CMapView 类 GetGeomInfo 方法。

### 34. OpCGetAttStructure 类

获取属性结构类

#### 34.1 LayerIndex 属性

**Public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCGetAttStructure 对象

#### 34.2 MapID 属性

**Public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCGetAttStructure 对象

### 35. OpCGetFeatureInformation 类

获取要素信息类

#### 35.1 FeatureID 属性

**public long FeatureID**

说明

要素号

**备注**

该属性隶属于 OpCGetFeatureInformation 对象

**示例**

见 **CMapView** 类 **GetFeatureGeometry** 方法。

## 35.2 InformationType 属性

**Public int InformationType****说明**

要素信息类型

**备注**

该属性隶属于 OpCGetFeatureInformation 对象

**示例**

见 **CMapView** 类 **GetFeatureGeometry** 方法。

## 35.3 LayerIndex 属性

**Public long LayerIndex****说明**

图层索引号

**备注**

该属性隶属于 OpCGetFeatureInformation 对象

**示例**

见 **CMapView** 类 **GetFeatureGeometry** 方法。

## 35.4 MapID 属性

**Public long MapID****说明**

地图 ID

**备注**

该属性隶属于 OpCGetFeatureInformation 对象

**示例**

见 **CMapView** 类 **GetFeatureGeometry** 方法。

## 36. OpCGetFeatureStyleInfo 类

获取要素附加信息类，如颜色、高宽等

## 36.1 FeatureID 属性

**public long FeatureID**

说明

要素号

备注

该属性隶属于 OpCGetFeatureStyleInfo 对象

## 36.2 LayerIndex 属性

**Public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCGetFeatureStyleInfo 对象

## 36.3 MapID 属性

**Public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCGetFeatureStyleInfo 对象

## 37. OpCGetGDBInfo 类

获取 GDB 信息

### 37.1 MapID 属性

**Public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCGetGDBInfo 对象

示例

见 CMapView 类 GetGDBInfo 方法。

## 38. OpCGetGDBLayerInfo 类

获取 GDB 信息

### 38.1 MapID 属性

#### **Public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCGetGDBLayerInfo 对象

示例

见 **CMapView** 类 **GetGDBLayerInfo** 方法。

### 38.2 DataSource 属性

#### **Public string DataSource**

说明

GDB 数据源名称，如：MapGisLocal

备注

该属性隶属于 OpCGetGDBLayerInfo 对象

示例

见 **CMapView** 类 **GetGDBLayerInfo** 方法。

### 38.3 GDBName 属性

#### **Public string GDBName**

说明

GDB 名称，如 ChinaTest

备注

该属性隶属于 OpCGetGDBLayerInfo 对象

示例

见 **CMapView** 类 **GetGDBLayerInfo** 方法。

### 38.4 Password

#### **Public string password**

说明

网络数据密码

备注

该属性隶属于 OpCGetGDBLayerInfo 对象

示例

见 **CMapView** 类 **GetGDBLayerInfo** 方法。

## 38.5 userName

**Public string userName**

说明

网络数据用户名

备注

该属性隶属于 OpCGetGDBLayerInfo 对象

示例

见 **CMapView** 类 **GetGDBLayerInfo** 方法。

## 39. OpCGetGDBName 类

获取 GDB 名称类

### 39.1 MapID 属性

**Public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCGetGDBName 对象

示例

见 **CMapView** 类 **GetGDBName** 方法。

### 39.2 DataSource 属性

**Public String DataSource**

说明

GDB 数据源名称

备注

该属性隶属于 OpCGetGDBName 对象

示例

见 **CMapView** 类 **GetGDBName** 方法。

### 39.3 Password

**Public string password**

说明

网络数据密码

备注

该属性隶属于 OpCGetGDBName 对象

示例

见 **CMapView** 类 **GetGDBName** 方法。

### 39.4 userName

**Public string userName**

说明

网络数据用户名

备注

该属性隶属于 OpCGetGDBName 对象

示例

见 **CMapView** 类 **GetGDBName** 方法。

## 40. OpCGetMap 类

获取地图图像类

### 40.1 DisplayRectangle 属性

**public MAPGISIMS.EnterpriseLib.OpCRect DisplayRectangle**

说明

地图显示范围

备注

该属性隶属于 OpCGetMap 对象

### 40.2 FileType 属性

**public int FileType**

说明

地图图像文件的类型

备注

该属性隶属于 OpCGetMap 对象

### 40.3 Height 属性

**public ushort Height**

说明

地图在客户端显示的高

备注

该属性隶属于 OpCGetMap 对象

### 40.4 ImageFileName 属性

**public string ImageFileName**

说明

图像文件名

备注

该属性隶属于 OpCGetMap 对象

### 40.5 LayerCount 属性

**public int LayerCount**

说明

子图层个数

备注

该属性隶属于 OpCGetMap 对象

### 40.6 LayerInfoTree 属性

**public MAPGISIMS.EnterpriseLib.OpCLayerInformation[] LayerInfoTree**

说明

各个图层信息

备注

该属性隶属于 OpCGetMap 对象

### 40.7 MapId 属性

**public long MapId**

说明

地图 ID

备注

该属性隶属于 OpCGetMap 对象

## 40.8 Original 属性

### **Public int Original**

说明

地图是否还原显示

备注

该属性隶属于 OpCGetMap 对象

## 40.9 Transparent 属性

### **public int Transparent**

说明

地图是否透明显示

备注

该属性隶属于 OpCGetMap 对象

## 40.10 Width 属性

### **Public ushort Width**

说明

地图在客户端显示的宽

备注

该属性隶属于 OpCGetMap 对象

## 40.11 bufferShow

### **public int bufferShow**

说明

是否缓冲显示 0: 否 1: 是

备注

该属性隶属于 OpCGetMap 对象

## 40.12 dynamicNote

### **public int dynamicNote**

说明

是否显示动态标注

备注

该属性隶属于 OpCGetMap 对象

## 40.13 followScale

### **public int followScale**

说明

是否跟随比例显示 0: 否 1: 是

备注

该属性隶属于 OpCGetMap 对象

## 40.14 imageQuality

### **public int imageQuality**

说明

图像质量 1: 低 2: 中 3: 高

备注

该属性隶属于 OpCGetMap 对象

## 40.15 showCoordinate

### **public int showCoordinate**

说明

是否显示坐标点 0: 否 1: 是

备注

该属性隶属于 OpCGetMap 对象

## 40.16 showScale

### **public int showScale**

说明

是否显示标尺 0: 否 1: 是

备注

该属性隶属于 OpCGetMap 对象

## 41. OpCGetMapLst 类

获取 GIS 服务上的地图文档列表

### 41.1 flag 属性

#### **public short flag**

说明

要素个数

备注

该属性隶属于 OpCGetMapLst 对象

## 42. OpCGetMapLstResult 类

获取 GIS 服务上的地图文档列表结果类

### 42.1 mapDocList 属性

**public string[] mapDocList**

说明

地图文档

备注

该属性隶属于 OpCGetMapLstResult 对象

### 42.2 mapDocNum 属性

**public int mapDocNum**

说明

要素个数

备注

该属性隶属于 OpCGetMapLstResult 对象

## 43. OpCGetMapResult 类

获取地图图像类

### 43.1 ImageFileName 属性

**public string ImagefileName**

说明

地图图像文件名

备注

该属性隶属于 OpCGetMapResult 对象

## 43.2 ImageSize 属性

**public int ImageSize**

说明

地图图像大小

备注

该属性隶属于 OpCGetMapResult 对象

## 44. OpCGetNewMapInfo 类

重新获取地图文档信息

### 44.1 MapID 属性

**Public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCGetNewMapInfo 对象

## 45. OpCGetOnePage 类

分页类

### 45.1 AttSelect 属性

**public MAPGISIMS.EnterpriseLib.OpCAttSelectStru AttSelect**

说明

条件查询

备注

该属性隶属于 OpCGetOnePage 对象

示例

见 CMapView 类 GetOnePageResult 方法。

### 45.2 CircleSelect 属性

**public MAPGISIMS.EnterpriseLib.OpCCircleSelectClass CircleSelect**

说明

圆查询对象

备注

该属性隶属于 OpCGetOnePage 对象

示例

见 **CMapView** 类 **GetOnePageResult** 方法。

### 45.3 CircleWithAttSelect 属性

**public MAPGISIMS.EnterpriseLib.OpCCircleWithAttSelectStru CircleWithAttSelect**

说明

圆条件查询对象

备注

该属性隶属于 OpCGetOnePage 对象

示例

见 **CMapView** 类 **GetOnePageResult** 方法。

### 45.4 LayerIndex 属性

**Public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCGetOnePage 对象

示例

见 **CMapView** 类 **GetOnePageResult** 方法。

### 45.5 LineSelect 属性

**public MAPGISIMS.EnterpriseLib.OpCLineSelectClass LineSelect**

说明

线查询对象

备注

该属性隶属于 OpCGetOnePage 对象

示例

见 **CMapView** 类 **GetOnePageResult** 方法。

### 45.6 PntSelect 属性

**public MAPGISIMS.EnterpriseLib.OpCPointSelectClass PntSelect**

说明

点击查询对象

备注

该属性隶属于 OpCGetOnePage 对象

示例

见 **CMapView** 类 **GetOnePageResult** 方法。

## 45.7 PntWithAttSelect 属性

**public MAPGISIMS.EnterpriseLib.OpCPointWithAttSelectStru PntWithAttSelect**

说明

点击条件查询对象

备注

该属性隶属于 OpCGetOnePage 对象

示例

见 **CMapView** 类 **GetOnePageResult** 方法。

## 45.8 PolygonSelect 属性

**public MAPGISIMS.EnterpriseLib.OpCPolygonSelectClass PolygonSelect**

说明

多边形查询对象

备注

该属性隶属于 OpCGetOnePage 对象

示例

见 **CMapView** 类 **GetOnePageResult** 方法。

## 45.9 PolygonWithAttSelect 属性

**public MAPGISIMS.EnterpriseLib.OpCPolygonWithAttSelectStru PolygonWithAttSelect**

说明

多边形条件查询对象

备注

该属性隶属于 OpCGetOnePage 对象

示例

见 **CMapView** 类 **GetOnePageResult** 方法。

## 45.10 RecSelect 属性

**public MAPGISIMS.EnterpriseLib.OpCRectangleSelectClass RecSelect**

说明

矩形查询对象

备注

该属性隶属于 OpCGetOnePage 对象

示例

见 **CMapView** 类 **GetOnePageResult** 方法。

## 45.11 RectWintAttSelect 属性

**public MAPGISIMS.EnterpriseLib.OpCRectWithAttSelectStru RectWintAttSelect**

说明

矩形条件查询对象

备注

该属性隶属于 OpCGetOnePage 对象

示例

见 **CMapView** 类 **GetOnePageResult** 方法。

## 45.12 SelectType 属性

**public MAPGISIMS.EnterpriseLib.QueryTypeEnum SelectType**

说明

查询类型

备注

该属性隶属于 OpCGetOnePage 对象

示例

见 **CMapView** 类 **GetOnePageResult** 方法。

## 45.13 StartIndex 属性

**public long StartIndex**

说明

开始页数

备注

该属性隶属于 OpCGetOnePage 对象

示例

见 **CMapView** 类 **GetOnePageResult** 方法。

## 46.OpCLayerAttStructure 类

属性结构类

## 46.1 FieldNumber 属性

### **Public int FieldNumber**

说明

属性字段个数

备注

该属性隶属于 OpCLayerAttStructure 对象

示例

见 **CMapView** 类 **GetFeaturelayerAttributeStructure** 方法。

## 46.2 FieldTypes 属性

### **Public int[] FieldTypes**

说明

属性字段类型

备注

该属性隶属于 OpCLayerAttStructure 对象

示例

见 **CMapView** 类 **GetFeaturelayerAttributeStructure** 方法。

## 46.3 FieldNames 属性

### **Public string[] FieldNames**

说明

属性字段名

备注

该属性隶属于 OpCLayerAttStructure 对象

示例

见 **CMapView** 类 **GetFeaturelayerAttributeStructure** 方法。

## 46.4 FieldLength 属性

### **Public ushort[] FieldLengths**

说明

属性字段长度

备注

该属性隶属于 OpCLayerAttStructure 对象

示例

见 **CMapView** 类 **GetFeaturelayerAttributeStructure** 方法。

## 47. OpCLayerInformation 类

图层信息

### 47.1 ChildLayerCount 属性

#### Public int ChildLayerCount

说明

包含子图层的个数

备注

该属性隶属于 OpCLayerInformation 对象

示例

```
//得到地图文档中图层的个数
sLayerNum = viewport.MapObj.GetLayerNum(out LayerNum);
for(int i=0;i<LayerNum;i++)
{
    //依次向集合中添加图层
    rdat.Append(i);
    //依次得到图层对象
    sItem = viewport.MapObj.GetItem(rdat,out pMapLayer);
    if(sItem==1)//对象生成成功
    {
        strLayerNo = i.ToString();
        string LayerName = pMapLayer.LayerName;//得到图层名称
        //得到该图层下子图层个数
        short LayerCount = pMapLayer.ChildLayerCount;
        Count++;
        //递归显示子结点
        DispChild(LayerCount,rdat,strLayerNo);
    }
    rdat.RemoveAll();
}
```

### 47.2 ChildLayers 属性

#### public MAPGISIMS.EnterpriseLib.OpCLayerInformation[] ChildLayers

说明

子图层

备注

该属性隶属于 OpCLayerInformation 对象

### 47.3 DynamicNote 属性

#### **Public short DynamicNote**

说明

是否动态标注标志

备注

该属性隶属于 OpCLayerInformation 对象

### 47.4 Editable 属性

#### **Public short Editable**

说明

图层可编辑标志

0: 不可编辑

1: 可编辑

备注

该属性隶属于 OpCLayerInformation 对象

示例

```
//得到地图文档中图层个数
viewport.MapObj.GetLayerNum(out LayerNum);

//设置地图文档中所有图层不可编辑
for(short i=0;i<LayerNum;i++)
{
    LayerInformation layerObj = null;
    getMapLayerObj(i.ToString(),out layerObj);
    //设置图层不可编辑
    layerObj.Editable = 0;
    viewport.UpdateLayer(layerObj);
}
```

### 47.5 FeatureClass 属性

#### **Public int FeatureClass**

说明

要素类对象

备注

该属性隶属于 OpCLayerInformation 对象

### 47.6 FeatureClassID 属性

#### **Public int FeatureClassID**

说明

要素类的 ID

备注

该属性隶属于 OpCLayerInformation 对象

## 47.7 FieldName 属性

**Public string FieldName**

说明

标注字段的名称

备注

该属性隶属于 OpCLayerInformation 对象

## 47.8 Font 属性

**Public short Font**

说明

标注字体

备注

该属性隶属于 OpCLayerInformation 对象

## 47.9 FontColor 属性

**Public long FontColor**

说明

标注颜色

备注

该属性隶属于 OpCLayerInformation 对象

## 47.10 FontSize 属性

**Public short FontSize**

说明

标注字高

备注

该属性隶属于 OpCLayerInformation 对象

## 47.11 LayerBound 属性

**public MAPGISIMS.EnterpriseLib.OpCRect LayerBound**

说明

图层的显示范围

备注

该属性隶属于 OpCLayerInformation 对象

## 47.12 LayerIndex 属性

**Public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCLayerInformation 对象

## 47.13 LayerName 属性

**Public string LayerName**

说明

图层名称

备注

该属性隶属于 OpCLayerInformation 对象

## 47.14 LayerState 属性

**Public short LayerState**

说明

图层状态

备注

该属性隶属于 OpCLayerInformation 对象

## 47.15 LayerType 属性

**Public short LayerType**

说明

图层类型

备注

该属性隶属于 OpCLayerInformation 对象

## 47.16 MaxScale 属性

**Public double MaxScale**

说明

最大显示比例

备注

该属性隶属于 OpCLayerInformation 对象

## 47.17 MinScale 属性

**Public double MaxScale**

说明

最小显示比例

备注

该属性隶属于 OpCLayerInformation 对象

## 47.18 NoteMode 属性

**Public short NoteMode**

说明

标注模式

备注

该属性隶属于 OpCLayerInformation 对象

## 47.19 ScaleSymbolFlag 属性

**Public Double ScaleSymbolFlag**

说明

标尺符号标志

备注

该属性隶属于 OpCLayerInformation 对象

## 47.20 Selectable 属性

**Public short Selectable**

说明

图层可查询标志

0: 不可查询

1: 可查询

备注

该属性隶属于 OpCLayerInformation 对象

示例

```
//得到地图文档中图层个数
viewport.MapObj.GetLayerNum(out LayerNum);

//设置地图文档中所有图层不可查询
for(short i=0;i<LayerNum;i++)
{
    LayerInformation layerObj = null;
    getMapLayerObj(i.ToString(),out layerObj);
    //设置图层不可查询
    layerObj.Selectable = 0;
    viewport.UpdateLayer(layerObj);
}
```

## 47.21 Valid 属性

### Public short Valid

说明

图层是否有效打开标记：1—打开；0—未打开

备注

该属性隶属于 OpCLayerInformation 对象

## 47.22 Visible 属性

### Public short Visible

说明

图层是否可见标志

0: 不可见

1: 可见

备注

该属性隶属于 OpCLayerInformation 对象

示例

```
//得到地图文档中图层个数
viewport.MapObj.GetLayerNum(out LayerNum);

//设置地图文档中所有图层不可见
for(short i=0;i<LayerNum;i++)
{
    LayerInformation layerObj = null;
    getMapLayerObj(i.ToString(),out layerObj);
    //设置图层不可见
    layerObj.Visible = 0;
    viewport.UpdateLayer(layerObj);
}
```

## 48. OpCLayerSelectResult 类

图层查询结果类;

### 48.1 Count 属性

#### **Public ushort Count**

说明

该图层上查询结果的总个数

备注

该属性隶属于 OpCLayerSelectResult 对象

### 48.2 FeatureIDs 属性

#### **Public long[] FeatureIDs**

说明

查询到的要素类的 ID

备注

该属性隶属于 OpCLayerSelectResult 对象

### 48.3 FieldNames 属性

#### **Public string[] FieldNames**

说明

查询结果的属性字段名

备注

该属性隶属于 OpCLayerSelectResult 对象

### 48.4 FieldNumber 属性

#### **Public int FieldNumber**

说明

查询结果的属性字段数

备注

该属性隶属于 OpCLayerOpCSelectResult 对象

### 48.5 LayerIndex 属性

#### **Public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCLayerSelectResult 对象

## 48.6 PropertyValues 属性

**public System.Collections.ArrayList<T> PropertyValues**

说明

查询结果的属性值

备注

该属性隶属于 OpCLayerSelectResult 对象

## 49. OpCLine 类

简单线

### 49.1 PointNumber 属性

**Public int PointNumber**

说明

线上的点数

备注

该属性隶属于 OpCLine 对象

示例

见 CMapView 类 AppendOneArc 方法。

### 49.2 ThePoints 属性

**public MAPGISIMS.EnterpriseLib.OpCPoint2D[] ThePoints**

说明

点对象数组

备注

该属性隶属于 OpCLine 对象

示例

见 CMapView 类 AppendOneArc 方法。

## 50. OpCLineSelectClass 类

线查询类

## 50.1 LayerIndex 属性

**public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCLineSelectClass 对象

示例

见 CMapView 类 SelectByLine 方法。

## 50.2 LineRadius 属性

**public double LineRadius**

说明

线查询误差半径

备注

该属性隶属于 OpCLineSelectClass 对象

示例

见 CMapView 类 SelectByLine 方法。

## 50.3 SelectLine 属性

**public MAPGISIMS.EnterpriseLib.OpCLine SelectLine**

说明

线对象

备注

该属性隶属于 OpCLineSelectClass 对象

示例

见 CMapView 类 SelectByLine 方法。

## 51. OpCLoadMap 类

加载地图类

### 51.1 MapName 属性

Public String MapName

说明

地图名

备注

该属性隶属于 OpCLoadMap 对象

## 52. OpCLoadMapResult 类

地图返回结果类

### 52.1 LayerInfoTree 属性

**public MAPGISIMS.EnterpriseLib.OpCLayerInformation[] LayerInfoTree**

说明

子图层信息

备注

该属性隶属于 OpCLoadMapResult 对象

### 52.2 LayerNumber 属性

**Public int LayerNumber**

说明

指示图层树的图层个数

备注

该属性隶属于 OpCLoadMapResult 对象

### 52.3 MapBound 属性

**public MAPGISIMS.EnterpriseLib.OpCRect MapBound**

说明

地图的显示范围

备注

该属性隶属于 OpCLoadMap 对象

### 52.4 MapID 属性

**Public int MapID**

说明

地图 ID

备注

该属性隶属于 OpCLoadMap 对象

## 53. OpCMultiLine 类

复杂线类

### 53.1 LineNumber 属性

**public ushort LineNumber**

说明

复杂线上包含的线数

备注

该属性隶属于 OpCMultiLine 对象

### 53.2 TheLines 属性

**public MAPGISIMS.EnterpriseLib.OpCLine[] TheLines**

说明

线对象数组

备注

该属性隶属于 OpCMultiLine 对象

## 54. OpCNetAnalysisResult 类

网络分析返回值类

### 54.1 Start 属性

**public MAPGISIMS.EnterpriseLib.OpCPoint2D Start**

说明

开始节点

备注

该属性隶属于 OpCNetAnalysisResult 对象

示例

见 **CMapView** 类 **NetAnalysis** 方法。

### 54.2 End 属性

**public MAPGISIMS.EnterpriseLib.OpCPoint2D End**

说明

终节点

**备注**

该属性隶属于 OpCNetAnalysisResult 对象

**示例**

见 **CMapView** 类 **NetAnalysis** 方法。

### 54.3 InnerNodeNumber 属性

**public short InnerNodeNumber****说明**

节点个数

**备注**

该属性隶属于 OpCNetAnalysisResult 对象

**示例**

见 **CMapView** 类 **NetAnalysis** 方法。

### 54.4 InnerNodes 属性

**public MAPGISIMS.EnterpriseLib.OpCPoint2D[ ] InnerNodes****说明**

节点数组

**备注**

该属性隶属于 OpCNetAnalysisResult 对象

**示例**

见 **CMapView** 类 **NetAnalysis** 方法。

### 54.5 PathNumber 属性

**public short PathNumber****说明**

路径个数

**备注**

该属性隶属于 OpCNetAnalysisResult 对象

**示例**

见 **CMapView** 类 **NetAnalysis** 方法。

### 54.6 Paths 属性

**public MAPGISIMS.EnterpriseLib.OpCLine[ ] Paths****说明**

网络分析路径

**备注**

该属性隶属于 OpCNetAnalysisResult 对象

示例

见 **CMapView** 类 **NetAnalysis** 方法。

## 54.7 SResult 属性

**public MAPGISIMS.EnterpriseLib.OpCSelectResult Sresult**

说明

查询结果

备注

该属性隶属于 OpCNetAnalysisResult 对象

示例

见 **CMapView** 类 **NetAnalysis** 方法。

## 54.8 layerSelectResults 属性

**public MAPGISIMS.EnterpriseLib.OpCLayerSelectResult[ ] layerSelectResults**

说明

网络分析结果数组，第一个是点，第二个是线

备注

该属性隶属于 OpCNetAnalysisResult 对象

示例

见 **CMapView** 类 **NetAnalysis** 方法。

## 55.OpCOverlayByIDPntSeq 类

要素叠加分析结果类（点序列）

### 55.1 iFeatureNum 属性

**public int iFeatureNum**

说明

要素个数

备注

该属性隶属于 OpCOverlayByIDPntSeq 对象

### 55.2 pInfo 属性

**public MAPGISIMS.EnterpriseLib.OpCFeatureGeometry[ ] pInfo**

说明

要素类数组

**备注**

该属性隶属于 OpCOverlayByIDPntSeq 对象

## 56. OpCPoint3D 类

三维点

### 56.1 X 属性

**public double X**

**说明**

三维点 X 坐标

**备注**

该属性隶属于 OpCPoint3D 对象

### 56.2 Y 属性

**public double Y**

**说明**

三维点 Y 坐标

**备注**

该属性隶属于 OpCPoint3D 对象

### 56.3 Z 属性

**public double Z**

**说明**

三维点 Z 坐标

**备注**

该属性隶属于 OpCPoint3D 对象

## 57. OpCPointSelectClass 类

点击查询类

### 57.1 Distance 属性

**public double Distance**

说明

误差半径

备注

该属性隶属于 OpCPointSelectClass 对象

示例

见 **CMapView** 类 **SelectByHitting** 方法。

## 57.2 Dot 属性

**public MAPGISIMS.EnterpriseLib.OpCPoint2D Dot**

说明

点击点二维坐标对象

备注

该属性隶属于 OpCPointSelectClass 对象

示例

见 **CMapView** 类 **SelectByHitting** 方法。

## 57.3 LayerIndex 属性

**public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCPointSelectClass 对象

示例

见 **CMapView** 类 **SelectByHitting** 方法。

## 57.4 MapID 属性

**public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCPointSelectClass 对象

示例

见 **CMapView** 类 **SelectByHitting** 方法。

## 57.5 PageIndex 属性

**public long PageIndex**

说明

查询结果当前页的第一条记录的索引

备注

该属性隶属于 OpCPointSelectClass 对象

示例

见 CMapView 类 SelectByHitting 方法。

## 57.6 recordCountOfPage 属性

**Public int recordCountOfPage**

说明

查询操作返回的每页的记录条数

备注

该属性隶属于 OpCPointSelectClass 对象

示例

见 CMapView 类 SelectByHitting 方法。

## 58. OpCPointWithAttSelectStru 类

点击条件查询类

### 58.1 Question 属性

**public string Question**

说明

查询条件

备注

该属性隶属于 OpCPointWithAttSelectStru 对象

示例

见 CMapView 类 SelectByCondition 方法。

## 59. OpCPolygonSelectClass 类

多边形查询类

### 59.1 PointNum 属性

**public int[] PointNum**

说明

各个多边形上的点数数组

备注

该属性隶属于 OpCPolygonSelectClass 对象

示例

见 **CMapView** 类 **SelectByPolygon** 方法。

## 59.2 Polygon 属性

**public MAPGISIMS.EnterpriseLib.OpCPoint2D[] Polygon**

说明

多边形上的坐标点对象数组，

备注

该属性隶属于 OpCPolygonSelectClass 对象

示例

见 **CMapView** 类 **SelectByPolygon** 方法。

## 59.3 LayerIndex 属性

**public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCPolygonSelectClass 对象

示例

见 **CMapView** 类 **SelectByPolygon** 方法。

## 59.4 MapID 属性

**public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCPolygonSelectClass 对象

示例

见 **CMapView** 类 **SelectByPolygon** 方法。

## 59.5 PageIndex 属性

**public long PageIndex**

说明

查询结果当前页的第一条记录的索引

备注

该属性隶属于 OpCPolygonSelectClass 对象

示例  
见 **CMapView** 类 **SelectByPolygon** 方法。

## 59.6 PolygonNum 属性

**public int PolygonNum**

说明

多边形个数

备注

该属性隶属于 OpCPolygonSelectClass 对象

示例

见 **CMapView** 类 **SelectByPolygon** 方法。

## 59.7 recordCountOfPage 属性

**Public int recordCountOfPage**

说明

查询操作返回的每页的记录条数

备注

该属性隶属于 OpCPolygonSelectClass 对象

示例

见 **CMapView** 类 **SelectByPolygon** 方法。

## 60. OpCPolygonWithAttSelectStru 类

多边形条件查询类

### 60.1 Question 属性

**public string Question**

说明

查询条件

备注

该属性隶属于 OpCPolygonWithAttSelectStru 对象

示例

见 **CMapView** 类 **SelectByCondition** 方法。

## 61. OpCPrjTran 类

投影转换类

## 61.1 BZoneType 属性

### **public short BZoneType**

说明

投影分带类型 0: 6度 1: 3度 2: 1.5度

备注

该属性隶属于 OpCPrjTran 对象

## 61.2 DAngUnit 属性

### **public short DAngUnit**

说明

空间数据水平坐标单位 (GCS: 角度) 4: 度 5: 度分秒

备注

该属性隶属于 OpCPrjTran 对象

## 61.3 DLon 属性

### **public double DLon**

说明

投影中心点经度

备注

该属性隶属于 OpCPrjTran 对象

## 61.4 IPointNum 属性

### **public short IPointNum**

说明

所要转换的点个数

备注

该属性隶属于 OpCPrjTran 对象

## 61.5 IProjTypeId 属性

### **public short IProjTypeId**

说明

投影类型 5: 高斯-克吕格(横切椭圆柱等角)投影

备注

该属性隶属于 OpCPrjTran 对象

## 61.6 ISpheroid 属性

### **public short ISpheroid**

#### 说明

参考椭球体参数类型 1：北京 54/克拉索夫斯基(1940 年)椭球  
2：西安 80/1975 年 I.U.G.G 推荐椭球

#### 备注

该属性隶属于 OpCPrjTran 对象

## 61.7 IType 属性

### **public short IType**

#### 说明

地理坐标系 1：地理坐标系 3：投影平面直角坐标系

#### 备注

该属性隶属于 OpCPrjTran 对象

## 61.8 IZone 属性

### **public short IZone**

#### 说明

投影带号

#### 备注

该属性隶属于 OpCPrjTran 对象

## 61.9 ThePoint 属性

### **public MAPGISIMS.EnterpriseLib.OpCPoint2D[] ThePoint**

#### 说明

所要转换的点序列

#### 备注

该属性隶属于 OpCPrjTran 对象

## 61.10 Unit 属性

### **public short Unit**

#### 说明

空间数据水平坐标单位（PCS：长度） 2：米

#### 备注

该属性隶属于 OpCPrjTran 对象

## 62. OpCPrjTranRlt 类

投影转换结果类

### 62.1 PrjPointNum 属性

**public short PrjPointNum**

说明

点个数

备注

该属性隶属于 OpCPrjTranRlt 对象

### 62.2 PrjRltPoint 属性

**public MAPGISIMS.EnterpriseLib.OpCPoint2D[] PrjRltPoint**

说明

点数组

备注

该属性隶属于 OpCPrjTranRlt 对象

## 63. OpCRectangleSelectClass 类

拉框查询类

### 63.1 LayerIndex 属性

**public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCRectangleSelectClass 对象

示例

见 CMapView 类 SelectByRect 方法。

### 63.2 RectangleArea 属性

**public MAPGISIMS.EnterpriseLib.OpCRect RectangleArea**

说明

矩形区

备注

该属性隶属于 OpCRectangleSelectClass 对象

示例

见 CMapView 类 SelectByRect 方法。

### 63.3 MapID 属性

**public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCRectangleSelectClass 对象

示例

见 CMapView 类 SelectByRect 方法。

### 63.4 PageIndex 属性

**public long PageIndex**

说明

查询结果当前页的第一条记录的索引

备注

该属性隶属于 OpCRectangleSelectClass 对象

示例

见 CMapView 类 SelectByRect 方法。

### 63.5 recordCountOfPage 属性

**Public int recordCountOfPage**

说明

查询操作返回的每页的记录条数

备注

该属性隶属于 OpCRectangleSelectClass 对象

示例

见 CMapView 类 SelectByRect 方法。

## 64. OpCRectWithAttSelectStru 类

矩形条件查询类

## 64.1 Question 属性

**public string Question**

说明

查询条件

备注

该属性隶属于 OpCRectWithAttSelectStru 对象

示例

见 **CMapView** 类 **SelectByCondition** 方法。

## 65. OpCRegion 类

二维区域

### 65.1 LineNumber 属性

**public ushort LineNumber**

说明

组成二维区域的线数

备注

该属性隶属于 OpCRegion 对象

### 65.2 TheLines 属性

**public MAPGISIMS.EnterpriseLib.OpCLine[] TheLines**

说明

线数组

备注

该属性隶属于 OpCRegion 对象

## 66. OpCSelectResult 类

查询结果类

### 66.1 LayerCount 属性

**public short LayerCount**

说明

本次查询涉及多少个图层

**备注**

该属性隶属于 OpCSelectResult 对象

**示例**

```

OpCSelectResult selRlt;
OpCLayerSelectResult[] layerSelectRlt;
layerSelectRlt = selRlt.LayerSelectResults;
for(short j=0;j<selRlt.LayerCount;j++)
{
    //取指定图层查询到的实体号
    long[] featureIDs = layerSelectRlt[j].FeatureIDs;
    count = layerSelectRlt[j].Count;

    //取指定图层查询到的记录值集合,每个PropertyValues[i]又是一个ArrayList[]
    ArrayList[] PropertyValues = layerSelectRlt[j].PropertyValues;
    for(ushort k=0;k<count;k++)
    {
        //取指定图层指定记录的实体号
        long featureID = featureIDs[k];
        //取指定图层指定记录的属性字段集合
        ArrayList fldValCollection = PropertyValues[k];
        //取指定图层指定记录的第一个属性字段
        string fldVal = (string)fldValCollection[0];
        ShowSearchRltLst(j,featureID,k,fldVal);
    }
}

```

## 66.2 TotalCount 属性

```
public long TotalCount
```

**说明**

总共的查询结果数目

**备注**

该属性隶属于 OpCSelectResult 对象

**示例**

```

OpCSelectResult selRlt;
OpCLayerSelectResult[] layerSelectRlt;
layerSelectRlt = selRlt.LayerSelectResults;
//得到本次查询后的总记录数
long totalCount = selRlt.TotalCount;
for(short j=0;j<selRlt.LayerCount;j++)
{
    //取指定图层查询到的实体号
    long[] featureIDs = layerSelectRlt[j].FeatureIDs;
    count = layerSelectRlt[j].Count;

    //取指定图层查询到的记录值集合,每个PropertyValues[i]又是一个ArrayList[]
    PropertyValues = layerSelectRlt[j].PropertyValues;
    for(ushort k=0;k<count;k++)
    {
        //取指定图层指定记录的实体号
        long featureID = featureIDs[k];
        //取指定图层指定记录的属性字段集合
        ArrayList fldValCollection = PropertyValues[k];
        //取指定图层指定记录的第一个属性字段
        string fldVal = (string)fldValCollection[0];
        ShowSearchRltLst(j,featureID,k,fldVal);
    }
}
}

```

### 66.3 LayerSelectResults 属性

**public MAPGISIMS.EnterpriseLib.OpCLayerSelectResult[] LayerSelectResults**

说明

图层查询结果对象数组

备注

该属性隶属于 OpCSelectResult 对象

示例

```

OpCSelectResult selRlt;
OpCLayerSelectResult[] layerSelectRlt;
layerSelectRlt = selRlt.LayerSelectResults;
//得到本次查询后的总记录数
long totalCount = selRlt.TotalCount;
for(short j=0;j<selRlt.LayerCount;j++)
{
    //取指定图层查询到的实体号
    long[] featureIDs = layerSelectRlt[j].FeatureIDs;
    count = layerSelectRlt[j].Count;

    //取指定图层查询到的记录值集合,每个PropertyValues[i]又是一个ArrayList[]
    PropertyValues = layerSelectRlt[j].PropertyValues;
    for(ushort k=0;k<count;k++)
    {
        //取指定图层指定记录的实体号
        long featureID = featureIDs[k];
        //取指定图层指定记录的属性字段集合
        ArrayList fldValCollection = PropertyValues[k];
        //取指定图层指定记录的第一个属性字段
        string fldVal = (string)fldValCollection[0];
        ShowSearchRltLst(j,featureID,k,fldVal);
    }
}
}

```

## 67. OpCSpatialBuffer 类

缓冲区分析类

### 67.1 LayerIndex 属性

**public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCSpatialBuffer 对象

示例

见 CMapView 类 SpatialAnalysisBuffer 方法。

### 67.2 Radius 属性

**public double Radius**

说明

缓冲区分析半径

备注

该属性隶属于 OpCSpatialBuffer 对象

示例

见 CMapView 类 SpatialAnalysisBuffer 方法。

### 67.3 FeatureClassName 属性

**public string FeatureClassName**

说明

缓冲区分析文件名称

备注

该属性隶属于 OpCSpatialBuffer 对象

示例

见 CMapView 类 SpatialAnalysisBuffer 方法。

### 67.4 FeatureNumber 属性

**public short FeatureNumber**

说明

缓冲区分析的要素个数

备注

该属性隶属于 OpCSpatialBuffer 对象

示例

见 CMapView 类 SpatialAnalysisBuffer 方法。

### 67.5 FeatureIDS 属性

**public long[] FeatureIDS**

说明

要素 ID 号

备注

该属性隶属于 OpCSpatialBuffer 对象

例子

见 CMapView 类 SpatialAnalysisBuffer 方法

### 67.6 MapID 属性

**public Long MapID**

说明

地图 ID

备注

该属性隶属于 OpCSpatialBuffer 对象

示例

见 **CMapView** 类 **SpatialAnalysisBuffer** 方法。

## 67.7 OverlayOnCurMap 属性

**public bool OverlayOnCurMap**

说明

是否把缓冲区分析结果叠加到当前显示的地图上

备注

该属性隶属于 OpCSpatialBuffer 对象

示例

见 **CMapView** 类 **SpatialAnalysisBuffer** 方法。

## 67.8 mapDisplayBound 属性

**public MAPGISIMS.EnterpriseLib.OpCRect mapDisplayBound**

说明

当前地图的显示范围

备注

该属性隶属于 OpCSpatialBuffer 对象

示例

见 **CMapView** 类 **SpatialAnalysisBuffer** 方法。

## 67.9 mapHeight 属性

**public short mapHeight**

说明

图像高

备注

该属性隶属于 OpCSpatialBuffer 对象

示例

见 **CMapView** 类 **SpatialAnalysisBuffer** 方法。

## 67.10 mapWidth 属性

**public short mapWidth**

说明

图像高

备注

该属性隶属于 OpCSpatialBuffer 对象

示例

见 **CMapView** 类 **SpatialAnalysisBuffer** 方法。

## 68. OpCSpatialCircleClip 类

圆裁剪类

### 68.1 LayerIndexLst 属性

**public string LayerIndexLst**

说明

图层索引号

备注

该属性隶属于 OpCSpatialCircleClip 对象

示例

见 **CMapView** 类 **SpatialAnalysisCircleClip** 方法。

### 68.2 LayerIndex 属性

**public Long LayerIndex**

说明

图层索引

备注

该属性隶属于 OpCSpatialCircleClip 对象

示例

见 **CMapView** 类 **SpatialAnalysisCircleClip** 方法。

### 68.3 MapID 属性

**public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCSpatialCircleClip 对象

示例

见 **CMapView** 类 **SpatialAnalysisCircleClip** 方法。

### 68.4 Radius 属性

**public double Radius**

说明

半径

备注

该属性隶属于 OpCSpatialCircleClip 对象

示例

见 **CMapView** 类 **SpatialAnalysisCircleClip** 方法。

## 68.5 FeatureClassName 属性

**public string FeatureClassName**

说明

裁减后生成的文件名

备注

该属性隶属于 OpCSpatialCircleClip 对象

示例

见 **CMapView** 类 **SpatialAnalysisCircleClip** 方法。

## 68.6 ClipType 属性

**public short ClipType**

说明

设置圆裁减类型

备注

该属性隶属于 OpCSpatialCircleClip 对象

示例

见 **CMapView** 类 **SpatialAnalysisCircleClip** 方法。

## 68.7 Center 属性

**public MAPGISIMS.EnterpriseLib.OpCPoint2D Center**

说明

中心

备注

该属性隶属于 OpCSpatialCircleClip 对象

示例

见 **CMapView** 类 **SpatialAnalysisCircleClip** 方法。

## 68.8 SetNearDis 方法

**public void SetNearDis(double dis)**

说明

设置裁剪容差半径

**参数**

dis: 半径

**返回值**

无

**备注**

该方法隶属于 OpCSpatialCircleClip 对象

**示例**

见 CMapView 类 SpatialAnalysisCircleClip 方法。

## 69. OpCSpatialOperResult 类

空间分析返回结果类

## 70. OpCSpatialOverlay 类

叠加分析类

### 70.1 GDBDataSourceName1 属性

**public string GDBDataSourceName1**

**说明**

叠加所需图层 1 数据源名称

**备注**

该属性隶属于 OpCSpatialOverlay 对象

**示例**

见 CMapView 类 SpatialAnalysisOverlay 方法。

### 70.2 GDBDataSourceName2 属性

**public string GDBDataSourceName2**

**说明**

叠加所需图层 2 数据源名称

**备注**

该属性隶属于 OpCSpatialOverlay 对象

**示例**

见 CMapView 类 SpatialAnalysisOverlay 方法。

### 70.3 GDBName1 属性

**public string GDBName1**

说明

叠加所需图层 1GDB 名

备注

该属性隶属于 OpCSpatialOverlay 对象

示例

见 **CMapView** 类 **SpatialAnalysisOverlay** 方法。

### 70.4 GDBName2 属性

**public string GDBName2**

说明

叠加所需图层 2d 的 GDB 名

备注

该属性隶属于 OpCSpatialOverlay 对象

示例

见 **CMapView** 类 **SpatialAnalysisOverlay** 方法。

### 70.5 FeatureIDLst1 属性

**public string FeatureIDLst1**

说明

第一个数据库的图层名称，多个用','连接

备注

该属性隶属于 OpCSpatialOverlay 对象

示例

见 **CMapView** 类 **SpatialAnalysisOverlay** 方法。

### 70.6 FeatureIDLst2 属性

**public string FeatureIDLst2**

说明

第二个数据库的图层名称，多个用','连接

备注

该属性隶属于 OpCSpatialOverlay 对象

示例

见 **CMapView** 类 **SpatialAnalysisOverlay** 方法。

## 70.7 LayerIndex1 属性

**public long LayerIndex1**

说明

第一个图层索引

备注

该属性隶属于 OpCSpatialOverlay 对象

示例

见 **CMapView** 类 **SpatialAnalysisOverlay** 方法。

## 70.8 LayerIndex2 属性

**public long LayerIndex2**

说明

第二个图层索引

备注

该属性隶属于 OpCSpatialOverlay 对象

示例

见 **CMapView** 类 **SpatialAnalysisOverlay** 方法。

## 70.9 OverlayType 属性

**public short OverlayType**

说明

叠加类型

备注

该属性隶属于 OpCSpatialOverlay 对象

示例

见 **CMapView** 类 **SpatialAnalysisOverlay** 方法。

## 70.10 OverlayGDBName 属性

**public string OverlayGDBName**

说明

叠加生成的文件名

备注

该属性隶属于 OpCSpatialOverlay 对象

示例

见 **CMapView** 类 **SpatialAnalysisOverlay** 方法。

## 70.11 MapID 属性

**public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCSpatialOverlay 对象

示例

见 **CMapView** 类 **SpatialAnalysisOverlay** 方法。

## 70.12 password1 属性

**public string password1**

说明

网络数据密码

备注

该属性隶属于 OpCSpatialOverlay 对象

示例

见 **CMapView** 类 **SpatialAnalysisOverlay** 方法。

## 70.13 password2 属性

**public string password2**

说明

网络数据密码

备注

该属性隶属于 OpCSpatialOverlay 对象

示例

见 **CMapView** 类 **SpatialAnalysisOverlay** 方法。

## 70.14 userName1 属性

**public string userName1**

说明

网络数据用户名

备注

该属性隶属于 OpCSpatialOverlay 对象

示例

见 **CMapView** 类 **SpatialAnalysisOverlay** 方法。

## 70.15 userName2 属性

**public string userName2**

说明

网络数据用户名

备注

该属性隶属于 OpCSpatialOverlay 对象

示例

见 CMapView 类 SpatialAnalysisOverlay 方法。

## 71. OpCSpatialOverlayByID 类

要素叠加分析类

### 71.1 iNumOfLst1 属性

**public int iNumOfLst1**

说明

图层 1 中所要提取的要素 ID 个数

备注

该属性隶属于 OpCSpatialOverlayByID 对象

示例

见 CMapView 类 SpatialOverlayByID 方法。

### 71.2 iNumOfLst2 属性

**public int iNumOfLst2**

说明

图层 2 中所要提取的要素 ID 个数

备注

该属性隶属于 OpCSpatialOverlayByID 对象

示例

见 CMapView 类 SpatialOverlayByID 方法。

### 71.3 layerIndex1 属性

**public long layerIndex1**

说明

图层 1 序号

备注

该属性隶属于 OpCSpatialOverlayByID 对象

示例

见 **CMapView** 类 **SpatialOverlayByID** 方法。

## 71.4 layerIndex2 属性

**public long layerIndex2**

说明

图层 2 序号

备注

该属性隶属于 OpCSpatialOverlayByID 对象

示例

见 **CMapView** 类 **SpatialOverlayByID** 方法。

## 71.5 mapID 属性

**public long mapID**

说明

地图 ID

备注

该属性隶属于 OpCSpatialOverlayByID 对象

示例

见 **CMapView** 类 **SpatialOverlayByID** 方法。

## 71.6 OverlayType 属性

**public short OverlayType**

说明

叠加类型 如：1

备注

该属性隶属于 OpCSpatialOverlayByID 对象

示例

见 **CMapView** 类 **SpatialOverlayByID** 方法。

## 71.7 pFeatureIDLst1 属性

**public long[] pFeatureIDLst1**

说明

图层 1 中所要提取的要素 ID 序列

备注

该属性隶属于 OpCSpatialOverlayByID 对象

示例  
见 **CMapView** 类 **SpatialOverlayByID** 方法。

## 71.8 pFeatureIDLst2 属性

**public long[] pFeatureIDLst2**

说明

图层 2 中所要提取的要素 ID 序列

备注

该属性隶属于 OpCSpatialOverlayByID 对象

示例

见 **CMapView** 类 **SpatialOverlayByID** 方法。

## 72. OpCSpatialPolygonClip 类

多边形裁剪类

### 72.1 LayerIndex 属性

**public long LayerIndex**

说明

图层索引号

备注

该属性隶属于 OpCSpatialPolygonClip 对象

示例

见 **CMapView** 类 **SpatialAnalysisPolygonClip** 方法。

### 72.2 LayerIndexLst 属性

**public string LayerIndexLst**

说明

多图层裁剪

备注

该属性隶属于 OpCSpatialPolygonClip 对象

示例

见 **CMapView** 类 **SpatialAnalysisPolygonClip** 方法。

### 72.3 ClipType 属性

**public int ClipType**

说明

裁剪类型

备注

该属性隶属于 OpCSpatialPolygonClip 对象

示例

见 **CMapView** 类 **SpatialAnalysisPolygonClip** 方法。

## 72.4 FeatureClassName 属性

**public string FeatureClassName**

说明

裁剪后生成的文件名称

备注

该属性隶属于 OpCSpatialPolygonClip 对象

示例

见 **CMapView** 类 **SpatialAnalysisPolygonClip** 方法。

## 72.5 PolygonNum 属性

**public int PolygonNum**

说明

多边形个数

备注

该属性隶属于 OpCSpatialPolygonClip 对象

示例

见 **CMapView** 类 **SpatialAnalysisPolygonClip** 方法。

## 72.6 Points 属性

**public MAPGISIMS.EnterpriseLib.OpCPoint2D[] Points**

说明

坐标点序列

备注

该属性隶属于 OpCSpatialPolygonClip 对象

示例

见 **CMapView** 类 **SpatialAnalysisPolygonClip** 方法。

## 72.7 LpPointNum 属性

**public int[] LpPointNum**

说明

各个多边形点的个数

**备注**

该属性隶属于 OpCSpatialPolygonClip 对象

**示例**

见 **CMapView** 类 **SpatialAnalysisPolygonClip** 方法。

## 72.8 MapID 属性

**public long MapID**

**说明**

地图 ID

**备注**

该属性隶属于 OpCSpatialPolygonClip 对象

**示例**

见 **CMapView** 类 **SpatialAnalysisPolygonClip** 方法。

## 72.9 SetNearDis 方法

**public void SetNearDis(double dis)**

**说明**

设置误差半径

**参数**

dis: 误差半径

**返回值**

无

**备注**

该属性隶属于 OpCSpatialPolygonClip 对象

**示例**

见 **CMapView** 类 **SpatialAnalysisPolygonClip** 方法。

## 73. OpCTest 类

获取 GIS 服务上的地图文档列表

### 73.1 op 属性

**public short op**

**说明**

要素个数

**备注**

该属性隶属于 OpCTest 对象

## 74. OpCUniformResult 类

添加、删除、更新要素统一返回值类

### 74.1 OperCode 属性

**public int OperCode**

说明

操作代码

备注

该属性隶属于 OpCUniformResult 对象

### 74.2 Flag 属性

**public int Flag**

说明

返回标志

备注

该属性隶属于 OpCUniformResult 对象

### 74.3 Information 属性

**public string Information**

说明

信息

备注

该属性隶属于 OpCUniformResult 对象

## 75. OpCUpdateFeature 类

更新点、线、区要素类

### 75.1 FeatureID 属性

**public long FeatureID**

说明

要素 ID

备注

该属性隶属于 OpCUpdateFeature 对象

## 75.2 LayerIndex 属性

**public long LayerIndex**

说明

图层索引

备注

该属性隶属于 OpCUpdateFeature 对象

## 75.3 MapID 属性

public long MapID

说明

地图 ID

备注

该属性隶属于 OpCUpdateFeature 对象

## 75.4 NewFeature 属性

**public MAPGISIMS.EnterpriseLib.OpCFeatureGeometry NewFeature**

说明

新的要素几何信息

备注

该属性隶属于 OpCUpdateFeature 对象

## 76. OpCUpdateFeatureAttribute 类

更新点、线、区要素类

### 76.1 LayerIndex 属性

**public long LayerIndex**

说明

图层索引

备注

该属性隶属于 OpCUpdateFeatureAttribute 对象

示例

见 CMapView 类 UpdateFeature 方法。

## 76.2 FeatureID 属性

**public long FeatureID**

说明

要素 ID

备注

该属性隶属于 OpCUpdateFeatureAttribute 对象

示例

见 **CMapView** 类 **UpdateFeature** 方法。

## 76.3 FieldNumber 属性

**public int FieldNumber**

说明

字段个数

备注

该属性隶属于 OpCUpdateFeatureAttribute 对象

示例

见 **CMapView** 类 **UpdateFeature** 方法。

## 76.4 NewFieldValues 属性

**public string[] NewFieldValues**

说明

新字段值

备注

该属性隶属于 OpCUpdateFeatureAttribute 对象

示例

见 **CMapView** 类 **UpdateFeature** 方法。

## 76.5 MapID 属性

**public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCUpdateFeatureAttribute 对象

示例

见 **CMapView** 类 **UpdateFeature** 方法。

## 77. OpCUpdateFeatureStyleInfo 类

更新要素样式属性类

### 76.1 LayerIndex 属性

**public long LayerIndex**

说明

图层索引

备注

该属性隶属于 OpCUpdateFeatureStyleInfo 对象

### 77.2 FeatureId 属性

**public long FeatureId**

说明

要素 ID

备注

该属性隶属于 OpCUpdateFeatureStyleInfo 对象

### 77.3 MapID 属性

**public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCUpdateFeatureStyleInfo 对象

## 78. OpCUpdateLayerInfomation 类

更新图层信息类

### 78.1 LayerIndex 属性

**public long LayerIndex**

说明

图层索引

备注

该属性隶属于 OpCUpdateLayerInfomation 对象

## 78.2 MapID 属性

**public long MapID**

说明

地图 ID

备注

该属性隶属于 OpCUpdateLayerInfomation 对象

## 78.3 UpdatedLayer 属性

**public MAPGISIMS.EnterpriseLib.OpCLayerInformation UpdatedLayer**

说明

图层信息

备注

该属性隶属于 OpCUpdateLayerInfomation 对象

## 79. OpCUpdateLayerLst 类

更新多个图层信息类

### 79.1 layerInfo 属性

**public MAPGISIMS.EnterpriseLib.OpCLayerInformation[ ] layerInfo**

说明

各个图层信息数组

备注

该属性隶属于 OpCUpdateLayerLst 对象

### 79.2 layerNum 属性

**public int layerNum**

说明

图层个数

备注

该属性隶属于 OpCUpdateLayerLst 对象

### 79.3 mapID 属性

**public int mapID**

说明

地图 ID

备注

该属性隶属于 OpCUpdateLayerLst 对象

## 80. GridServerInfo 类

栅格流服务器信息

### 80.1 HdfInfoArray 属性

**public MAPGISIMS.EnterpriseLib.HDFInfo[] HdfInfoArray**

说明

HDF 信息数组

备注

该属性隶属于 GridServerInfo 对象

### 80.2 ServerIP 属性

**public string ServerIP**

说明

栅格流服务器 IP

备注

该属性隶属于 GridServerInfo 对象

### 80.3 ServerPort 属性

**public string ServerPort**

说明

栅格流服务器端口

备注

该属性隶属于 GridServerInfo 对象

## 81. HDFInfo 类

HDF 信息

## 81.1 Name 属性

**public string Name**

说明

HDF 名称

备注

该属性隶属于 HDFInfo 对象

## 81.2 ShowLevel 属性

**public string ShowLevel**

说明

显示的级别区间,如 0,6 表示该 HDF 数据从第 0 级显示到第 6 级

备注

该属性隶属于 HDFInfo 对象

## 81.3 Xmax 属性

**public string Xmax**

说明

地图坐标 X 方向最大值

备注

该属性隶属于 HDFInfo 对象

## 81.4 Ymax 属性

**public string Ymax**

说明

地图坐标 Y 方向最大值

备注

该属性隶属于 HDFInfo 对象

## 81.5 Xmin 属性

**public string Xmin**

说明

地图坐标 X 方向最小值

备注

该属性隶属于 HDFInfo 对象

## 81.5 Ymin 属性

**public string Ymin**

说明

地图坐标 Y 方向最小值

备注

该属性隶属于 HDFInfo 对象

## 82. IMSConfigManager 类

IMS 配置管理器

### 82.1 GetDataProvider 方法

**public string GetDataProvider()**

说明

获取数据提供者信息

备注

该属性隶属于 IMSConfigManager 对象

### 82.2 GetDBConnectionString 方法

**public string GetDBConnectionString()**

说明

获取数据库连接字符串

备注

该属性隶属于 IMSConfigManager 对象

### 82.3 GetGridServerInfo 方法

**public MAPGISIMS.EnterpriseLib.GridServerInfo[] GetGridServerInfo()**

说明

获取栅格流 GIS 服务器信息

备注

该属性隶属于 IMSConfigManager 对象

### 82.4 GetNeedGisMgrInfo 方法

**public bool GetNeedGisMgrInfo()**

**说明**

获取 GIS 服务器管理器信息

**备注**

该属性隶属于 IMSConfigManager 对象

## 82.5 GetVectorServerInfo 方法

**public MAPGISIMS.EnterpriseLib.VectorServerInfo[] GetVectorServerInfo()**

**说明**

获取矢量 GIS 服务器信息

**备注**

该属性隶属于 IMSConfigManager 对象

## 82.6 IMSConfigManager 方法

**public IMSConfigManager(string configFile)**

**说明**

IMS 配置管理器

**备注**

该属性隶属于 IMSConfigManager 对象

## 83. MapDocInfo 类

地图文档信息

### 83.1 MapDocInfo 属性

**public MapDocInfo**

**说明**

地图文档信息

**备注**

该属性隶属于 MapDocInfo 对象

### 83.2 Name 属性

**public string Name**

**说明**

地图文档名称

**备注**

该属性隶属于 MapDocInfo 对象

### 83.3 Xmax 属性

**public double Xmax**

说明

地图坐标 X 方向最大值

备注

该属性隶属于 MapDocInfo 对象

### 83.4 Ymax 属性

**public double Ymax**

说明

地图坐标 Y 方向最大值

备注

该属性隶属于 MapDocInfo 对象

### 83.5 Xmin 属性

**public double Xmin**

说明

地图坐标 X 方向最小值

备注

该属性隶属于 MapDocInfo 对象

### 83.6 Ymin 属性

**public double Ymin**

说明

地图坐标 Y 方向最小值

备注

该属性隶属于 MapDocInfo 对象

## 84. OpCFeatureGeometries 类

多个要素空间信息类

### 84.1 iFeatureNum 方法

**public int iFeatureNum**

说明

要素个数

备注

该属性隶属于 OpCFeatureGeometries 对象

## 84.1 pInfo 方法

**public MAPGISIMS.EnterpriseLib.OpCFeatureGeometry[] pInfo**

说明

要素类数组

备注

该属性隶属于 OpCFeatureGeometries 对象

示例

```
OpCFeatureGeometry feature = viewport.GetSingleFclsGeometry(geo).pInfo[0];
```

## 85. OpCGetFclsRect 类

获取要素类逻辑范围

### 85.1 ClassToXML 方法

**public virtual void ClassToXML(ref System.Collections.ArrayList fieldValues, ref ushort[] loop, ref ushort index)**

说明

ClassToXML

参数

fieldValues:

loop:

index:

备注

该方法隶属于 OpCGetFclsRec 对象

### 85.2 OpCGetFclsRect 构造函数

**public OpCGetFclsRect()**

说明

获取空间分析结果要素类逻辑范围类构造函数

备注

该方法隶属于 OpCGetFclsRec 对象

### 85.3 XMLToClass 方法

**public virtual void XMLToClass(System.Collections.IEnumerator myEnumerator)**说明

XMLToClass

参数

myEnumerator:

备注

该方法隶属于 OpCGetFclsRec 对象

### 85.4 FclsName 属性

**public string FclsName**

说明

要素类名称

备注

该方法隶属于 OpCGetFclsRec 对象

### 85.5 HDFName 属性

**public string HDFName**

说明

要素类所属 HDF 名称

备注

该方法隶属于 OpCGetFclsRec 对象

### 85.6 LayerType 属性

**public MAPGISIMS.EnterpriseLib.LayerTypeEnum LayerType**

说明

图层类型(要素类、简单要素类)

备注

该方法隶属于 OpCGetFclsRec 对象

### 85.7 NetServiceName 属性

**public string NetServiceName**

说明

网络服务名

备注

该方法隶属于 OpCGetFclsRec 对象

## 85.8 Password 属性

**public string Password**

说明

密码

备注

该方法隶属于 OpCGetFclsRec 对象

## 85.9 UserName 属性

**public string UserName**

说明

用户名

备注

该方法隶属于 OpCGetFclsRec 对象

## 86. OpCGetFclsSel 类

获取要素类逻辑范围

### 86.1 ClassToXML 方法

**public virtual void ClassToXML(ref System.Collections.ArrayList fieldValues, ref ushort[] loop, ref ushort index)**

说明

ClassToXML

参数

fieldValues:

loop:

index:

备注

该方法隶属于 OpCGetFclsSel 对象

### 86.2 OpCGetFclsSel 构造函数

**public OpCGetFclsSel()**

说明

查询空间分析结果类构造函数

备注

该方法隶属于 OpCGetFclsSel 对象

示例

```
//取要素图层的属性  
OpCGetFclsSel sr = new OpCGetFclsSel();  
sr.NetServiceName = this.gdbSourceList.SelectedValue;
```

## 86.3 XMLToClass 方法

**public virtual void XMLToClass(System.Collections.IEnumerator myEnumerator)**说明

XMLToClass

参数

myEnumerator:

备注

该方法隶属于 OpCGetFclsSel 对象

## 86.4 FclsName 属性

**public string FclsName**

说明

要素类名称

备注

该方法隶属于 OpCGetFclsSel 对象

## 86.5 HDFName 属性

**public string HDFName**

说明

要素类所属 HDF 名称

备注

该方法隶属于 OpCGetFclsSel 对象

## 86.6 LayerType 属性

**public MAPGISIMS.EnterpriseLib.LayerTypeEnum LayerType**

说明

图层类型(要素类、简单要素类)

备注

该方法隶属于 OpCGetFclsSel 对象

## 86.7 NetServiceName 属性

**public string NetServiceName**

说明

网络服务名

备注

该方法隶属于 OpCGetFclsSel 对象

## 86.8 Password 属性

**public string Password**

说明

密码

备注

该方法隶属于 OpCGetFclsSel 对象

## 86.9 UserName 属性

**public string UserName**

说明

用户名

备注

该方法隶属于 OpCGetFclsSel 对象

## 87. OpCGetSingleFclsGeo 类

获取单要素类某个要素空间信息类

### 87.1 ClassToXML 方法

**public virtual void ClassToXML(ref System.Collections.ArrayList fieldValues, ref ushort[] loop, ref ushort index)**

说明

ClassToXML

参数

fieldValues:

loop:

index:

备注

该方法隶属于 OpCGetSingleFclsGeo 对象

## 87.2 OpCGetSingleFclsGeo 方法

**public OpCGetSingleFclsGeo()**

备注

该方法隶属于 OpCGetSingleFclsGeo 对象

示例

```
//取要素图层的空间
OpCGetSingleFclsGeo geo = new OpCGetSingleFclsGeo();
geo.NetServiceName = this.gdbSourceList.SelectedValue;
```

## 87.3 XMLToClass 方法

**public virtual void XMLToClass(System.Collections.IEnumerator myEnumerator)说明**

XMLToClass

参数

myEnumerator:

备注

该方法隶属于 OpCGetSingleFclsGeo 对象

## 87.4 FeatureID 属性

**public long[] FeatureID**

说明

要获取空间信息的要素 ID

备注

该方法隶属于 OpCGetSingleFclsGeo 对象

## 87.5 FeatureNum 属性

**public string FeatureNum**

说明

要素个数

备注

该方法隶属于 OpCGetSingleFclsGeo 对象

## 88. OpCSpatialBufferByGeom 类

获取单要素类某个要素空间信息类

## 88.1 ClassToXML 方法

**public virtual void ClassToXML(ref System.Collections.ArrayList fieldValues, ref ushort[] loop, ref ushort index)**

说明

ClassToXML

参数

fieldValues:

loop:

index:

备注

该方法隶属于 OpCSpatialBufferByGeom 对象

## 88.2 OpCSpatialBufferByGeom 方法

**public OpCSpatialBufferByGeom ()**

备注

该方法隶属于 OpCSpatialBufferByGeom 对象

示例

```
CMapView viewport = (CMapView)Session["viewport"];
OpCSpatialBufferByGeom buffByGeom = new OpCSpatialBufferByGeom ();
buffByGeom.FeatureNumber = 1;
```

## 88.3 XMLToClass 方法

**public virtual void XMLToClass(System.Collections.IEnumerator myEnumerator)**说明

XMLToClass

参数

myEnumerator:

备注

该方法隶属于 OpCSpatialBufferByGeom 对象

## 88.4 FeatureGeometry 属性

**public MAPGISIMS.EnterpriseLib.OpCFeatureGeometry[] FeatureGeometry**

说明

要素空间信息

备注

该方法隶属于 OpCSpatialBufferByGeom 对象

## 88.5 FeatureNumber 属性

**public int FeatureNumber**

说明

要素个数

备注

该方法隶属于 OpCSpatialBufferByGeom 对象

## 88.6 MapID 属性

**public int MapID**

说明

地图文档 ID

备注

该方法隶属于 OpCSpatialBufferByGeom 对象

## 88.7 Radius 属性

**public double Radius**

说明

缓冲分析半径

备注

该方法隶属于 OpCSpatialBufferByGeom 对象

## 88.8 RebuildFlag 属性

**public bool RebuildFlag**

说明

重建面积长度(默认要重建)

备注

该方法隶属于 OpCSpatialBufferByGeom 对象

## 88.9 ResultFeatureClassName 属性

**public string ResultFeatureClassName**

说明

分析结果生成的要素类名

备注

该方法隶属于 OpCSpatialBufferByGeom 对象

## 89. OpCSpatialDiffer 类

判断要素拓扑关系类

### 89.1 ClassToXML 方法

**public virtual void ClassToXML(ref System.Collections.ArrayList fieldValues, ref ushort[] loop, ref ushort index)**

说明

ClassToXML

参数

fieldValues:

loop:

index:

备注

该方法隶属于 OpCSpatialDiffer 对象

### 89.2 OpCSpatialDiffer 方法

**public OpCSpatialDiffer ()**

备注

该方法隶属于 OpCSpatialDiffer 对象

示例

```
protected void AddTopo2 ()
{
    CMapView viewport = (CMapView)Session["viewport"];
    OpCSpatialDiffer sd = (OpCSpatialDiffer)Session["Differ"];
    sd.FeatureGeometry2 = GetGeom(viewport);
    Session["Differ"] = sd;
}
```

### 89.3 XMLToClass 方法

**public virtual void XMLToClass(System.Collections.IEnumerator myEnumerator)说明**

XMLToClass

参数

myEnumerator:

备注

该方法隶属于 OpCSpatialDiffer 对象

## 89.4 FeatureGeometry1 属性

**public MAPGISIMS.EnterpriseLib.OpCFeatureGeometry FeatureGeometry1**

说明

参与判断的第一个要素

备注

该方法隶属于 OpCSpatialDiffer 对象

## 89.5 FeatureGeometry2 属性

**public MAPGISIMS.EnterpriseLib.OpCFeatureGeometry FeatureGeometry2**

说明

参与判断的第二个要素

备注

该方法隶属于 OpCSpatialDiffer 对象

## 90. OpCSpatialOverlayByGeom 类

要素与要素根据空间信息叠加分析类

### 90.1 ClassToXML 方法

**public virtual void ClassToXML(ref System.Collections.ArrayList fieldValues, ref ushort[] loop, ref ushort index)**

说明

ClassToXML

参数

fieldValues:

loop:

index:

备注

该方法隶属于 OpCSpatialOverlayByGeom 对象

### 90.2 OpCSpatialOverlayByGeom 方法

**public OpCSpatialOverlayByGeom ()**

备注

该方法隶属于 OpCSpatialOverlayByGeom 对象

### 90.3 XMLToClass 方法

**public virtual void XMLToClass(System.Collections.IEnumerator myEnumerator)**说明

XMLToClass

参数

myEnumerator:

备注

该方法隶属于 OpCSpatialOverlayByGeom 对象

### 90.4 FeatureGeometry1 属性

**public MAPGISIMS.EnterpriseLib.OpCFeatureGeometry FeatureGeometry1**

说明

参与判断的第一个要素

备注

该方法隶属于 OpCSpatialOverlayByGeom 对象

### 90.5 FeatureGeometry2 属性

**public MAPGISIMS.EnterpriseLib.OpCFeatureGeometry FeatureGeometry2**

说明

参与判断的第二个要素

备注

该方法隶属于 OpCSpatialOverlayByGeom 对象

### 90.6 OverlayType 属性

**public int OverlayType**

说明

叠加类型

备注

该方法隶属于 OpCSpatialOverlayByGeom 对象

## 91. VectorServerInfo 类

矢量服务器信息

## 91.1 LogPath 属性

**public string LogPath**

说明

日志的路径

备注

该方法隶属于 VectorServerInfo 对象

## 91.2 MapDocInfoArray 属性

**public MAPGISIMS.EnterpriseLib.MapDocInfo[] MapDocInfoArray**

说明

地图文档信息数组

备注

该方法隶属于 VectorServerInfo 对象

## 91.3 NeedGisMgr 属性

**public bool NeedGisMgr**

说明

是否启用 GIS 服务器管理器

备注

该方法隶属于 VectorServerInfo 对象

## 91.4 ServerIP 属性

**public string ServerIP**

说明

GIS 服务器连接 IP 地址

备注

该方法隶属于 VectorServerInfo 对象

## 91.5 ServerPort 属性

**public string ServerPort**

说明

GIS 服务器连接端口号

备注

该方法隶属于 VectorServerInfo 对象

## 91.6 WriteLog 属性

**public bool WriteLog**

说明

日志标志符

备注

该方法隶属于 VectorServerInfo 对象

## 92. MapSetting 类

**public class MapSetting**

备注

http: //localhost/EntComm/ 的成员

### 92.1 GetEditableLayersIndex 方法

**public string GetEditableLayersIndex()**

说明

获取可编辑图层索引字符串

备注

该方法隶属于 MapSetting 对象

示例

```
MapSetting ms = new MapSetting();  
string layerStr = ms.GetEditableLayersIndex();
```

### 92.2 GetGridLayer 方法

**private string GetGridLayer()**

备注

该方法隶属于 MapSetting 对象

示例

```
sb.Append(this.GetGridLayer());
```

### 92.3 GetGridLayer 方法

**public MAPGISIMS.EnterpriseLib.OpCLayerInformation[] GetLayerList()**

说明

根据图层显示索引获得图层列表

备注

该方法隶属于 MapSetting 对象

示例

```
case "GetMapLayerList"://获取图层列表
    MapSetting ms = new MapSetting();
    context.Response.Write(ms.GetLayerListByJSON());
    break;
```

## 92.4 GetGridLayer 方法

**public string GetLayerListByJSON()**

说明

以 JSON 的方式返回图层列表属性信息

备注

该方法隶属于 MapSetting 对象

示例

```
case "GetMapLayerList"://获取图层列表
    MapSetting ms = new MapSetting();
    context.Response.Write(ms.GetLayerListByJSON());
    break;
```

## 92.5 GetLayerStatus 方法

**private string GetLayerStatus(MAPGISIMS.EnterpriseLib.OpCLayerInformation pMapLayer)**

说明

返回图层状态

参数

pMapLayer:

备注

该方法隶属于 MapSetting 对象

示例

```
private string GetLayerStatus(OpCLayerInformation pMapLayer)
```

## 92.6 GetLayerType 方法

**public MAPGISIMS.EnterpriseLib.LayerTypeEnum GetLayerType(int layerIndex)**

说明:

获取图层类型

备注

该方法隶属于 MapSetting 对象

示例

```
fclsWithGeom.LayerType = ms.GetLayerType(Convert.ToInt32(layerStr));
```

## 92.7 GetMapNameConfig 方法

### **public static string GetMapNameConfig()**

说明

获取 web.config 中<mapName>的值

备注

该方法隶属于 MapSetting 对象

示例

```
public static void ReLoadMap ()
{
    MapDisplay.LoadMap (MapSetting.GetMapNameConfig(), 10, 10);
}
```

## 92.8 GetNetLayerIndex 方法

### **public int GetNetLayerIndex()**

说明

获取网文件图层索引

备注

该方法隶属于 MapSetting 对象

示例

```
public int GetNetLayerIndex ()
{
    int index = -1;
    int layerNum = _viewport.MapObj.mapCount;
    OpCLayerInformation[] layers = new OpCLayerInformation[layerNum];
    for (int i = 0; i < layerNum; i++)
    {
        OpCLayerInformation pMapLayer = _viewport.MapObj.getLayerInformation(i);
        if ((pMapLayer.Editable == 1 || pMapLayer.Selectable==1) && pMapLayer.LayerType == 22)
            index = Convert.ToInt32(pMapLayer.LayerIndex);
    }
    return index;
}
```

## 92.9 GetSelectableLayersIndex 方法

### **public string GetSelectableLayersIndex()**

说明

获取可选图层索引字符串

备注

该方法隶属于 MapSetting 对象

示例

```
int index=-1;
MapSetting ms = new MapSetting();
string layerStr = ms.GetSelectableLayersIndex();
```

## 92.10 GetVisibleLayersIndex 方法

### public string GetVisibleLayersIndex()

说明

获取可见图层索引字符串

备注

该方法隶属于 MapSetting 对象

示例

```
public string GetVisibleLayersIndex()
{
    StringBuilder sb = new StringBuilder();
    int layerNum = _viewport.MapObj.mapCount;
    OpCLayerInformation[] layers = new OpCLayerInformation[layerNum];
```

## 92.11 SetLayerStatus 方法

### private void SetLayerStatus(ref MAPGISIMS.EnterpriseLib.OpCLayerInformation pMapLayer, string status)

说明

设定图层状态

参数:

pMapLayer:

备注

该方法隶属于 MapSetting 对象

示例

```
for (int i = 0; i < layerNum; i++)
{
    int index = Convert.ToInt32(layerArr[i + 1].Split('=')[0].Substring(5));
    OpCLayerInformation pMapLayer = _viewport.MapObj.getLayerInformation(index);
    pMapLayer.DisplayIndex = i;
    SetLayerStatus(ref pMapLayer, layerArr[i + 1].Split('=')[1]);
    maplayer.layerInfo[index] = pMapLayer;
}
```

## 92.12 UpdateMap 方法

### public void UpdateMap(string requestString)

说明

更新地图文档

参数:

idxList: 新的图层序号

备注

该方法隶属于 MapSetting 对象

示例

```
case "UpdateMap"://更新地图文档
    ms = new MapSetting();
    ms.UpdateMap(context.Request.Form.ToString());
    break;
```

## 93. ScriptWriter 类

**public class ScriptWriter**

备注

http: //localhost/EntComm/ 的成员

### 93.1 AddClassData 方法

**public void AddClassData(params string[] list)**

说明

添加 JSON 数据，与 SetClassAttribute 中的属性一一对应

参数:

list: 属性数据，可多个

备注

该方法隶属于 ScriptWriter 对象

示例

```
sw.AddClassData(param);
```

### 93.2 ClearJSONData 方法

**public void ClearJSONData()**

说明

清空 JSON 数据

备注

该方法隶属于 ScriptWriter 对象

示例

```
public void ClearJSONData()
{
    if (_sb.Length > 0)
        _sb.Remove(0, _sb.Length);
}
```

### 93.3 GetJSONArrStr 方法

**public string GetJSONArrStr()**

说明

返回 JSON 对象数组字符串

备注

该方法隶属于 ScriptWriter 对象

示例

```
public string GetJSONArrStr ()
{
    if (_sb.Length > 0)
    {
        _sb.Insert(0, JSON_ARRAY_BEGIN);
        _sb.Insert(_sb.Length, JSON_ARRAY_END);
        return _sb.ToString();
    }
    else
        return "";
}
```

### 93.4 GetJSONObjStr 方法

**public string GetJSONObjStr()**

说明

返回 JSON 对象字符串

备注

该方法隶属于 ScriptWriter 对象

示例

```
public string GetJSONObjStr ()
{
    if (_sb.Length > 0)
        return _sb.ToString();
    else
        return "";
}
```

### 93.5 ScriptWriter 方法

**public ScriptWriter(System.Web.UI.Page scriptInPage)**

说明

带参数的 ScriptWriter 构造函数

参数

scriptInPage: 要写入脚本的页面

备注

该方法隶属于 ScriptWriter 对象

示例

```
public ScriptWriter(Page scriptInPage)
{
    _page = scriptInPage;
}
```

### 93.6 ScriptWriter 方法

**public ScriptWriter()**

**说明**

不带参数的 ScriptWriter 构造函数

**备注**

该方法隶属于 ScriptWriter 对象

**示例**

```
ScriptWriter sw = new ScriptWriter();
```

## 93.7 SetClassAttribute 方法

**public void SetClassAttribute(params string[] atts)****说明**

定义数组对象中，每个元素对象的属性,用于 JSON 数据

**参数**

atts: 属性字符串，可多个

**备注**

该方法隶属于 ScriptWriter 对象

**示例**

```
sw.SetClassAttribute("id", "text", "leaf", "cls", "qtip", "listeners");
```

## 93.8 Write 方法

**public void Write(string key, string script)****说明**

写一般脚本到前台页面

**参数**

key: 要注册的脚本键值

script: 要注册的脚本文本

**备注**

该方法隶属于 ScriptWriter 对象

**示例**

```
ScriptWriter sw = new ScriptWriter();
```

## 93.9 WriteJSONArray 方法

**public void WriteJSONArray(string key)****说明**

写带 JSON 数据的脚本到页面

**参数**

key: 要添加的数组对象的名称

script: 要注册的脚本文本

**备注**

该方法隶属于 ScriptWriter 对象

示例

```
public void WriteJSONArray(string key)
{
    _sb.Insert(0, "var " + key + "=" + JSON_ARRAY_BEGIN);
    _sb.Insert(_sb.Length, JSON_ARRAY_END + ";");

    ClientScriptManager csm = _page.ClientScript;
    Type cstype = _page.GetType();
    //form标记结束之前
    if (!csm.IsStartupScriptRegistered(cstype, key))
        csm.RegisterStartupScript(cstype, key, _sb.ToString(), true);
}
```

## A.1.2 .NET 结构体

### 1. ChineseWordUnit 结构体

```
public struct ChineseWordUnit
```

#### 1.1 power 属性

```
public string power
```

说明:

权重

备注

隶属于 ChineseWordUnit 对象

示例:

```
string path = HttpContext.Current.Server.MapPath(".") + @"\ChineseDictionary.txt";
ChineseParse cp = new ChineseParse(path);
ChineseWordUnit[] arryWords = cp.ParseChinese(key);
String value= arryWords[i].power;
```

#### 1.2 word 属性

```
public string word
```

说明:

词语

备注

隶属于 ChineseWordUnit 对象

示例:

```
string path = HttpContext.Current.Server.MapPath(".") + @"\ChineseDictionary.txt";
ChineseParse cp = new ChineseParse(path);
ChineseWordUnit[] arryWords = cp.ParseChinese(key);
```

```
String value= arryWords[i].word;
```

### 1.3 ChineseWordUnit 方法

**public ChineseWordUnit ( string word , string power )**

说明:

结构初始化。

参数:

word: 中文词语

power: 该词语的权重

备注

隶属于 ChineseWordUnit 对象

示例:

```
ChineseWordUnit[] arryWords
```

## 2. Dot\_2D 结构体

**public struct Dot\_2D**

说明:

二维点结构体

### 2.1 X 属性

**public double x**

说明:

x 坐标, InfoDot 结构体成员之一

备注

隶属于 Dot\_2D 对象

### 2.2 Y 属性

**public double y**

说明:

y 坐标, InfoDot 结构体成员之一

备注

隶属于 Dot\_2D 对象

### 2.3 Dot\_2D 方法

**public Dot\_2D(double x, double y)**

**说明**

二维点构造函数

**参数**

x: X 坐标

y: Y 坐标

**备注**

隶属于 Dot\_2D 对象

### 3. GDBInfomation 结构体

**public struct GDBInfomation**

**说明**

GDB 信息结构体

#### 3.1 dataSourceName 属性

**public string dataSourceName**

**说明**

GDB 数据源名称

**备注**

该属性隶属于 GDBInfomation 对象

#### 3.2 HDFName 属性

**public string HDFName**

**说明**

数据源下所有数据库名，用逗号分割

**备注**

该属性隶属于 GDBInfomation 对象

#### 3.3 layerName 属性

**public string[] layerName**

**说明**

数据库下图层名称，用逗号分割

**备注**

该属性隶属于 GDBInfomation 对象

## 4. HDFConfig 结构体

### **public struct HDFConfig**

说明:

HDF 地图库配置,包括 GIS 服务器地址, GIS 服务器端口, HDF 文件名称等配置信息。

示例:

```
HDFConfig gisConfig= HDFConfig ( gisServerIP , gisServerPort , hdfName , rect)
double logicXlength = gisConfig.MapRect.XMaxMap - gisConfig.MapRect.XMinMap;
gisConfig= HDFConfig ( gisServerIP , gisServerPort , hdfName , rect)
double logicXlength = gisConfig.MapRect.XMaxMap - gisConfig.MapRect.XMinMap;
```

### 4.1 GisServerIP 属性

#### **public String GisServerIP**

说明:

GIS 服务器地址

备注

隶属于 HDFConfig 对象

示例:

```
HDFConfig gisConfig= HDFConfig ( gisServerIP , gisServerPort , hdfName , rect)
double logicXlength = gisConfig.MapRect.XMaxMap - gisConfig.MapRect.XMinMap;
```

### 4.2 GisServerPort 属性

#### **public int GisServerPort**

说明:

GIS 服务器端口

备注

隶属于 HDFConfig 对象

示例:

```
HDFConfig gisConfig= HDFConfig ( gisServerIP , gisServerPort , hdfName , rect)
CMapViewPort viewport = new CMapViewPort(gisConfig.GisServerIP,
gisConfig.GisServerPort);
```

### 4.3 HDFName 属性

#### **public String HDFName**

说明:

HDF 文件名称

备注

隶属于 HDFConfig 对象

示例:

```
HDFConfig gisConfig= HDFConfig ( gisServerIP , gisServerPort , hdfName , rect)
buffer = viewport.GetMapImage(gisConfig.HDFName, gisConfig.StartLevel, row, cell,
mapLevel);
```

## 4.4 MapRect 属性

**public MAPGISIMS.EnterpriseLib.MapRectangle MapRect**

说明:

地图坐标范围

备注

隶属于 HDFConfig 对象

示例:

```
HDFConfig gisConfig= HDFConfig ( gisServerIP , gisServerPort , hdfName , rect)
double logicXlength = gisConfig.MapRect.XMaxMap - gisConfig.MapRect.XMinMap;
```

## 4.5 StartLevel 属性

**public int StartLevel**

说明:

地图库中开始的级数

备注

隶属于 HDFConfig 对象

示例:

```
HDFConfig gisConfig= HDFConfig ( gisServerIP , gisServerPort , hdfName , rect)
buffer = viewport.GetMapImage(gisConfig.HDFName, gisConfig.StartLevel, row, cell,
mapLevel);
```

## 4.6 DealSize 属性

**public int DealSize**

说明:

地图缓冲处理的大小

备注

隶属于 HDFConfig 对象

## 4.7 Degree 属性

**public int Degree**

说明:

将地图进行经纬度切分时的大小

备注

隶属于 HDFConfig 对象

## 4.8 ImgWidth 属性

**public int ImgWidth**

说明:

单个地图图片的大小

备注

隶属于 HDFConfig 对象

## 4.9 HDFConfig 方法

**public HDFConfig(string gisServerIP, int gisServerPort, string hdfName, MAPGISIMS.EnterpriseLib.MapRectangle rect)**

说明:

HDF地图库配置，默认从第0级开始

参数:

gisServerIP: GIS服务器地址

gisServerPort: GIS服务器端口

hdfName: HDF文件名称

rect: 地图坐标范围

备注

隶属于 HDFConfig 对象

示例:

```
HDFConfig gisConfig= HDFConfig ( gisServerIP , gisServerPort , hdfName , rect)
```

```
buffer = viewport.GetMapImage(gisConfig.HDFName, gisConfig.StartLevel, row, cell, mapLevel);
```

## 4.10 HDFConfig 方法

**public HDFConfig(string gisServerIP, int gisServerPort, string hdfName, MAPGISIMS.EnterpriseLib.MapRectangle rect, int startLevel)**

说明:

HDF地图库配置

参数:

gisServerIP: GIS服务器地址

gisServerPort: GIS服务器端口

hdfName: HDF文件名称

rect: 地图坐标范围

startLevel: 地图库中开始的级数

备注

隶属于 HDFConfig 对象

示例:

```
HDFConfig gisConfig= HDFConfig (gisServerIP ,gisServerPort , hdfName , rect, lev)  
buffer = viewport.GetMapImage(gisConfig.HDFName, gisConfig.StartLevel, row, cell,  
gisConfig .lev);
```

## 4.11 HDFConfig 方法

**public HDFConfig(string gisServerIP, int gisServerPort, string hdfName,  
MAPGISIMS.EnterpriseLib.MapRectangle rect, int imgWidth, int dealSize)**

说明:

HDF地图库配置

参数:

gisServerIP: GIS服务器地址

gisServerPort: GIS服务器端口

hdfName: HDF文件名称

rect: 地图坐标范围

startLevel: 地图库中开始的级数

dealSize: 地图缓冲处理的大小

备注

隶属于 HDFConfig 对象

## 4.12 HDFConfig 方法

**public HDFConfig(string gisServerIP, int gisServerPort, string hdfName,  
MAPGISIMS.EnterpriseLib.MapRectangle rect, int startLevel, int imgWidth, int  
dealSize, double degree)**

说明:

HDF地图库配置

参数:

gisServerIP: GIS服务器地址

gisServerPort: GIS服务器端口

hdfName: HDF文件名称

rect: 地图坐标范围

startLevel: 地图库中开始的级数

imgWidth: 单个地图图片的大小

dealSize: 地图缓冲处理的大小

degree: 将地图进行经纬度切分时的大小

备注

隶属于 HDFConfig 对象

## 5. MapRectangle 结构体

**public struct MapRectangle**

说明:

地图坐标范围,坐标点为左下右上型

### 5.1 XMinMap 属性

**public Double XMinMap**

说明:

矩形左下角点的 X 坐标

备注

隶属于 MapRectangle 对象

### 5.2 XMaxMap 属性

**public Double XMaxMap**

说明:

矩形右上角点的 X 坐标

备注

隶属于 MapRectangle 对象

### 5.3 YMinMap 属性

**public Double YMinMap**

说明:

矩形左下角点的 Y 坐标

备注

隶属于 MapRectangle 对象

### 5.4 YMaxMap 属性

**public Double YMaxMap**

说明:

矩形右上角点的 Y 坐标

备注

隶属于 MapRectangle 对象

## 5.5 MapRectangle 方法

**public MapRectangle(double xMin, double yMin, double xMax, double yMax)**

说明:

根据参数初始化矩形范围

参数:

xMin: 矩形左下角点的X坐标

yMin: 矩形左下角点的Y坐标

xMax: 矩形右上角点的X坐标

yMax: 矩形右上角点的 Y 坐标

备注

隶属于 MapRectangle 对象

## 6. OpCPoint2D 结构体

**public struct OpCPoint2D**

说明

定义二维点

### 6.1 X 属性

**public double X**

说明

X 坐标

备注

该属性隶属于 OpCPoint2D 对象

### 6.2 Y 属性

**public double Y**

说明

Y 坐标

备注

该属性隶属于 OpCPoint2D 对象

### 6.3 OpCPoint2D 方法

**public OpCPoint2D(double xPoint, double yPoint)**

说明

构造函数

**参数**

xPoint: X 坐标

yPoint: Y 坐标

**备注**

该属性隶属于 OpCPoint2D 对象

## 7. OpCRect 结构体

**public struct OpCRect****说明**

二维矩形区

### 7.1 XMax 属性

**public double XMax****说明**

X 轴最大值

**备注**

该属性隶属于 OpCRect 对象

### 7.2 XMin 属性

**public double XMin****说明**

X 轴最小值

**备注**

该属性隶属于 OpCRect 对象

### 7.3 YMax 属性

**public double YMax****说明**

Y 轴最大值

**备注**

该属性隶属于 OpCRect 对象

### 7.4 YMin 属性

**public double YMin****说明**

Y 轴最小值

**备注**

该属性隶属于 OpCRect 对象

## 7.5 OpCRect 方法

**public OpCRect(double xmi, double xma, double ymi, double yma)**

**说明**

构造函数

**参数**

xmi: X 轴最小值

xma: X 轴最大值

ymi: Y 轴最小值

yma: Y 轴最大值

**备注**

该属性隶属于 OpCRect 对象

## A.1.3 .NET 枚举

### 1. DatabaseType 枚举

**public enum DatabaseType**

**说明:**

表示数据库类型，用于查询数据库时使查询函数能自适应不同数据库类型。

成员名称	说明
Oracle	ORACLE 数据库
SQLServer	MS SQLSERVER 数据库
OleDb	通过 OLEDB 方式连接的数据库

### 2. ImageFormatEnum 枚举

**public enum ImageFormatEnum**

**说明**

图像格式

成员名称	说明
bmp	bmp 格式
gif	gif 格式
jpg	jpg 格式
png	png 格式

### 3. ImageQuality 枚举

**public enum ImageQuality**

说明:

图像质量

成员名称	说明
high	高质量
low	低质量
middle	中等质量

### 4. LayerStatusEnum 枚举

**public enum LayerStatusEnum**

说明:

图层状态标志

成员名称	说明
Close	关闭
Editable	可编辑
Selectable	可查询
Visible	可见

### 5. OutputType 枚举

**public enum OutputType**

说明:

换乘方案输出格式类型

成员名称	说明
str	字符串形式
xml	xml 格式字符串

### 6. QueryTypeEnum 枚举

**public enum QueryTypeEnum**

说明:

查询类型

成员名称	说明
SelectByCircle	圆查询
SelectByCircleCon	圆条件查询
Selectable	条件查询
SelectByHitting	点击查询

SelectByLine	线查询
SelectByNone	无查询操作
SelectByPointCon	点击条件查询
SelectByPolygon	多边形查询
SelectByPolygonCon	多边形条件查询据
SelectByRect	矩形查询
SelectByRectCon	矩形条件查询

## 7. FieldTypeEnum 枚举

public enum **FieldTypeEnum**

说明:

属性字段类型

成员名称	说明
Bool	布尔型
Byte	字节型
Date	日期型
Decimal	数值型
Double	双精度浮点型
Float	单精度浮点型
Int64	64 位整型
Long	长整型
None	未知类型
Short	短整型
String	字符串型
Time	时间型

## 8. LayerTypeEnum 枚举

public enum **LayerTypeEnum**

说明:

图层类型(要素类、简单要素类)

成员名称	说明
成员名称	说明
FeatureClass	要素类
None	未知类型
SimpleFeatureClass	简单要素类

## 9. LayerTypeEnum 枚举

**public enum OverlayType**

说明:

叠加类型

成员名称	说明
Combination	合并运算
Intersect	相交运算
Subtract	相减运算
Symmetrical	对称差运算

## 10. SpatialRltDbTypeEnum 枚举

**public enum SpatialRltDbTypeEnum**

说明:

空间分析操作存放的数据库的类型

成员名称	说明
ORACLE	oracle 数据类型
SQL	SQL 数据类型

## 11. TopoTypeEnum 枚举

**public enum TopoTypeEnum**

说明:

拓扑关系类型

成员名称	说明
Adjacent	相邻
Disjoin	相离
Include	包含
Intersect	相交
None	未知类型

## A.2 JAVA 函数库

**jmapgis-enterprise.jar** 包应用于 MAPGIS7-IMS 行业版，里面包含了很多类库，方便用户进行行业版的 java 环境的二次开发。

## A.2.1 JAVA 包

### 1. com.mapgis.ims.see.com 包

#### 1.1 CMapViewPort 类

地图视图类。主要用来取地图与保存地图。

##### 1.1.1 public CMapViewPort()

**说明:**

默认构造函数。

**参数:**

无

**返回值:**

无

##### 1.1.2 public CMapViewPort (String host, int port )

**说明:**

初始化CMapViewPort类的实例，根据主机名和端口号连接GIS服务器。

**参数:**

host: GIS服务器IP地址

port: GIS 服务器端口号

**返回值:**

无

**示例:**

```
String gisIP = myconfig.getInitParameter("gisService");  
int gisPort = Integer.parseInt(myconfig.getInitParameter("gisPort"));  
CMapViewPort viewport = new CMapViewPort(gisIP,gisPort);
```

##### 1.1.3 public CMapViewPort (String host, int port,int buffSize )

**说明:**

初始化CMapViewPort类的实例，根据主机名和端口号连接GIS服务器。

**参数:**

host: GIS服务器IP地址

port: GIS 服务器端口号

bufferSize: 缓存大小(MB)

**返回值:**

无

**备注:**

实例化此对象与1.1.2不同之处在于可设置缓存大小。

**示例:**

```
String gisIP = myconfig.getInitParameter("gisService");
int gisPort = Integer.parseInt(myconfig.getInitParameter("gisPort"));
int bufferSize = Integer.parseInt(myconfig.getInitParameter("bufferSize"));
CMapViewPort viewport = new CMapViewPort(gisIP,gisPort, bufferSize);
```

#### **1.1.4 public byte[] GetMapImage(String strHDFName, int iStartLvl, int iRowNo, int iColNo, int iLevel) throws IOException**

**说明:**

获取指定参数下的地图图片

**参数:**

strHDFName: HDF地图文件名

iStartLvl: HDF库存储图片开始的级数

iRowNo: 图片所在的行号

iColNo: 图片所在的列号

iLevel: 图片所处的级数

**返回值:**

返回地图图片的字节数组

**备注:**

此类获取的地图图片是栅格图片。

**示例:**

#### **1.1.5 public byte[] GetMapImage(String strHDFName, int iStartLvl, int iRowNo, int iColNo, int iLevel, int dealSize) throws IOException**

**说明:**

获取指定参数下的地图图片

**参数:**

strHDFName: HDF地图文件名

iStartLvl: HDF库存储图片开始的级数

iRowNo: 图片所在的行号

iColNo: 图片所在的列号

iLevel: 图片所处的级数

dealSize: 地图缓冲处理的大小

**返回值:**

返回地图图片的字节数组

**备注:**

此类获取的地图图片是栅格图片。

**示例:**

```
byte[] by_mapImage = viewport.GetMapImage(hdfName,0,rowNo,lineNo,level,dealSize);
```

#### **1.1.6 public byte[] SaveMap(CHDFConfig[] hdfConfig, CMapRectangle winRect, int iMapLevel)**

**说明:**

保存指定逻辑范围内的地图为文件

**参数:**

hdfConfig: HDF相关配置

winRect: 指定的逻辑范围

iMapLevel: 地图级数

**返回值:**

返回图片的字节数组

**示例:**

```
CHDFConfig[] hdfConfig = new CHDFConfig[t_layerArr.length];
//配置多个地图对象
for(int i = 0; i < t_layerArr.length; i++) {
    String HDFTmpName = "HDFName" + t_layerArr[i];
    hdfConfig[i] = new CHDFConfig();
    hdfConfig[i].setGisIP(gisIP);
    hdfConfig[i].setGisPort(gisPort);
    hdfConfig[i].setHDFName(session.getAttribute(HDFTmpName).toString());
    hdfConfig[i].setStartlevel(StartLevel);
    hdfConfig[i].setMapRect(logicCoordinate);
}
//窗口显示的地图逻辑范围
CMapRectangle windowCoordinate = new CMapRectangle(startX, startY, endX, endY);
CMapViewPort viewport = new CMapViewPort(gisIP,gisPort);
//保存地图
byte[] byMapImage = viewport.SaveMap(hdfConfig, windowCoordinate, level);
```

### **1.1.7 public void SaveMap(CHDFConfig[] hdfConfig, CMapRectangle winRect, int iMapLevel, String strSavePath)**

**说明:**

保存指定逻辑范围内的地图为文件，可指定保存的路径

**参数:**

hdfConfig: HDF相关配置

winRect: 指定的逻辑范围

iMapLevel: 地图级数

strSavePath: 指定的保存路径

**返回值:**

返回图片的字节数组

**示例:**

与1.1.6类似，添加参数保存图片的路径

//保存地图

```
byte[] byMapImage = viewport.SaveMap(hdfConfig,windowCoordinate,level, strSavePath);
```

### 1.1.8 public void SetIndexOffset(long offset)

说明:

取地图时设置索引偏移量

参数:

**Offset:** 索引偏移量。0为新版HDF；2为旧版HDF

返回值:

无

示例:

```
viewport.SetIndexOffset(0);
```

## 1.2 CHDFConfig 类

HDF 相关参数配置，示例见 CMapViewPort 类 SaveMap 方法。

### 1.2.1 public void setGisIP(java.lang.String strGisIP)

说明:

设置GIS服务器IP地址

参数:

strGisIP: GIS服务器IP

### 1.2.2 public java.lang.String getGisIP()

说明:

获取GIS服务器IP地址

### 1.2.3 public void setGisPort(int iGisPort)

说明:

设置GIS服务器端口号

参数:

iGisPort: GIS服务器端口号

### 1.2.4 public int getGisPort()

说明:

获取GIS服务器端口号

### 1.2.5 public void setHDFName(java.lang.String strHdfName)

说明:

设置保存的HDF数据名称

参数:

strHdfName: HDF地图文件名

### 1.2.6 public java.lang.String getHDFName()

说明:

获取保存的HDF地图文件名

### 1.2.7 public void setMapRect(CMapRectangle mapRect)

说明:

设置保存的地图范围

参数:

mapRect: CMapRectangle对象, 地图范围

### 1.2.8 public CMapRectangle getMapRect()

说明:

获取地图范围, 返回CMapRectangle对象

### 1.2.9 public void setStartlevel(int iStartlev)

说明:

设置地图起始级别, 默认值为0

参数:

iStartlev: 地图起始级别

### 1.2.10 public int getStartlevel()

说明:

获取地图起始级别

参数:

strGisIP: GIS服务器IP

strHdfName: HDF地图文件名

mapRect: CMapRectangle对象, 保存的地图范围

## 1.3 CMapRectangle 类

HDF 相关参数配置, 示例见 CMapViewPort 类 SaveMap 方法。

### 1.3.1 public CMapRectangle(double xmin,double ymin, double xmax,double ymax)

说明:

根据参数初始化矩形范围

参数:

xmin: 矩形左下角点的X坐标

xmax: 矩形左下角点的Y坐标

ymin: 矩形右上角点的X坐标

ymax: 矩形右上角点的Y坐标

## 1.4 CHotSpot 类

三维热点类，用于三维热点的显示与入库

### 1.4.1 public String GetHotSpotData(String grids)

**说明：**

根据给定网格号获取id、名称、坐标等热点信息

**参数：**

grids: 网格号

**返回值：**

返回热点信息字符串

**备注：**

适用于旧版的热点信息查询

**示例：**

```
CHotSpot dbHot = new CHotSpot();  
dbHot.GetHotSpotData(grids);
```

### 1.4.2 public String GetBuildName(String strName)

**说明：**

获取与给定参数相符的热点名称，用以给出查询输入提示信息

**参数：**

strName: 用户输入查询字符

**返回值：**

返回与参数名称相符的热点名字符串

**示例：**

```
String name = request.getParameter("name");  
out.print(dbHot.GetBuildName(name));
```

### 1.4.3 public String AddNew3D(String strGid, String strName, String strAddress, String strCoords)

**说明：**

在热点数据库中添加所指定的热点

**参数：**

strGid: 网格号

strName: 热点名

strAddress: 热点地址

strCoords: 热点坐标

**返回值：**

入库成功则返回OK，失败则返回空字符串

**示例：**

```
String strRes = dbHot.AddNew3D(gid, name, adr, coords);
```

#### 1.4.4 public String InputCoords(String name,String coords)

**说明:**

更改指定热点的坐标范围

**参数:**

name: 热点名

coords: 坐标范围

**返回值:**

更改成功返回OK, 失败则返回ERROR

**示例:**

```
String strRes = dbHot.InputCoords(name, coords);
```

### 1.5 CGetVipData 类

该类提供 VIP 点信息以及地图标注点信息的获取和更新等方法。

#### 1.5.1 public String MapDate(String strlayer, String grids, int grade)

**说明:**

获取VIP点的名称、坐标等信息

**参数:**

strlayer: 查询表名。若为空指在默认表中查找。

coords: 网格号

grade: 查询标志位, 若为0指默认查询语句; 若为非0可重新设置查询语句, 按需求查询VIP点信息。

**返回值:**

成功则返回查询结果; 失败时则返回空字符串

**示例:**

```
String strRlt = VipData.MapDate("", "", 0);
```

#### 1.5.2 public int getIndex()

**方法说明:**

获取标注点最大索引号

**参数说明:**

无

**返回值:**

返回最大索引号加 1 后的值

#### 1.5.3 public String getLabel(int iUserId)

**说明:**

获取指定ID的标注点信息, 并转化为XML格式

**参数:**

iUserId: 标注点ID

**返回值:**

返回 XML 格式化信息字符串

**备注:**

根据标注点的 ID 号可得到此标注的信息: 如逻辑坐标值、标注的内容等。

#### **1.5.4 public void updataLabel(String strLabelName, String strLabelDetail, int iLevelNum,int iLabelID, float labelPosX, float labelPosY)**

**方法说明:**

新增标注点入库

**参数说明:**

strLabelName: 标注点名称

strLabelDetail: 标注点详细信息

iLevelNum: 图层号

iLabelID: 标注点ID

labelPosX: 标注点X坐标

labelPosY: 标注点Y坐标

**返回值:**

无返回值

**备注:**

一般是先新增标注点信息 updataLabel 之后, 再调用 getLabel 获取标注点信息。

## **2. com.mapgis.ims.sec.util 包**

### **2.1 GISDataQuery 类**

此类主要提供图形查询相关的方法, 如: 点击查询、拉框查询、画圆查询、多边形查询等。该类中的查询方法可以支持 SQL Server、Oracle、MySQL 等多种数据库类型的图形查询。

#### **2.1.1 public GISDataQuery ()**

**说明:**

GISDataQuery的构造函数, 初始化数据查询方法

**参数:**

无

**返回值:**

无

**示例:**

```
CGISDataQuery gisDataQuery = new CGISDataQuery();
```

#### **2.1.2 public String GetDataByHitting(String gridcode, double ox, double oy, double offset)**

**方法说明:**

点击查询调用方法

**参数说明:**

gridcode: 图片的网格号

ox: 点x坐标

oy: 点y坐标

offset: 点的误差偏移量

**返回值:**

点击查询结果字符串

### **2.1.3 public String GetDataByHitting(String gridcode, double ox, double oy, double offset, int[] layers)**

**方法说明:**

点击查询调用方法

**参数说明:**

gridcode: 图片的网格号

ox: 点x坐标

oy: 点y坐标

offset: 点的误差偏移量

layers: 所选择图层, 用于分类查询

**返回值:**

点击查询结果字符串

### **2.1.4 public String GetDataByCircle(String gridcode, int level, double ox, double oy, double radius)**

**说明:**

圆查询调用方法。

**参数:**

gridcode: 图片的网格号

level: 网格所在级数

ox: 圆心x坐标

oy: 圆心y坐标

radius: 半径

**返回值:**

圆查询结果字符串

**示例:**

```
String[] paramArray = param.split("#");
```

```
String strResultXml =
```

```
dataQuery.GetDataByCircle(gridcode,level,Double.parseDouble(paramArray[0]),
```

```
Double.parseDouble(paramArray[1]), Double.parseDouble(paramArray[2]));
```

### **2.1.5 public String GetDataByCircle(String gridcode, int level, double ox, double oy, double radius, int[] layers)**

**说明:**

圆查询调用方法。

**参数:**

gridcode: 图片的网格号

level: 网格所在级数

ox: 圆心x坐标

oy: 圆心y坐标

radius: 半径

layers: 所选择图层, 用于分类查询

**返回值:**

圆查询结果字符串

### **2.1.6 public String GetDataByRectangle(String gridcode, int level, double startX, double startY, double endX, double endY)**

**说明:**

矩形拉框查询调用方法

**参数:**

gridcode: 图片的网格号

level: 网格所在级数

startX: 矩形左上角点x坐标

startY: 矩形左上角点y坐标

endX: 矩形右下角点x坐标

endY: 矩形右下角点y坐标

**返回值:**

矩形拉框查询结果字符串

**示例:**

```
String[] paramArray = param.split("#");
```

```
String strResultXml = dataQuery.GetDataByRectangle(gridcode,level,
```

```
Double.parseDouble(paramArray[0]), Double.parseDouble(paramArray[1]),
```

```
Double.parseDouble(paramArray[2]), Double.parseDouble(paramArray[3]));
```

### **2.1.7 public String GetDataByRectangle(String gridcode, int level, double startX, double startY, double endX, double endY, int[] layers)**

**说明:**

矩形拉框查询调用方法

**参数:**

gridcode: 图片的网格号

level: 网格所在级数

startX: 矩形左上角点x坐标

startY: 矩形左上角点y坐标

endX: 矩形右下角点x坐标  
endY: 矩形右下角点y坐标  
layers: 所选择图层, 用于分类查询  
返回值:  
矩形拉框查询结果字符串

### **2.1.8 public String GetDataByPolygon(String gridcode, int level, double[] Xarry, double[] Yarry)**

说明:  
多边形查询调用方法  
参数:  
gridcode: 图片的网格号  
level: 网格所在级数  
Xarry: x点坐标集  
Yarry: y点坐标集  
返回值:  
多边形查询结果字符串  
示例:  

```
String[] paramArray = param.split("#");  
String strResultXml = dataQuery.GetDataByPolygon(gridcode,level,  
StrToDouble(paramArray[0]), StrToDouble(paramArray[1]));
```

### **2.1.9 public String GetDataByPolygon(String gridcode, int level, double[] Xarry, double[] Yarry, int[] layers)**

说明:  
多边形查询调用方法  
参数:  
gridcode: 图片的网格号  
level: 网格所在级数  
Xarry: x点坐标集  
Yarry: y点坐标集  
layers: 所选择图层, 用于分类查询  
返回值:  
多边形查询结果字符串

### **2.1.10 public String GetDataByCondition(String[] key)**

说明:  
关键字查询调用方法  
参数:  
key: 关键字字符串数组  
返回值:  
关键字查询结果字符串  
示例:

```

String key = String.valueOf(request.getParameter("key"));
String[] arrKey = key.split(" ");
//中文分词
strPath = application.getRealPath("ChineseDictionary.txt");
cpObj = new ChineseParse(strPath);
ChineseWordUnit[] t_arrWords = cpObj.ParseChinese(key);

int t_iArrLen = t_arrWords.length;
String[] strQueryArr = new String[t_iArrLen];
for (int i = 0; i < t_iArrLen; i++) {
    if (t_arrWords[i] != null) {
        strQueryArr[i] = t_arrWords[i].getWord();
    } else {
        break;
    }
}
//关键字查询
String res = dbQuery.GetDataByCondition(strQueryArr);

```

**备注:**

在调用查找关键字之前，可先调用中文分词的方法，可以对所查词语拆分后进行查找，使查找的信息更多。

**2.1.11 public String GetHotSpotData(String[] keywords)****说明:**

关键字查询三维热点信息

**参数:**

key: 关键字字符串数组

**返回值:**

三维热点查询结果字符串

**示例:**

```

GISDataQuery dbQuery = new GISDataQuery();
String res = dbQuery.GetHotSpotData(keywords);

```

**2.1.12 public String GetLayerData()****说明:**

从图层分类表中获取图层信息

**参数:**

无

**返回值:**

返回图层查询结果字符串

**示例:**

```

GISDataQuery dbQuery = new GISDataQuery();
String res = dbQuery.GetLayerData ();

```

### 2.1.13 public String QueryPlaceXY(String placeName)

**说明:**

查询地点位置的坐标值

**参数:**

placeName: 地点名称

**返回值:**

地点坐标值, 类型是CPoint2D

**示例:**

```
//获取起点与终点的坐标值
CPoint2D[] WNetDot = new CPoint2D[2];
//查询地点位置的坐标值
for (int i = 0; i < 2; i++) {
    WNetDot[i] = dbQuery.QueryPlaceXY(station[i]);
}
```

### 2.1.14 public void setLayerName(String layers)

**说明:**

设置所要查询数据库表名

**参数:**

layers: 数据库表名

**返回值:**

无

**示例:**

```
//设置所要查询的图层名称
CGISDataQuery dbQuery = new CGISDataQuery();
dbQuery.setLayerName("C027");
```

### 2.1.15 public void setSelectLayer(String strSelectlayer)

**说明:**

设置分图层查询

**参数:**

strSelectlayer: 所选择的图层, -1指选择全部

**返回值:**

无

**示例:**

```
//设置所要查询的图层名称
CGISDataQuery dbQuery = new CGISDataQuery();
//关键字查询
dbQuery.setSelectLayer(type)
```

## 2.2 CPathQuery 类

### 2.2.1 public String GetDrivePath(String[] station, String stringX, String stringY,String DBSVR,String NetName,String gisIP,int gisPort)

**说明:**

路径分析方法。根据起始站点的名称在指定的网络数据中查找最优路径，并返回路径分析的逻辑坐标值。

**参数:**

station: 起始点位置名字符数组

stringX: 障碍点X坐标，默认为空

stringY: 障碍点Y坐标，默认为空

DBSVR: 附加到MAPGIS里的GDB名称(db)

NetName: 网络图层名称(武汉道路网)

gisIP: GIS服务器IP地址

gisPort: GIS服务器端口

**返回值:**

路径查询结果字符串

**示例:**

```
String roads = pathQuery.GetDrivePath(station, stringX, stringY,  
DBSVR, NetName, gisIP, gisPort);
```

## 3.com.mapgis.ims.see.buscom 包

### 3.1 CBusCom 类

此类用于公交换乘。根据入库的公交数据进行换乘分析，提供最优换乘方案。

#### 3.1.1 public CBusCom(CBusData dataCache)

**说明:**

初始化公交换乘查询类方法。

**参数:**

dataCache: 是 BusData 类对象，参见 CBusData 类

**返回值:**

无

**示例:**

```
CBusData busData = new CBusData();  
busData.Init();  
CBusCom dbCar = new CBusCom(busData);
```

### 3.1.2 public void GetGjhcScheme(String startStop, String endStop, int iTimeLimit, String opType)

**说明:**

公交换乘方案，初始入口函数

**参数:**

startStop: 客户在公交换乘查询文本框中键入的起点名

endStop: 客户在公交换乘查询文本框中键入的终点名

iTimeLimit: 换乘次数，指小于等于这个数的公交换乘次数

opType: 输出格式，有两种格式：字符串 str 或 xml

**返回值:**

无

**示例:**

```
CBusCom dbCar = new CBusCom(busData);  
dbCar.GetGjhcScheme(staPos, endPos, 3, "str");
```

## 3.2 CBusData 类

### 3.2.1 public void Init()

**说明:**

初始化公交换乘的信息

**参数:**

无

**返回值:**

无

**示例:**

```
CBusData busData = new CBusData();  
busData.Init();
```

### 3.2.2 public String BusStationQuery(String strStationValue)

**说明:**

获得公交站点提示信息，通过给定关键字查询包含该字符串的站点名集合。

**参数:**

strStationValue: 查询关键字

**返回值:**

以字符串形式返回包含关键字的站点集合(只取前 8 条)

**示例:**

```
CBusData dbCarQuery = new CBusData();  
String strBusStation = dbCarQuery.BusStationQuery(stopvalue);
```

## 4.com.mapgis.ims.pro.function 包

### 4.1 MapViewPort 类

#### 4.1.1 public MapViewPort(java.net.InetSocketAddress[] gisSvrArr)

说明:

MapViewPort 构造函数,适合在有 GIS 服务器集群的环境。

参数:

gisSvrArr - GIS 服务器或 GIS 服务管理器的 IP 和端口序列。

示例:

```
InetSocketAddress[] gisSvrArr = new InetSocketAddress[1];
gisSvrArr[0] = new InetSocketAddress("127.0.0.1", 5141);
MapViewPort viewport = new MapViewPort(gisSvrArr);
```

#### 4.1.2 public MapViewPort(java.lang.String ip,int port)

说明:

MapViewPort 构造函数,适合在仅有一台 GIS 服务器环境。

参数:

ip - GIS 服务器的 IP;

port - GIS 服务器的端口。

示例:

```
MapViewPort viewport = new MapViewPort("127.0.0.1", 5141);
```

#### 4.1.3 public void setOpTimeOut(int timeout)

说明:

设置操作最长等待时间。

参数:

timeout - 毫秒为单位, 0 为无限等待, 默认为零。

示例:

```
Viewport. setOpTimeOut(0);
```

#### 4.1.4 public void setBuildFileMode(boolean save)

说明:

设置出图方式。

参数:

true: 先在本地生成图片; false: 直接生成二进制流。

示例:

```
Viewport. setBuildFileMode (true);
```

#### 4.1.5 public boolean getBuildFileMode()

说明:

得到出图方式。

**返回:**

true: 先在本地生成图片; false: 直接生成二进制流。

**示例:**

```
boolean flag = viewport.getBuildFileMode();
```

#### **4.1.6 public void setImageDisplayParam(java.lang.String[] params)**

**说明:**

设置显示参数。

**参数:**

依次为: 显示标尺、显示坐标点、随比例显示、缓冲显示、还原显示、动态显示、图像质量

**示例:**

```
viewport.setImageDisplayParam({"true","false","true","false","true","false","3"});
```

#### **4.1.7 public static void setNeedGisMgr(boolean setUpGisMgr)**

**说明:**

设置 GIS 集群管理器标志位。

**参数:**

setUpGisMgr - ture 需要; false 不需要.默认为 false

**示例:**

```
MapViewPort.setNeedGisMgr(false)
```

#### **4.1.8 public void setImageFolder(java.lang.String imageFolder)**

**说明:**

设置图片下载路径;

**参数:**

imageFolder - 图片下载路径;

**示例:**

```
viewport.setImageFolder("D: \\mapimage");
```

#### **4.1.9 public java.lang.String getImageFolder()**

**说明:**

获取图片下载路径

**返回:**

图片下载路径

#### **4.1.10 public java.lang.String getZipFileName()**

**说明:**

获取空间分析结果压缩包文件名

**返回:**

压缩包文件名

#### **4.1.11 public java.lang.String getMapImageName()**

说明:

获取最后得到地图的图片名

返回:

最新一张地图的图片名

#### **4.1.12 public java.lang.String getEagleImageName()**

说明:

获取最后得到鹰眼图的图片名

返回:

最新一张鹰眼图的图片名

#### **4.1.13 public void setImageHeight(int imageHeight)**

说明:

设置地图高度

参数:

imageHeight -地图高度

#### **4.1.14 public int getImageHeight()**

说明:

得到地图高度

返回:

地图高度

#### **4.1.15 public void setImageWidth(int imageWidth)**

说明:

设置地图宽度

参数:

imageWidth -地图宽度

#### **4.1.16 public int getImageWidth()**

说明:

得到地图宽度

返回:

地图宽度

#### **4.1.17 public void setOriginalFlag(java.lang.String original)**

说明:

设置还原显示标志位

参数:

originalFlag – “true”：设置；“false”：不设置

#### 4.1.18 public byte getOriginalFlag()

说明:

获得还原显示标志位

返回:

还原显示标志位: 0: 否 1: 是

#### 4.1.19 public void setShowScale(java.lang.String value)

说明:

是否显示标尺

参数:

value - "false": 否 "true": 是

#### 4.1.20 public int getShowScale()

说明:

获得显示标尺标志位

返回:

显示标尺: 0: 否 1: 是

#### 4.1.21 public void setShowCoordinate(java.lang.String value)

说明:

是否显示坐标点

参数:

value - "false": 否 "true": 是

#### 4.1.22 public int getShowCoordinate()

说明:

获得显示坐标点标志位

返回:

显示坐标点: 0: 否 1: 是

#### 4.1.23 public void setFollowScale(java.lang.String value)

说明:

设置是否跟随比例显示

参数:

value - "false": 否 "true": 是

#### 4.1.24 public int getFollowScale()

说明:

获得跟随比例显示标志位

返回:

跟随比例显示: 0: 否 1: 是

**4.1.25 public void setBufferShow(java.lang.String value)**

说明:

设置是否缓冲显示

参数:

value - “false”: 否 “true”: 是

**4.1.26 public int getBufferShow()**

说明:

获得缓冲显示标志位

返回:

缓冲显示: 0: 否 1: 是

**4.1.27 public void setImageQuality(java.lang.String value)**

说明:

设置图像质量

参数:

value - 1: 低 2: 中 3: 高

**4.1.28 public int getImageQuality()**

说明:

获得图像质量等级

返回:

图像质量: 1: 低 2: 中 3: 高

**4.1.29 public void setTransparentFlag(byte transparentFlag)**

说明:

设置是否透明显示

参数:

transparentFlag - 否 0 : 是 1

**4.1.30 public byte getTransparentFlag()**

说明:

获得是否透明显示标志位

返回:

透明显示: 0: 否 1: 是

**4.1.31 public void setDynamicMark(int value)**

说明:

设置动态标识

参数:

value - 0: 否 ; 1: 是

#### 4.1.32 public int getDynamicMark()

说明:

获得是否动态标识标志位

返回:

动态标识: 0: 否 1: 是

#### 4.1.33 public void setImageFormat(byte imageFormat)

说明:

设置生成文件类型

参数:

imageFormat - 2 生成.png 文件,1 生成.gif 文件

#### 4.1.34 public byte getImageFormat()

说明:

获得生成文件类型

返回:

2 生成.png 文件,1 生成.gif 文件

#### 4.1.35 public void setMapID(long mapID)

说明:

设置地图的唯一标识

参数:

mapID - 地图的唯一标识

#### 4.1.36 public long getMapID()

说明:

获取地图的唯一标识

返回:

地图的唯一标识

#### 4.1.37 public double getEagleOrgX()

说明:

获得鹰眼显示的左上角逻辑横坐标

返回:

鹰眼显示的左上角逻辑横坐标

#### 4.1.38 public void setEagleOrgX(double eagleOrgX)

说明:

设置鹰眼显示的左上角逻辑横坐标

参数:

eagleOrgX - 鹰眼显示的左上角逻辑横坐标

#### **4.1.39 public double getEagleOrgY()**

说明:

获得鹰眼显示的左上角逻辑纵坐标

返回:

鹰眼显示的左上角逻辑纵坐标

#### **4.1.40 public void setEagleOrgY(double eagleOrgY)**

说明:

设置鹰眼显示的左上角逻辑纵坐标

参数:

eagleOrgY -鹰眼显示的左上角逻辑纵坐标

#### **4.1.41 public double getEagleZoomRate()**

说明:

获取鹰眼的显示比率

返回:

鹰眼的显示比率

#### **4.1.42 public void setEagleZoomRate(double eaglezoomRate)**

说明:

设置鹰眼的显示比率

参数:

eaglezoomRate -鹰眼的显示比率

#### **4.1.43 public CRect getMapOriginalBound()**

说明:

获取地图初始范围

返回:

地图初始范围

#### **4.1.44 public CRect getMapDisplayBound()**

说明:

获取地图当前显示范围

返回:

地图当前显示范围

#### **4.1.45 public java.net.InetSocketAddress getGisSvrMgr()**

说明:

获取 GIS 服务管理器 IP 和端口

返回:

GIS 服务管理器 IP 和端口

**4.1.46 public java.net.InetSocketAddress getGisSvr()**

说明:

取 GIS 服务器 IP 和端口

返回:

GIS 服务器 IP 和端口

**4.1.47 public com.mapgis.ims.pro.function.CWebMap getMapObj()**

说明:

获得地图对象

返回:

地图对象

**4.1.48 public com.mapgis.ims.pro.function.modules.CMpAtt getpAtt()**

说明:

获取对象的属性信息

返回:

地图对象的属性信息

**4.1.49 public CSelectResult getLastSelectResult()**

说明:

获取最后一次的查询结果

返回:

最后一次的查询结果

**4.1.50 public void setSelectResult(CSelectResult selectResult)**

说明:

设置查询结果

参数:

selectResult - 查询结果集

**4.1.51 public long getPagesCanDisplay()**

说明:

获得显示记录的页数

返回:

查询结果记录的页数

**4.1.52 public void setMapDisplayBound(CRect rect)**

说明:

设置显示范围

参数:

value - 显示范围

**4.1.53 public short GetNewMapInfo(long mapID)**

说明:

加载 mapID 对应的地图文档

参数:

mapID - 地图文档标识

返回:

1: 成功; 0: 失败

**4.1.54 public CFeatureGeometry GetFeatureGeometry (CGetFeatureInformation factor)**

说明:

根据要素信息获得要素的几何信息

参数:

featureToGet - 要素信息

返回:

要素的几何信息

示例:

```

//获取查询结果的几何信息
CFeatureGeometry featureGeometry = null;
CGetFeatureInformation featureToGet = new CGetFeatureInformation();
featureToGet.setFeatureID(featureID);
featureToGet.setLayerIndex(LayerIndex);
featureToGet.setInformationType((byte) (0));
featureToGet.setMapID(viewport.getMapID());
featureGeometry = viewport.GetFeatureGeometry(featureToGet);
if (featureGeometry == null) {
    throw new Exception("获取结果失败! 请重新查询!");
}

```

**4.1.55 public void LoadMap(String mapName)**

说明:

加载地图文档

参数:

mapName - \*.Map 地图名称

**4.1.56 public byte[] GetMapImage()**

说明:

获取地图

返回:

图片字节流

示例:

```

viewport = new MapViewPort("127.0.0.1","5141");
viewport.LoadMap("ChinaTest.map");
viewport.setImageFolder("D: \\mapimage");
viewport.setImageHeight(550);

```

```
viewport.setImageWidth(600);  
viewport.setBuildFileMode(true);  
byte[] buffer = viewport.GetMapImage();
```

#### **4.1.57 public byte[] getFeatureClassImg(String imgFileName, String hdfName, String rltfName, String netSvrName, String username, String password)**

**说明:**

根据指定 Hdf 的要素图像，得到空间分析结果的图像

**参数:**

imgFileName - 空间分析结果的图像名

hdfName - hdf 名

rltfName - 空间分析结果名

netSvrName - 远程数据库名

username - 用户名

password - 密码

**返回:**

图片字节流

#### **4.1.58 public byte[] GetMapImage(String imgFileName)**

**说明:**

获取地图

**参数:**

imgFileName - 生成地图的名称

**返回:**

图片字节流

#### **4.1.59 public byte[] GetEagleImage(short imageWidth, short imageHeight, double zoomRateIn)**

**说明:**

获取鹰眼图像

**参数:**

imageWidth - 地图显示宽度

imageHeight - 地图显示高度

zoomRateIn - 地图显示比率

**返回:**

图片字节流

#### **4.1.60 public void GetEagleImage(short imageWidth, short imageHeight, double zoomRateIn, String imageFileName)**

**说明:**

获取鹰眼图像

**参数:**

imageWidth - 地图显示宽度  
imageHeight - 地图显示高度  
zoomRateIn - 地图显示比率  
imageFileName - 鹰眼地图名称

#### **4.1.61 public short GetOnePageResult(int n, CGetOnePage page)**

**说明:**

得到指定页查询结果

**参数:**

nPage - 页码

**返回:**

成功 1,失败-1

**示例:**

```
CPoint2D center = new CPoint2D(10001,10002);  
CCircleSelectClass search = new CCircleSelectClass(center,20,viewport.getMapID());  
CGetOnePage partialSearch = new CGetOnePage();  
partialSearch.setQueryObj(search);  
short pFlag = viewport.GetOnePageResult(3, partialSearch);
```

#### **4.1.62 public short SelectByLine(CLineSelectClass selectLine)**

**说明:**

线查询

**参数:**

selectLine - 线信息

**返回:**

成功 1,失败-1

#### **4.1.63 public short SelectByHitting(CPointSelectClass selectPoint)**

**说明:**

点击查询

**参数:**

selectPoint - 所点击的点信息

**返回:**

成功 1,失败-1

**示例:**

```
CPoint2D dot = new CPoint2D(10001,10002);  
double tolerance = 10 * lengthOfOnePixel();  
CPointSelectClass search = new CPointSelectClass(dot,tolerance,viewport.getMapID());  
viewport.SelectByHitting(search);
```

#### **4.1.64 public short SelectByRect(CRectangleSelectClass selectRect)**

**说明:**

拉框查询

**参数:**

selectRect - 所拉框范围

**返回:**

成功 1,失败-1

**示例:**

```
CRect wholeMap = viewport.getMapOriginalBound();
CRectangleSelectClass search =
    new CRectangleSelectClass(wholeMap,viewport.getMapID());
viewport. SelectByRect (search);
```

**4.1.65 public short SelectByCircle(CCircleSelectClass selectCircle)****说明:**

画圆查询

**参数:**

selectCircle - 所画圆信息

**返回:**

成功 1,失败-1

**示例:**

```
CPoint2D center = new CPoint2D(10001,10002);
CCircleSelectClass search =
    new CCircleSelectClass(center,20,viewport.getMapID());
viewport. SelectByCircle (search);
```

**4.1.66 public short SelectByPolygon(cCPolygonSelectClass selectPlgon)****说明:**

多边形查询

**参数:**

selectPlgon - 多边形信息

**返回:**

成功 1,失败-1

**示例:**

```
CPolygonSelectClass search = new CPolygonSelectClass(constructAPolygon(4),
new short[]{4},viewport.getMapID());
viewport. SelectByPolygon (search);
```

**4.1.67 public short SelectByCondition(CAttSelectStru sqlQuery)****说明:**

条件查询

**参数:**

SqlQuery - 属性条件查询信息

**返回:**

成功 1,失败-1

**示例:**

```
CAttSelectStru search = new CAttSelectStru(viewport.getMapID());
search.setAttCondition("ID>0");
viewport. SelectByCondition (search);
```

#### **4.1.68 public short SelectByCondition(CCircleWithAttSelectStru**

##### **circleQ)**

**说明:**

画圆条件查询

**参数:**

circleQ - 画圆条件查询信息

**返回:**

成功 1,失败-1

**示例:**

```
CCircleWithAttSelectStru search = new CCircleWithAttSelectStru(
getPointsInBound(1)[0],30*lengthOfOnePixel(),viewport.getMapID());
search.setAttCondition("ID>0");
viewport. SelectByCondition (search);
```

#### **4.1.69 public short SelectByCondition(CRectWithAttSelectStru rectQ)**

**说明:**

拉框条件查询

**参数:**

rectSqlQuery - 拉框条件查询信息

**返回:**

成功 1,失败-1

**示例:**

```
CRectWithAttSelectStru search = new CRectWithAttSelectStru(
viewport.getMapOriginalBound(),viewport.getMapID());
search.setAttCondition("ID>0");
viewport. SelectByCondition (search);
```

#### **4.1.70 public short SelectByCondition(CPolygonWithAttSelectStru pgQ)**

**说明:**

多边形条件查询

**参数:**

pgQ - 多边形条件查询信息

**返回:**

成功 1,失败-1

**示例:**

```
CPolygonWithAttSelectStru search = new CPolygonWithAttSelectStru(
constructAPolygon(4),new short[] {4},viewport.getMapID());
search.setAttCondition("ID>0");
viewport. SelectByCondition (search);
```

#### 4.1.71 public short executeSelect(SelectObj obj)

说明:

可执行除线查询相关外的任何查询 参数为除线查询相关任意查询结构体

参数:

obj – 所有查询结构体的父类

#### 4.1.72 public void AppendOnePoint(CAppendAPoint thePoint)

说明:

添加点

参数:

thePoint - 点结构体对象

示例:

```
final long pointLayerIndex = getPointLayerIndex();
CAppendAPoint point = new CAppendAPoint();
CPoint2D position = getPointsInBound(1)[0];
point.setLayerIndex(pointLayerIndex);
point.setMapID(viewport.getMapID());
point.setThePoint(position);
point.setPicAngle(10);
viewport.AppendOnePoint(point);
```

#### 4.1.73 public void AppendOneArc(CAppendAArcLine theArc)

说明:

添加弧段

参数:

theArc - 弧段结构体对象

示例:

```
final long lineLayerIndex = getlineLayerIndex();
CAppendAArcLine arc = new CAppendAArcLine();
CPoint2D[] linePoints = getPointsInBound(3);
CLine line = new CLine(linePoints);
arc.setLayerIndex(lineLayerIndex);
arc.setTheArc(line);
arc.setMapID(viewport.getMapID());
viewport.AppendOneArc(arc);
```

#### 4.1.74 public void AppendOneRegion(CAppendARegion theRegion)

说明:

添加区

参数:

theRegion - 区结构体对象

示例:

```
final long regionLayerIndex = getRegionLayerIndex();
```

```

CAppendARegion region = new CAppendARegion();
region.setLayerIndex(regionLayerIndex);
region.setMapID(viewport.getMapID());
CRegion2D region2d = new CRegion2D();
region2d.setArcNumber((short)1);
CPoint2D[] points = getPointsInBound(4);
CLine[] lines = new CLine[1];
lines[0] = new CLine(points);
region2d.setTheArcs(lines);
region.setTheRegion(region2d );
region.setPicAngle(10f);
viewport.AppendOneRegion(region);

```

#### 4.1.75 public CAppendOperResult appendOneElement(XMLSerializableObj obj)

**说明:**

可执行添加点、添加线、添加区功能

**参数:**

obj - 添加要素的结构体

**返回:**

添加的要素的信息，包含要素 ID

**示例:**

```

CAppendOperResult result = viewport.appendOneElement(appendObj);
long featureID = result.getFeatureID();

```

#### 4.1.76 public CUniformResult DeleteOneFeature(CDeleteFeature feature)

**说明:**

删除要素

**参数:**

feature - 删除的要素信息

**返回:**

CUniformResult 检查操作是否成功

**示例:**

```

CDeleteFeature element = new CDeleteFeature();
long id = elementID;
element.setFeatureID(id);
element.setMapID(viewport.getMapID());
element.setLayerIndex(layerIndex);
try{
    CUniformResult result = viewport.DeleteOneFeature(element);
    assertEquals("SUCCESS",result.getMessage());
} catch(Exception e){
    fail();
}

```

**4.1.77 public CUniformResult UpdateFeature(CUpdateFeatureAttribute o)**

说明:

更新要素

参数:

o - 要更新的要素信息

返回:

CUniformResult

示例:

```

//定义更新要素类对象、返回值对象
CUpdateFeatureAttribute newAttribute = new CUpdateFeatureAttribute();
CUniformResult updateRlt = null;
//为新要素对象赋值
newAttribute.setFieldNumber((short) fldNum); //字段个数

newAttribute.setNewFieldValues(fldVal); //字段值
newAttribute.setLayerIndex(LayerIdx); //要素所在图层
newAttribute.setFeatureID(featureID); //要素ID号
newAttribute.setMapID(viewport.getMapID());
//调用要素更新函数
try
{
updateRlt = viewport.UpdateFeature(newAttribute);
}
catch(Exception e)
{
    request.setAttribute("tip", "要素类文件修改失败。");
    return (mapping.findForward("操作异常, 弹出提示"));
}

```

**4.1.78 public CUniformResult UpdateSpatialFeature(CUpdateFeature o)**

说明:

更新空间要素

参数:

newSpatialInfo - 要更新的空间要素信息

返回:

CUniformResult 类对象

**4.1.79 public CLayerAttStructure GetFeaturelayerAttributeStructure(long layer Index, long mapID)**

说明:

获取图层属性结构

参数:

layerIndex - 图层索引

mapID - 地图 ID

返回:

CLayerAttStructure 返回图层属性结构

示例:

```

CLayerAttStructure AttStru = new CLayerAttStructure();
try
{
AttStru = viewport.GetFeaturelayerAttributeStructure(1,viewport.getMapID());
}
catch(Exception e)
{
request.setAttribute("tip","请先提交图层。");
return (mapping.findForward("操作异常,弹出提示"));
}

```

#### 4.1.80 public short UpdateLayer(CLayerInformation layerObj)

说明:

更新图层信息

参数:

layerObj - 图层对象信息

返回:

成功 1,失败-1

示例:

//显示图层的数目

```

if(strLayerNoStr1.length()>0)
{
strLayerNoStr1 = strLayerNoStr1.substring(0,strLayerNoStr1.length()-1);
strLayerNoStr2 = strLayerNoStr1.split("|");
for(int i=0;i<strLayerNoStr2.length;i++)
{
strLayerNo = strLayerNoStr2[i];
if(strLayerNo.compareTo("") == 0||strLayerNo.compareTo("|") == 0)
continue;
CLayerInformation mapLayerObj = null;
mapLayerObj = getMapLayerObj(strLayerNo);
mapLayerObj.setVisible((short)1);
viewport.UpdateLayer(mapLayerObj);
}
}

```

#### 4.1.81 public short UpdateLayerLst(CUpdateLayerLst layerObj)

说明:

更新图层组信息

参数:

layerObj - 图层对象信息

返回:

成功 1,失败-1

#### 4.1.82 public CUniformResult GetGeomInfo(CGeomInfo geomInfo)

说明:

根据图层索引号得到图层信息，包括图层名称、图层类型

**参数:**

geoInfo - 地理数据库信息

**返回:**

返回 CUniformResult(flag>0 代表它所属的类型)

#### **4.1.83 public CUniformResult DeleteLayer(CGeomInfo geomInfo)**

**说明:**

根据图层索引号删除某个图层

**参数:**

geomInfo -

**返回:**

返回 CUniformResult(flag==1 代表删除成功)

#### **4.1.84 public CUniformResult AddLayer(CAddLayerInfo addLayerInfo)**

**说明:**

添加图层

**参数:**

addLayerInfo -

**返回:**

返回 CUniformResult(flag==1 代表成功)

#### **4.1.85 public CUniformResult GetGDBInfo(CGetGDBInfo gdbInfo)**

**说明:**

获取取所有数据源的名称

**参数:**

GDBInfo -

**返回:**

返回 CUniformResult(flag==1 代表成功,包含所有数据源名,格式为: 数据源名,数据源名,数据源名)

#### **4.1.86 public CUniformResult GetGDBName(CGetGDBName gdbName)**

**说明:**

获取 GDB 名称

**参数:**

GDBName - GDB 数据源名称

**返回:**

返回 CUniformResult(flag==1 代表成功,包含所有 GDB 名,格式为: GDB 名,GDB 名,GDB 名)

#### **4.1.87 public CUniformResult GetGDBLayerInfo(CGetGDBLayerInfo layer)**

**说明:**

获取 GDB 图层信息

参数:

gdbLayerInfo -

返回:

返回 CUniformResult(flag=1 代表成功,包含所有图层名,格式为: 图层名,图层名,图层名)

#### 4.1.88 public CSpatialOperResult SpatialAnalysisOverlay(CSpatialOverlay overlay)

说明:

空间分析中的叠加操作

参数:

overlay - 叠加分析对象

返回:

CSpatialOperResult 空间分析结果

示例:

```
//声明叠加分析对象
CSpatialOverlay overlay = new CSpatialOverlay();
CSpatialOperResult SpatialOperRlt = null;

//为叠加分析的两个图层赋值
overlay.setLayerIndex1(-1);
overlay.setLayerIndex2(-1);
overlay.setGDBDataSourceName1(GDBDataSourceName);
overlay.setGDBName1(GDBName+".HDF");
overlay.setGDBName2("");
overlay.setGDBDataSourceName2("");
overlay.setFeatureIDSLst2("");
overlay.setFeatureIDSLst1(layerName1 + "," + layerName2);
//叠加分析的类型,如并、差、交等
overlay.setOverlayType(Short.parseShort(operation));
//叠加分析后生成的文件名
overlay.setFeatureClassName(fcLsName);
overlay.setMapID(viewport.getMapID());
//叠加分析的返回结果对象
try
{
    SpatialOperRlt = viewport.SpatialAnalysisOverlay(overlay);
}
catch(Exception e)
{
    SpatialOperRlt = null;
}
```

#### 4.1.89 public CSpatialOperResult SpatialAnalysisPolygonClip(CSpatialPolygonClip polygonClip)

说明:

空间分析中的多边形裁剪

参数:

polygonClip - 多边形裁剪对象

返回:

CSpatialOperResult 空间分析结果

示例:

```

CSpatialPolygonClip polygonClip = new CSpatialPolygonClip();
//得到客户端点击点个数
String PointOfPolygon = request.getParameter("Points");
String[] PpointsCoor = PointOfPolygon.split(",");
//申请相应空间来存放由客户端传过来的点坐标集,此时已是逻辑坐标
short PntNum = (short) PpointsCoor.length;

CPoint2D[] PdotLst = new CPoint2D[PntNum / 2];
for (int i = 0; i < PntNum / 2; i++) {
    PdotLst[i] = new CPoint2D();
    PdotLst[i].setPointX(Double.parseDouble(PpointsCoor[2 * i]));
    PdotLst[i].setPointY(Double.parseDouble(PpointsCoor[2 * i + 1]));
}

polygonClip.setPointNumber((short) (PdotLst.length));
polygonClip.setPoints(PdotLst);
polygonClip.setClipType((short) 1);
polygonClip.setFeatureClassName("polygonClip"+ String.valueOf(rand.nextDouble()*1000));
polygonClip.setMapID(MapID);
polygonClip.setLayerIndex(Long.parseLong(layerIdx));
polygonClip.setLayerIndexLst(session.getAttribute("layerIndexLst").toString());
SpatialOperRlt = viewport.SpatialAnalysisPolygonClip(polygonClip);

```

#### 4.1.90 public CSpatialOperResult SpatialAnalysisCircleClip(CSpatialCircleClip circleClip)

说明:

空间分析中的圆裁剪

参数:

circleClip - 圆裁剪对象

返回:

CSpatialOperResult 空间分析结果

示例:

```

//声明圆裁剪对象、返回值对象
CSpatialCircleClip circleClip = new CSpatialCircleClip();
//接收客户端传来的圆心、半径及裁剪半径
CPoint2D dotCenter = new CPoint2D();
dotCenter.setPointX(xCir);
dotCenter.setPointY(yCir);
circleClip.setCenter(dotCenter);
circleClip.setRadius(radius);
circleClip.setMapID(MapID);
circleClip.setLayerIndex(Long.parseLong(layerIdx));
circleClip.setLayerIndexLst(session.getAttribute("layerIndexLst").toString());
//裁剪类型
circleClip.setClipType((short) 1);
circleClip.setFeatureClassName("CircleClip" + String.valueOf(rand.nextDouble()*1000));
SpatialOperRlt = viewport.SpatialAnalysisCircleClip(circleClip);
if(SpatialOperRlt==null)
{
    flg = "-1";
}

```

**4.1.91 public CSpatialOperResult SpatialAnalysisBuffer(CSpatialBuffer buffer)**

说明:

空间分析中的缓冲区分析

参数:

buffer - 缓冲分析对象

返回:

CSpatialOperResult 空间分析结果

示例:

```

//声明缓冲区分析对象、返回值描述对象
CSpatialBuffer buffer = new CSpatialBuffer();
Random rand = new Random();
//缓冲区分析后生成文件的名称
buffer.setFeatureClassName("buffer" + String.valueOf(rand.nextDouble() * 1000));
//要进行缓冲区分析的要素实体号
buffer.setFeatureIDS(featureIDS);
//要进行缓冲区分析的要素个数
buffer.setFeatureNumber({short} (featureIDS.length));
//缓冲区分析半径
    buffer.setRadius(radius);
//要进行缓冲区分析的图层索引号
buffer.setLayerIndex(layerSelectRlt[0].getLayerIndex());
//调用缓冲区分析函数,并通过返回值验证分析是否成功
buffer.setMapID(viewport.getMapID());
CSpatialOperResult SpatialOperRlt = null;
String flg;
try
{
    SpatialOperRlt = viewport.SpatialAnalysisBuffer(buffer);
}
catch(Exception e)
{
    SpatialOperRlt=null;
}

```

**4.1.92 public void GetSpatialAnalysisResult(CSpatialOperResult result)**

已过时。得到空间分析的结果,已废用

参数:

result - 空间分析结果对象

**4.1.93 public CNetAnalysisResult NetAnalysis(CBasicNetAnalysis net)**

说明:

路径分析

参数:

net - 网络分析对象

返回:

CNetAnalysisResult 网络分析结果对象

**4.1.94 public CPrjTranRlt TransPrj(CPrjTran prjTran)**

说明:

坐标投影转换

**参数:**

坐标及投影信息 -

**返回:**

成功 1,失败-1

#### **4.1.95 public COverlayByIDPntSeq SpatialOverlayByID(CSpatialOverlayByID overlaybyID)**

**说明:**

根据要素的标识进行要素间的叠加，并且返回包含叠加结果的几何信息

**参数:**

overlaybyID – 要素叠加的结构体

**返回:**

包含叠加结果的几何信息

#### **4.1.96 public String gisCreateAbsAndViewImgName(int format)**

**说明:**

创建唯一的图像名称

**参数:**

format - 生成图像类型

**返回:**

地图名称

#### **4.1.97 public COverlayByIDPntSeq SpatialOverlayByGeom (CSpatialOverlayByGeom overlaybyGeom)**

**说明:**

根据两个要素的几何信息以及叠加类型，得到叠加结果的几何信息

**参数:**

overlaybyGeom – 包含两个要素的几何信息以及叠加类型的结构体

**返回:**

叠加结果的几何信息

#### **4.1.98 public void setDisplayedPerPage(int count)**

**说明:**

设置每次查询从服务器返回的最大记录数目

**参数:**

count– 每次最多返回的记录条数

#### **4.1.99 public CPoint2D WpToLp(CPoint2D windowPoint)**

**说明:**

窗口坐标转逻辑坐标

**参数:**

windowPoint - 窗口坐标

返回： 逻辑坐标

#### **4.1.100 public void loadMapResultsSave(boolean flag)**

说明：

是否缓存地图文档信息

参数：

flag - ture: 是; false: 否

#### **4.1.101 public void setLogFolder(java.lang.String path)**

说明：

设置操作日志存放路径

参数：

path - 如 C: \temp 文件下

#### **4.1.102 public int getRecordsPerPage()**

说明：

得到每次返回的查询结果最大条数

返回：

返回的查询结果最大条数

#### **4.1.103 public void enableResizeBufferRlt()**

说明：

使缓冲区分析结果与原先地图位置相符合

#### **4.1.104 public void disableResizeBufferRlt()**

说明：

仅显示缓冲区结果图像

#### **4.1.105 public CFeatureStyleInfo getFeatureStyleInfo(CGetFeatureStyleInfo factorInfo)**

说明：

根据要素信息获取要素的样式信息

参数：

要素信息

返回：

要素的样式信息

#### **4.1.106 public CUniformResult updateFeatureStyleInfo (CUpdateFeatureStyleInfo updateInfo)**

说明：

更新要素的样式信息

参数：

要素样式信息

**返回:**

返回 CUniformResult 对象，包含操作成功与否标志位

#### **4.1.107 public int getLayerTypeByIndex(int layerIndex)**

**说明:**

得到索引号对应的图层数据类型

**参数:**

layerIndex - 图层索引号

**返回:**

4: 点图层; 5: 线图层; 6: 区图层

#### **4.1.108 public CRect getFclsRect(CGetFclsRect rect)**

**说明:**

得到指定 Hdf 中的要素的范围

**参数:**

rect -指定 Hdf 中的要素

**返回:**

该要素的逻辑范围

#### **4.1.109 public CSelectResult featureClassSelect(CGetFclsSel fclsSel)**

**说明:**

得到指定 hdf 的要素的信息

**参数:**

fclsSel - Hdf 的信息

**返回:**

该 Hdf 中的所有要素信息

#### **4.1.110 public CFeatureGeometry getSingleFclsGeometry (CGetSingleFclsGeo geo)**

**说明:**

得到指定 hdf 的要素的空间属性信息

**参数:**

geo -指定 hdf 的要素信息

**返回:**

该要素的空间几何信息

#### **4.1.111 public String getSpatialOpRltHdfName()**

**说明:**

返回最后一次空间分析的结果 HDF 名

**返回:**

最后一次空间分析的结果 HDF 名

#### 4.1.112 public getSpatialOpFclsName()

说明:

返回最后一次空间分析的结果要素名

返回:

最后一次空间分析的结果要素名

## 4.2 CAddLayerInfo 类

添加图层信息类

#### 4.2.1 public CAddLayerInfo()

说明:

构造函数

参数:

无

返回值:

无

#### 4.2.2 public void setMapID(long mapID)

说明:

设置要更新属性的要素字段所在地图文档 ID

参数:

mapID: 地图文档 ID

返回值:

无

#### 4.2.3 public long getMapID()

说明:

获取要更新属性的要素字段所在地图文档 ID

参数:

无

返回值:

地图文档 ID

#### 4.2.4 public void setDBName(java.lang.String dbName)

说明:

设置要添加图层的地图文档所在的 DB 数据源

参数:

dbName: 地图文档所在的 DB 数据源名称

返回值:

无

#### 4.2.5 public java.lang.String getDBName()

说明:

获取要添加图层的地图文档所在的 DB 数据源

参数:

无

返回值:

地图文档所在的 DB 数据源名称

#### 4.2.6 public void setUsername(java.lang.String userName)

说明:

设置远程用户名

参数:

userName: 地图文档所在的 GDB 数据源名称

返回值:

无

#### 4.2.7 public java.lang.String getUsername()

说明:

获取远程用户名, 若为本地用户则该属性为空

参数:

无

返回值:

远程用户名

#### 4.2.8 public void setPassword(java.lang.String passWord)

说明:

设置要添加图层的地图文档所在的 GDB 数据源密码

参数:

passWord: 地图文档所在的 GDB 数据源密码

返回值:

无

#### 4.2.9 public java.lang.String getPassword()

说明:

获取要添加图层的地图文档所在的 GDB 数据源密码

参数:

无

返回值:

地图文档所在的 GDB 数据源密码

#### 4.2.10 public void setGDBName(java.lang.String gdbName)

说明:

设置要添加图层的地图文档所在的 GDB 数据源

**参数:**

gdbName: 地图文档所在的 GDB 数据源名称

**返回值:**

无

#### **4.2.11 public java.lang.String getGDBName()**

**说明:**

获取要添加图层的地图文档所在的 GDB 数据源

**参数:**

无

**返回值:**

地图文档所在的 GDB 数据源名称

#### **4.2.12 public void setLayerIndex(long layerIndex)**

**说明:**

设置要添加图层的索引号

**参数:**

layerIndex: 所添加图层的索引号

**返回值:**

无

#### **4.2.13 public long getLayerIndex()**

**说明:**

获取要添加图层的索引号

**参数:**

无

**返回值:**

图层索引号

#### **4.2.14 public void setLayerInformation(CLayerInformation layerInformation)**

**说明:**

设置要添加的图层信息

**参数:**

layerInformation: 图层信息, 具体参见 CLayerInformation 类

**返回值:**

无

#### **4.2.15 public CLayerInformation getLayerInformation()**

**说明:**

获取添加的图层信息

**参数:**

无

返回值：  
图层信息

## 4.3 CAppendArcLine 类

添加线要素类。

示例：

```
void viewport.AppendOneArc(CAppendArcLine theArc)
```

说明：

函数的入口参数。

### 4.3.1 public CAppendArcLine()

说明：

添加线要素对象的构造函数

### 4.3.2 public void setLayerIndex(long layerIndex)

说明：

设置要添加线要素的图层索引号

参数：

layerIndex: 图层索引号

### 4.3.3 public long getLayerIndex()

说明：

获取线要素的图层索引号

### 4.3.4 public void setPropertyNumber(short propertyNumber)

说明：

设置要添加线要素的属性字段个数

参数：

propertyNumber: 属性字段个数

### 4.3.5 public short getPropertyNumber()

说明：

获取线要素的属性字段个数

### 4.3.6 public void setPropertyValues(java.lang.String[] propertyValues)

说明：

设置要添加线要素的属性字段值，数组个数由 propertyNumber 属性决定。

参数：

propertyValues: 属性字段值

**4.3.7 public java.lang.String[] getPropertyValues()**

说明:

获取要添加线要素的属性字段值

**4.3.8 public void setTheArc(CLine theArc)**

说明:

设置要添加线要素的点集合对象。

参数:

theArc: 线要素的点集合对象

**4.3.9 public CLine getTheArc()**

说明:

获取线要素的点集合对象

**4.3.10 public void setLineType(int lineType)**

说明:

设置要添加线要素的线类型

参数:

lineType: 线要素的线类型

**4.3.11 public int getLineType()**

说明:

获取要添加线要素的线类型

**4.3.12 public void setPenWidth(float penWidth)**

说明:

设置要添加线要素的外部笔宽

参数:

penWidth: 线要素的外部笔宽

**4.3.13 public float getPenWidth()**

说明:

获取要添加线要素的外部笔宽

**4.3.14 public void setLinColor(int linColor)**

说明:

设置要添加线要素的线颜色

参数:

linColor: 线要素的线颜色

#### 4.3.15 public int getLinColor()

说明:

获取要添加线要素的线颜色

#### 4.3.16 public void setLineRateX(float lineRateX)

说明:

设置要添加线要素的线比例 X 系数

参数:

lineRateX: 线要素的线比例 X 系数

#### 4.3.17 public float getLineRateX()

说明:

获取要添加线要素的线比例 X 系数

#### 4.3.18 public void setlineRateY(float lineRateY)

说明:

设置要添加线要素的线比例 Y 系数

参数:

lineRateY: 线要素的线比例 Y 系数

#### 4.3.19 public float getlineRateY()

说明:

获取要添加线要素的线比例 Y 系数

#### 4.3.20 public void setMapID(long mapID)

说明:

设置要添加线要素所在的地图文档 ID

参数:

mapID: 地图文档 ID

#### 4.3.21 public long getMapID()

说明:

获取地图文档 ID

### 4.4 CAppendAPoint 类

添加点要素类。

示例:

```
void viewport.AppendOnePoint(CAppendAPoint thePoint)
```

说明:

函数的入口参数。

#### **4.4.1 public CAppendAPoint()**

说明:

添加点要素对象的构造函数

#### **4.4.2 public float getPicAngle()**

说明:

获取添加点要素的角度

#### **4.4.3 public void setPicAngle(float picAngle)**

说明:

设置添加点要素的图片角度

参数:

picAngle: 点要素的图片角度

#### **4.4.4 public void setLayerIndex(long layerIndex)**

说明:

设置要添加点要素的图层索引号

参数:

layerIndex: 图层索引号

#### **4.4.5 public long getLayerIndex()**

说明:

获取要添加点要素的图层索引号

#### **4.4.6 public void setPropertyNumber(short propertyNumber)**

说明:

设置要添加点要素的属性字段个数

参数:

propertyNumber: 属性字段个数

#### **4.4.7 public short getPropertyNumber()**

说明:

获取要添加点要素的属性字段个数

#### **4.4.8 public void setPropertyValues(java.lang.String[] propertyValues)**

说明:

设置要添加点要素的属性字段值，数组个数由 PropertyNumber 属性决定

参数:

propertyValues: 要素的属性字段值

**4.4.9 public java.lang.String[] getPropertyValues()**

说明:

获取要添加点要素的属性字段值

**4.4.10 public void setThePoint(CPoint2D thePoint)**

说明:

设置要添加点要素的点集合对象

参数:

thePoint: 点要素的点集合对象

**4.4.11 public CPoint2D getThePoint()**

说明:

获取要添加点要素的点集合对象

**4.4.12 public void setPicType(int picType)**

说明:

设置地图子图号

参数:

picType: 点要素的子图号

**4.4.13 public int getPicType()**

说明:

返回地图子图号

**4.4.14 public void setPicHeight(float picHeight)**

说明:

设置地图子图的高度, 默认单位为米

参数:

picHeight: 地图子图的高度

**4.4.15 public float getPicHeight()**

说明:

返回地图子图的高度

**4.4.16 public void setPicWidth(float picWidth)**

说明:

设置地图子图的宽度, 默认单位为米

参数:

picWidth: 地图子图的宽度

#### 4.4.17 public float getPicWidtht()

说明:

获取地图子图的宽度

#### 4.4.18 public void setPicColor(int picColor)

说明:

设置地图子图的颜色

参数:

picColor: 地图子图的颜色

#### 4.4.19 public int getPicColor()

说明:

获取地图子图的颜色

#### 4.4.20 public void setMapID(long mapID)

说明:

设置要添加点要素所在的地图文档 ID

参数:

mapID: 地图文档 ID

#### 4.4.21 public long getMapID()

说明:

获取要添加点要素所在的地图文档 ID

### 4.5 CAppendARegion 类

添加区要素类。

示例:

void viewport. AppendOneRegion(CAppendAPoint thePoint)函数的入口参数。

#### 4.5.1 public CAppendARegion()

说明:

添加区要素对象的构造函数

#### 4.5.2 public float getPicAngle()

说明:

获取添加区要素的角度

#### 4.5.3 public void setPicAngle(float picAngle)

说明:

设置添加区要素的角度

参数:

picAngle: 区要素的角度

#### **4.5.4 public void setLayerIndex(long layerIndex)**

说明:

设置要添加区要素的图层索引号

参数:

layerIndex: 图层索引号

#### **4.5.5 public long getLayerIndex()**

说明:

获取要添加区要素的图层索引号

#### **4.5.6 public void setPropertyNumber(short propertyNumber)**

说明:

设置要添加区要素的属性字段个数

参数:

propertyNumber: 属性字段个数

#### **4.5.7 public short getPropertyNumber()**

说明:

获取要添加区要素的属性字段个数

#### **4.5.8 public void setPropertyValues(java.lang.String[] propertyValues)**

说明:

设置要添加区要素的属性字段值，数组个数由 PropertyNumber 属性决定

参数:

propertyValues: 要素的属性字段值

#### **4.5.9 public java.lang.String[] getPropertyValues()**

说明:

获取要添加区要素的属性字段值

#### **4.5.10 public void setTheRegion(CRegion2D theRegion)**

说明:

设置要添加区要素的点集合对象

参数:

theRegion: 区要素的点集合对象

#### **4.5.11 public CRegion2D getTheRegion()**

说明:

获取要添加区要素的点集合对象

**4.5.12 public void setRegionPic(int regionPic)**

说明:

设置要添加区要素的区填充图案

参数:

regionPic: 区填充图案编号

**4.5.13 public int getRegionPic()**

说明:

获取要添加区要素的区填充图案

**4.5.14 public void setRegionColor(int regionColor)**

说明:

设置要添加区要素的区填充色

参数:

regionColor: 区要素的区填充色

**4.5.15 public int getRegionColor()**

说明:

获取要添加区要素的区填充色

**4.5.16 public void setPicHeight(float picHeight)**

说明:

设置要添加区要素的图案高

参数:

picHeight: 区要素的图案高度

**4.5.17 public float getPicHeight()**

说明:

获取要添加区要素的图案高

**4.5.18 public void setPicWidth(float picWidth)**

说明:

设置要添加区要素的图案宽

参数:

picWidth: 区要素的图案宽

**4.5.19 public float getPicWidth()**

说明:

返回要添加区要素的图案宽

#### 4.5.20 public void setPicColor(int picColor)

说明:

设置要添加区要素的区填充色

参数:

picColor: 区填充色

#### 4.5.21 public int getPicColor()

说明:

获取要添加区要素的区填充色

#### 4.5.22 public void setMapID(long mapID)

说明:

设置要添加区要素所在的地图文档 ID

参数:

mapID: 地图文档 ID

#### 4.5.23 public long getMapID()

说明:

获取要添加区要素所在的地图文档 ID

## 4.6 CAttSelectStru 类

条件查询类。

示例:

viewport.SelectByCondition (CAttSelectStru queryString)函数的入口参数

#### 4.6.1 public CAttSelectStru(long mapID)

说明:

初始化条件查询类

参数:

mapID: 地图文档 ID

#### 4.6.2 public CAttSelectStru()

说明:

初始化条件查询类

参数:

无

#### 4.6.3 public void setLayerIndex(long layerIndex)

说明:

设置要添加区要素的图层索引号

**参数:**

layerIndex: 图层索引号

#### **4.6.4 public long getLayerIndex()**

**说明:**

获取要添加区要素的图层索引号

#### **4.6.5 public void setAttCondition(java.lang.String sqlQuery)**

**说明:**

设置条件查询的条件语句

**参数:**

sqlQuery: 条件查询的条件语句

#### **4.6.6 public java.lang.String getAttCondition()**

**说明:**

获取条件查询的条件语句

#### **4.6.7 public void setMapID(long mapID)**

**说明:**

设置要条件查询的要素所在的地图文档 ID

**参数:**

mapID: 地图文档 ID

#### **4.6.8 public long getMapID()**

**说明:**

获取要条件查询的要素所在的地图文档 ID

#### **4.6.9 public void setPageIndex(long pageIndex)**

**说明:**

设置条件查询结果中分页显示第几页的首记录索引号

**参数:**

pageIndex: 结果分页显示第几页的首记录索引号

#### **4.6.10 public void setRevNum(int number)**

**说明:**

设置每次返回的最大查询结果条数

**参数:**

number: 每次返回的最大查询结果条数

#### **4.6.11 public int getRevOnce()**

**说明:**

获取每次返回的最大查询结果条数

## 4.7 CBasicNetAnalysis 类

路径分析类。

示例：

```
CNetAnalysis result = viewport. NetAnalysis (CBasicNetAnalysis net)
```

说明：

函数的入口参数。

### 4.7.1 public CBasicNetAnalysis()

说明：

构造函数

参数：

无

### 4.7.2 public void setLayerIndex(long layerIndex)

说明：

设置进行路径分析的图层索引号

参数：

layerIndex： 图层索引号

### 4.7.3 public long getLayerIndex()

说明：

获取进行路径分析的图层索引号

### 4.7.4 public void setAccessNodeNum(int NodeNum)

说明：

设置进行路径分析的节点个数

参数：

NodeNum： 节点个数

### 4.7.5 public int getAccessNodeNum()

说明：

获取进行路径分析的节点个数

### 4.7.6 public void setRadius(double radius)

说明：

设置路径分析误差半径

参数：

radius： 路径分析误差半径

**4.7.7 public double getRadius()**

说明:

获取路径分析误差半径

**4.7.8 public void setAccessNodes(CPoint2D[] nodes)**

说明:

设置路径分析节点数组

参数:

nodes: 路径分析节点数组

**4.7.9 public CPoint2D[] getAccessNodes()**

说明:

获取路径分析节点数组

**4.7.10 public void setObstacleNodes(CPoint2D[] nodes)**

说明:

设置进行路径分析的障碍点数组

参数:

nodes: 障碍点数组

**4.7.11 public CPoint2D[] getObstacleNodes()**

说明:

获取进行路径分析的障碍点数组

**4.7.12 public void setMapID(long mapID)**

说明:

设置要路径分析所在的地图文档 ID

参数:

mapID: 地图文档 ID

**4.7.13 public long getMapID()**

说明:

获取要路径分析所在的地图文档 ID

**4.7.14 public void setObstacleNodeNum(int value)**

说明:

设置进行路径分析的障碍节点个数

参数:

value: 障碍节点个数

#### 4.7.15 public int getObstacleNodeNum()

说明:

获取进行路径分析的障碍节点个数

参数:

获取进行路径分析的障碍节点个数

### 4.8 CCircleSelectClass 类

地图文档/图层的圆查询类。

示例:

```
void viewport.SelectByCircle(CCircleSelectClass CircleSelect)
```

说明:

函数的入口参数。

#### 4.8.1 public CCircleSelectClass()

说明:

圆查询类的构造函数

#### 4.8.2 public CCircleSelectClass(CPoint2D center, double radius, long mapID)

说明:

初始化圆查询类

参数:

center: 圆心

radius: 半径

mapID: 地图文档号

#### 4.8.3 public void setLayerIndex(long layerIndex)

说明:

设置圆查询的图层索引号

参数:

layerIndex: 图层索引号

#### 4.8.4 public long getLayerIndex()

说明:

获取圆查询的图层索引号

#### 4.8.5 public void setCenter(CPoint2D center)

说明:

设置圆查询的圆心坐标

参数:

center: 圆查询的圆心坐标

**4.8.6 public CPoint2D getCenter()**

说明:

获取圆查询的圆心坐标

**4.8.7 public void setRadius(double radius)**

说明:

设置要进行圆查询的查询半径

参数:

radius: 圆查询的查询半径

**4.8.8 public double getRadius()**

说明:

获取要进行圆查询的查询半径

**4.8.9 public void setMapID(long mapID)**

说明:

设置地图标识

参数:

mapID: 地图文档 ID

**4.8.10 public long getMapID()**

说明:

获取地图标识

**4.8.11 public void setPageIndex(long pageIndex)**

说明:

设置要进行圆查询的分页显示中第几页的首记录索引号

参数:

pageIndex: 查询记录的起始索引号

**4.8.12 public long getPageIndex()**

说明:

获取圆查询结果记录的起始索引号

**4.8.13 public void setRevNum(int number)**

说明:

设置圆查询每次返回的最大查询结果条数

参数:

number: 每次返回的最大查询结果条数

#### 4.8.14 public int getRevOnce()

说明:

获取每次圆查询最大结果条数

### 4.9 CCircleWithAttSelectStru 类

画圆条件查询类。

示例:

```
viewport. SelectByCondition (CcircleWithAttSelectStru circleAndAtt)
```

说明:

函数的入口参数。

#### 4.9.1 public CCircleWithAttSelectStru(CPoint2D center, double radius, long map ID)

说明:

初始化画圆条件查询类

参数:

center: 圆心

radius: 半径

mapID: 地图文档号

#### 4.9.2 public CCircleWithAttSelectStru()

说明:

初始化画圆条件查询类

参数:

无

#### 4.9.3 public void setLayerIndex(long layerIndex)

说明:

设置圆条件查询的图层索引号

参数:

layerIndex: 图层索引号

#### 4.9.4 public long getLayerIndex()

说明:

获取圆条件查询的图层索引号

#### 4.9.5 public void setAttCondition(java.lang.String sqlQuery)

说明:

设置圆条件查询的条件语句

参数:

sqlQuery: 查询的条件语句

#### **4.9.6 public java.lang.String getAttCondition()**

说明:

获取圆条件查询的条件语句

#### **4.9.7 public void setMapID(long mapID)**

说明:

设置地图标识

参数:

mapID: 地图文档 ID

#### **4.9.8 public long getMapID()**

说明:

获取地图标识

#### **4.9.9 public void setCenter(CPoint2D center)**

说明:

设置圆条件查询的圆心点

参数:

center: 圆查询的圆心坐标

#### **4.9.10 public CPoint2D getCenter()**

说明:

获取圆条件查询的圆心点

#### **4.9.11 public void setRadius(double radius)**

说明:

设置圆条件查询的圆半径

参数:

radius: 圆条件查询的查询半径

#### **4.9.12 public double getRadius()**

说明:

获取圆条件查询的圆半径

#### **4.9.13 public void setPageIndex(long pageIndex)**

说明:

设置要进行圆条件查询的分页显示中第几页的首记录索引号

参数:

pageIndex: 查询记录的起始索引号

#### 4.9.14 public long getPageIndex()

说明:

获取圆条件查询的分页显示中第几页的首记录索引号

#### 4.9.15 public void setRevNum(int number)

说明:

设置圆条件查询每次返回的最大查询结果条数

参数:

number: 每次返回的最大查询结果条数

#### 4.9.16 public int getRevOnce()

说明:

获取圆条件查询每次返回的最大查询结果条数

### 4.10 CDeleteFeature 类

要素删除类。

示例:

```
CUniformResult viewport.DeleteOneFeature(CDeleteFeature theFeature)
```

说明:

函数的入口参数。

#### 4.10.1 public CDeleteFeature()

说明:

删除要素的构造函数

参数:

无

#### 4.10.2 public void setLayerIndex(long layerIndex)

说明:

设置要删除要素的图层索引号

参数:

layerIndex: 图层索引号

#### 4.10.3 public long getLayerIndex()

说明:

获取要删除要素的图层索引号

#### 4.10.4 public void setFeatureID(long featureID)

说明:

设置要删除要素的要素 ID

参数:

featureID: 所要删除的要素 ID

#### **4.10.5 public long getFeatureID()**

说明:

获取要删除要素的要素 ID

#### **4.10.6 public void setDelGeometry(byte delGeometry)**

说明:

设置删除要素标记 1: 删除 0: 不删除

参数:

delGeometry: 删除要素标记

#### **4.10.7 public byte getDelGeometry()**

说明:

获取删除要素标记

#### **4.10.8 public void setMapID(long mapID)**

说明:

设置要删除要素的地图标识

参数:

mapID: 地图文档 ID

#### **4.10.9 public long getMapID()**

说明:

获取要删除要素的地图标识

### **4.11 CDrawMap 类**

画图类。

示例:

```
CDrawMap drawMap = new CDrawMap();  
byte[] bufferArray = drawMap.CreateMap(message,strPath,w, h);
```

#### **4.11.1 public CDrawMap()**

说明:

初始化画图类

#### **4.11.2 public byte[] CreateMap(String strContent, String strPath, int width, int height)**

说明:

初始化画图类

**参数:**

strContent: 图片显示内容

width: 图片宽

height: 图片高

## 4.12 CFeatureGeometry 类

返回要素几何信息类

**示例:**

CFeatureGeometry viewport. GetFeatureGeometry (CGetFeatureInformation featureToGet),  
函数的返回值。

### 4.12.1 public CFeatureGeometry()

**说明:**

要素几何信息类构造函数

**参数:**

无

### 4.12.2 public void setPolygonNumber(int polygonNumber)

**说明:**

设置要素含有多边形个数

**参数:**

polygonNumber: 多边形个数

### 4.12.3 public int getPolygonNumber()

**说明:**

获取要素含有多边形个数

### 4.12.4 public void setArcNumber(int arcNumber)

**说明:**

设置含有弧线个数

**参数:**

arcNumber: 弧线个数

### 4.12.5 public int getArcNumber()

**说明:**

获取含有弧线的个数

### 4.12.6 public void setPointNumber(int pointNumber)

**说明:**

设置要素含有点个数

参数:

pointNumber: 要素含有点个数

#### 4.12.7 public int getPointNumber()

说明:

获取要素含有点个数

#### 4.12.8 public void setPolygons(CRegion2D[] polygons)

说明:

设置多边形对象数组

参数:

polygons: 多边形对象数组

#### 4.12.9 public CRegion2D[] getPolygons()

说明:

获取多边形对象数组

#### 4.12.10 public void setArcLines(CMultiLine[] arcLines)

说明:

设置复杂线对象数组

参数:

arcLines: 要素类中复杂线对象数组

#### 4.12.11 public CMultiLine[] getArcLines()

说明:

获取复杂线对象数组

#### 4.12.12 public void setPoints(CPoint2D[] points)

说明:

设置点对象数组

参数:

points: 点对象数组

#### 4.12.13 public CPoint2D[] getPoints()

说明:

获取点对象数组

### 4.13 CFeatureStyleInfo 类

返回要素几何信息类, 如颜色、高宽等。

示例:

CFeatureStyleInfo patter = viewport.getFeatureStyleInfo(CGetFeatureStyleInfo getStyle); 返回值为要素几何信息。

#### **4.13.1 public CFeatureStyleInfo()**

说明:

初始化要素几何信息类

参数:

无

#### **4.13.2 public float getPicAngle()**

说明:

获取要素的空间角度

#### **4.13.3 public void setPicAngle(float picAngle)**

说明:

设置要素的空间角度，包括点、线、区要素

参数:

picAngle: 要素的空间角度

#### **4.13.4 public int getPatternIndex()**

说明:

获取要素图案号

#### **4.13.5 public void setPatternIndex(int patternIndex)**

说明:

设置要素图案号

参数:

patternIndex: 图案号(点: 子图号; 线: 线型号; 区: 填充图案号)

#### **4.13.6 public float getMarginWidth()**

说明:

获取要素图案的宽

#### **4.13.7 public void setMarginWidth(float marginWidth)**

说明:

设置要素图案的宽

参数:

marginWidth: 要素图案的宽度

#### **4.13.8 public float getMarginHeight()**

说明:

获取要素图案的高度

#### 4.13.9 public void setMarginHeight(float marginHeight)

说明:

获取要素图案的高度

参数:

marginHeight: 要素图案的高度

#### 4.13.10 public int getColor()

说明:

获取要素图案的颜色

#### 4.13.11 public void setColor(int color)

说明:

设置要素图案的颜色

参数:

color: 要素图案的颜色(点: 点图颜色; 线: 线颜色; 区: 填充颜色)

#### 4.13.12 public int getFeatureType()

说明:

获取要素的类型

#### 4.13.13 public void setFeatureType(int featureType)

说明:

设置要素的类型

参数:

featureType: 要素的类型(包括点、线、区要素)

### 4.14 CGeomInfo 类

地理信息类。

示例:

CUniformResult uniformRlt = viewport.GetGeomInfo(CGeomInfo geomInfo); 函数入口函数。

#### 4.14.1 public CGeomInfo()

说明:

初始化地理信息类

参数:

无

#### 4.14.2 public CGeomInfo(long layerIndex, long mapID, short flg)

说明:

初始化地理信息类

**参数:**

layerIndex: 图层索引号

mapID: 地图文档 ID

flg: 标志位。0: 取图层类型; 1 删除对应图层

#### **4.14.3 public void setLayerIndex(long layerIndex)**

**说明:**

设置要素的图层索引号

**参数:**

layerIndex: 图层索引号

#### **4.14.4 public long getLayerIndex()**

**说明:**

获取要素的图层索引号

#### **4.14.5 public void setMapID(long mapID)**

**说明:**

设置要素的地图文档 ID

**参数:**

mapID: 地图文档 ID

#### **4.14.6 public long getMapID()**

**说明:**

获取要素的地图文档 ID

#### **4.14.7 public void setFLAG(short sFlg)**

**说明:**

设置标识位

**参数:**

sFlg: 标识位。0: 取图层类型; 1 删除对应图层

#### **4.14.8 public short getFLAG()**

**说明:**

获取标识位, 0: 取图层类型; 1 删除对应图层

### **4.15 CGetFeatureInformation 类**

获取要素信息类

**示例:**

CFeatureGeometry viewport.GetFeatureGeometry(CGetFeatureInformation featureToGet) 函数入口参数。

**4.15.1 public CGetFeatureInformation()**

说明:

初始化要素信息类

参数:

无

**4.15.2 public void setLayerIndex(long layerIndex)**

说明:

设置要获取要素所在图层

参数:

layerIndex: 图层索引号

**4.15.3 public long getLayerIndex()**

说明:

获取要素所在图层

**4.15.4 public void setFeatureID(long featureID)**

说明:

设置要获取要素的要素 ID

参数:

featureID: 要素的要素 ID

**4.15.5 public long getFeatureID()**

说明:

获取要素的要素 ID

**4.15.6 public void setInformationType(byte informationType)**

说明:

设置要获取要素的信息类型 0: 要素信息 1: 几何信息

参数:

informationType: 要素信息类型

**4.15.7 public byte getInformationType()**

说明:

获取要素的信息类型

**4.15.8 public void setMapID(long mapID)**

说明:

设置地图文档 ID

参数:

mapID: 地图文档 ID

#### 4.15.9 public long getMapID()

说明:

获取地图文档 ID

### 4.16 CGetFeatureStyleInfo 类

获取要素样式信息类，如颜色、高宽等

示例:

CFeatureStyleInfo patter = viewport.getFeatureStyleInfo(CGetFeatureStyleInfo getStyle); 函数入口参数。

#### 4.16.1 public CGetFeatureStyleInfo(long layerIndex, long mapID, long factorID)

说明:

初始要素样式信息类

参数:

layerIndex: 图层索引号

mapID: 地图文档号

factorID: 要素 ID

#### 4.16.2 public CGetFeatureStyleInfo()

说明:

初始要素样式信息类

参数:

无

#### 4.16.3 public long getLayerIndex()

说明:

获取要素所在图层

#### 4.16.4 public void setLayerIndex(long layerIndex)

说明:

设置要获取要素所在图层

参数:

layerIndex: 图层索引号

#### 4.16.5 public long getFeatureID()

说明:

获取要素的要素 ID

#### 4.16.6 public void setFeatureID(long featureID)

说明:

设置要获取要素的要素 ID

**参数:**

featureID: 要获取要素的要素 ID

#### **4.16.7 public long getMapID()**

**说明:**

获取地图文档 ID

#### **4.16.8 public void setMapID(long mapID)**

**说明:**

设置地图文档 ID

**参数:**

mapID: 地图文档 ID

### **4.17 CGetGDBInfo 类**

获取 GDB 信息类;

**示例:**

viewport. GetGDBInfo (CGetGDBInfo GDBInfo)函数入口参数。

#### **4.17.1 public CGetGDBInfo()**

**说明:**

初始化 GDB 信息类

**参数:**

无

#### **4.17.2 public void setMapID(long mapID)**

**说明:**

设置地图文档 ID

**参数:**

mapID: 地图文档 ID

#### **4.17.3 public long getMapID()**

**说明:**

获取地图文档 ID

### **4.18 CGetGDBLayerInfo 类**

GDB 信息类。

**示例:**

viewport. GetGDBLayerInfo (CGetGDBLayerInfo GDBLayerInfo)函数入口参数。

#### 4.18.1 public CGetGDBLayerInfo()

说明:

初始化 GDB 信息类

参数:

无

#### 4.18.2 public void setMapID(long mapID)

说明:

设置地图文档 ID

参数:

mapID: 地图文档 ID

#### 4.18.3 public long getMapID()

说明:

获取地图文档 ID

#### 4.18.4 public void setDataSource(java.lang.String dataSource)

说明:

设置 GDB 数据源名称, 如: MapGisLocal

参数:

dataSource: GDB 数据源名称

#### 4.18.5 public java.lang.String getDataSource()

说明:

获取 GDB 数据源名称

#### 4.18.6 public void setGDBName(java.lang.String gdbName)

说明:

设置 GDB 名称, 如 ChinaTest

参数:

gdbName: GDB 名称

#### 4.18.7 public java.lang.String getGDBName()

说明:

获取 GDB 名称

#### 4.18.8 public void setUsername(java.lang.String userName)

说明:

设置 GDB 数据库用户名

参数:

userName: GDB 数据库用户名

#### 4.18.9 public java.lang.String getUsername()

说明:

获取 GDB 数据库用户名

#### 4.18.10 public void setPwd(java.lang.String pwd)

说明:

设置 GDB 数据库密码

参数:

pwd: GDB 数据库密码

#### 4.18.11 public java.lang.String getPwd()

说明:

获取 GDB 数据库密码

### 4.19 CGetGDBName 类

获取 GDB 名称类。

示例:

viewport. GetGDBName (CGetGDBName GDBName)函数入口参数。

#### 4.19.1 public CGetGDBName()

说明:

构造函数

参数:

无

#### 4.19.2 public void setMapID(long mapID)

说明:

设置地图文档 ID

参数:

mapID: 地图文档 ID

#### 4.19.3 public long getMapID()

说明:

获取地图文档 ID

#### 4.19.4 public void setDataSource(java.lang.String dataSource)

说明:

设置 GDB 数据源名称, 如: MapGisLocal

参数:

dataSource: GDB 数据源名称

#### 4.19.5 public java.lang.String getDataSource()

说明:

获取 GDB 数据源名称

#### 4.19.6 public void setUsername(java.lang.String userName)

说明:

设置 GDB 数据库用户名

参数:

userName: GDB 数据库用户名

#### 4.19.7 public java.lang.String getUsername()

说明:

获取 GDB 数据库用户名

#### 4.19.8 public void setPwd(java.lang.String pwd)

说明:

设置 GDB 数据库密码

参数:

pwd: GDB 数据库密码

#### 4.19.9 public java.lang.String getPwd()

说明:

获取 GDB 数据库密码

### 4.20 CGetOnePage 类

查询结果显示的分页类

示例:

```
CGetOnePage getpage = new CGetOnePage(); //定义查询结果显示的分页类
getpage.setLineSelect(lineSelectObj); //设置线查询对象
getpage.setQueryType(CQueryType.SelectByLine); //设置查询类型
```

#### 4.20.1 public CGetOnePage()

说明:

初始化分页类

参数:

无

#### 4.20.2 public void setLayerIndex(long layerIndex)

说明:

设置要获取要素所在图层

参数:

layerIndex: 图层索引号

#### **4.20.3 public long getLayerIndex()**

说明:

获取要素所在图层

#### **4.20.4 public void setStartIndex(long startIndex)**

说明:

设置查询结果的开始页数

参数:

startIndex: 查询结果的开始页数

#### **4.20.5 public long getStartIndex()**

说明:

获取查询结果的开始页数

#### **4.20.6 public void setPntSelect(CPointSelectClass pntSelect)**

说明:

设置点击查询对象

参数:

pntSelect: 点击查询对象

#### **4.20.7 public CPointSelectClass getPntSelect()**

说明:

获取点击查询对象

#### **4.20.8 public void setRectSelect(CRectangleSelectClass rectSelect)**

说明:

设置拉框查询对象

参数:

rectSelect: 拉框查询对象

#### **4.20.9 public CRectangleSelectClass getRectSelect()**

说明:

获取拉框查询对象

#### **4.20.10 public void setPolygonSelect(CPolygonSelectClass polygonSelect)**

说明:

设置多边形查询对象

参数:

polygonSelect: 多边形查询对象

**4.20.11 public CPolygonSelectClass getPolygonSelect()**

说明:

获取多边形查询对象

**4.20.12 public void setCircleSelect(CCircleSelectClass circleSelect)**

说明:

设置圆查询对象

参数:

circleSelect: 圆查询对象

**4.20.13 public CCircleSelectClass getCircleSelect()**

说明:

获取圆查询对象

**4.20.14 public void setLineSelect(CLineSelectClass lineSelect)**

说明:

设置线查询对象

参数:

lineSelect: 线查询对象

**4.20.15 public CLineSelectClass getLineSelect()**

说明:

获取线查询对象

**4.20.16 public void setAttSelectStru(CAttSelectStru attSelect)**

说明:

设置条件查询对象

参数:

attSelect: 条件查询对象

**4.20.17 public CAttSelectStru getAttSelectStru()**

说明:

获取条件查询对象

**4.20.18 public void setRectWithAttSelect (CRectWithAttSelectStru rectWithAttSelect)**

说明:

设置拉框条件查询对象

参数:

rectWithAttSelect: 拉框条件查询对象

#### **4.20.19 public CRectWithAttSelectStru getRectWithAttSelect()**

说明:

获取拉框条件查询对象

#### **4.20.20 public void setPolygonWithAttSelect(CPolygonWithAttSelectStru polygonWithAttSelect)**

说明:

设置多边形条件查询对象

参数:

polygonWithAttSelect: 多边形条件查询对象

#### **4.20.21 public CPolygonWithAttSelectStru getPolygonWithAttSelect()**

说明:

获取多边形条件查询对象

#### **4.20.22 public void setCircleWithAttSelect(CCircleWithAttSelectStru circleWithAttSelect)**

说明:

设置圆条件查询对象

参数:

circleWithAttSelect: 圆条件查询对象

#### **4.20.23 public CCircleWithAttSelectStru getCircleWithAttSelect()**

说明:

获取圆条件查询对象

#### **4.20.24 public void setQueryType(java.lang.String selectType)**

说明:

设置查询类型

参数:

selectType: 查询类型, 可为任意查询类型 (包括: 点击、拉框、画圆、多边形、条件查询等)

#### **4.20.25 public java.lang.String getQueryType()**

说明:

获取查询类型

### **4.21 CLayerAttStructure 类**

获取要素属性结构类

示例:

```
CLayerAttStructure viewport.GetFeaturelayerAttributeStructure(long layerIndex,long MapID)  
函数返回值。
```

#### **4.21.1 public CLayerAttStructure()**

说明:

初始化要素属性结构类

参数:

无

#### **4.21.2 public void setFieldNumber(short fieldNumber)**

说明:

设置要素属性字段个数

参数:

fieldNumber: 要素属性字段个数

#### **4.21.3 public short getFieldNumber()**

说明:

获取要素属性字段个数

#### **4.21.4 public void setFieldNames(java.lang.String[] fieldNames)**

说明:

设置要素属性字段名集合

参数:

fieldNames: 要素属性字段名集合

#### **4.21.5 public java.lang.String[] getFieldNames()**

说明:

获取要素属性字段名集合

#### **4.21.6 public void setFieldTypes(short[] fieldTypes)**

说明:

设置要素属性字段类型集合

参数:

fieldTypes: 要素属性字段类型集合

#### **4.21.7 public short[] getFieldTypes()**

说明:

获取要素属性字段类型集合

#### **4.21.8 public void setFieldLength(short[] fieldLength)**

说明:

设置要素属性字段长度集合

**参数:**

fieldLength: 要素属性字段长度集合

#### **4.21.9 public short[] getFieldLength()**

**说明:**

获取要素属性字段长度集合

## **4.22 CLayerSelectResult 类**

获取指定图层的查询结果类。

**示例:**

是 CSelectResult 类的属性 LayerSelectResults 的返回值类型。

#### **4.22.1 public CLayerSelectResult()**

**说明:**

构造函数

**参数:**

无

#### **4.22.2 public void setLayerIndex(long layerIndex)**

**说明:**

设置要获取要素所在图层

**参数:**

layerIndex: 图层索引号

#### **4.22.3 public long getLayerIndex()**

**说明:**

获取要素所在图层

#### **4.22.4 public void setCount(short count)**

**说明:**

设置指定图层查询结果记录数

**参数:**

count: 查询结果记录数

#### **4.22.5 public short getCount()**

**说明:**

获取指定图层查询结果记录数

**4.22.6 public void setFieldNumber(short fieldNumber)**

说明:

设置指定图层字段个数

参数:

fieldNumber: 图层字段个数

**4.22.7 public short getFieldNumber()**

说明:

获取图层字段个数

**4.22.8 public void setFeatureIDs(long[] featureIDs)**

说明:

设置指定图层查询到的要素 ID 数组

参数:

featureIDs: 查询到的要素 ID 数组

**4.22.9 public long[] getFeatureIDs()**

说明:

获取指定图层查询到的要素 ID 数组

**4.22.10 public void setFieldNames(java.lang.String[] fieldNames)**

说明:

设置指定图层字段名数组

参数:

fieldNames: 字段名数组

**4.22.11 public java.lang.String[] getFieldNames()**

说明:

获取指定图层字段名数组

**4.22.12 public void setPropertyValues(java.util.ArrayList[] propertyValues)**

说明:

设置指定图层查询到的字段值数组集合

参数:

propertyValues: 查询到的字段值数组集合

**4.22.13 public java.util.ArrayList[] getPropertyValues()**

说明:

获取指定图层查询到的字段值数组集合

## 4.23 CLine 类

简单线类

### 4.23.1 public CLine()

说明:

简单线类构造函数

### 4.23.2 public CLine(CPoint2D[] points)

说明:

简单线类

参数:

points 点对象数组

### 4.23.3 public void setPointNumber(short pointNumber)

说明:

设置线上的点数

参数:

pointNumber: 线上的点数

### 4.23.4 public short getPointNumber()

说明:

得到线上的点

### 4.23.5 public void setThePoints(CPoint2D[] thePoints)

说明:

设置线上的点

参数:

thePoints: 点对象数组

### 4.23.6 public CPoint2D[] getThePoint()

说明:

得到点对象数组

## 4.24 CLineSelectClass 类

线查询类

示例:

short viewport.SelectByLine (CLineSelectClass selectLine)函数的入口参数。

**4.24.1 public CLineSelectClass(CLine line, double tolerance, long mapID)**

说明:

构造函数

参数:

line: 要查询的简单线

tolerance:

mapID: 地图文档 ID

**4.24.2 public CLineSelectClass()**

说明:

不含参数的构造函数

**4.24.3 public long getMapID()**

说明:

设置要查询线属性所在地图文档 ID

**4.24.4 public void setMapID(long mapID)**

说明:

设置要查询线属性所在地图文档 ID

参数:

mapID: 地图文档 ID

**4.24.5 public void setLayerIndex(long layerIndex)**

说明:

设置要添加图层的索引号

参数:

layerIndex: 所添加图层的索引号

**4.24.6 public long getLayerIndex()**

说明:

获取要添加图层的索引号

**4.24.7 public void setAttCondition(java.lang.String sqlQuery)**

说明:

设置线查询的条件语句

参数:

sqlQuery: 查询语句

**4.24.8 public java.lang.String getAttCondition()**

说明:

获得线查询的条件语句

#### **4.24.9 public void setPageIndex(long pageIndex)**

说明:

设置查询记录的起始索引

参数:

pageIndex: 起始索引号

#### **4.24.10 public long getPageIndex()**

说明:

获取查询记录的起始索引

#### **4.24.11 public double getLineRadius()**

说明:

获得线误差半径

#### **4.24.12 public void setLineRadius(double lineRadius)**

说明:

设置线查询误差半径

参数:

lineRadius: 误差半径

#### **4.24.13 public void setRevNum(int number)**

说明:

设置

参数:

number : 每次返回的最大查询结果条数

### **4.25 CMultiLine 类**

复杂线类

#### **4.25.1 public CMultiLine()**

说明:

构造函数

#### **4.25.2 public void setArcNumber(short arcNumber)**

说明:

设置弧线线的个数

参数:

arcNumber : 线的个数

### 4.25.3 public short getArcNumber()

说明:

获得弧线的个数

### 4.25.4 public void setTheArcs(CLine[] theArcs)

说明:

设置复杂线

参数:

theArcs: 复杂线数组

### 4.25.5 public CLine[] getTheArcs()

说明:

获得复杂线数组

## 4.26 CNetAnalysisResult 类

网络分析返回值类

示例:

CNetAnalysisResult viewport. NetAnalysis (CBasicNetAnalysis net)函数的返回值。

### 4.26.1 public CNetAnalysisResult()

说明:

构造函数

### 4.26.2 public void NetAnalysisResult()

说明:

设置

参数:

number : 每次返回的最大查询结果条数

### 4.26.3 public void setStart(CPoint2D start)

说明:

设置起始节点值

参数:

Start: 起始节点坐标

### 4.26.4 public CPoint2D getStart()

说明:

获得起始节点值

#### **4.26.5 public void setEnd(CPoint2D end)**

说明:

设置终节点值

参数:

end: 终节点坐标

#### **4.26.6 public CPoint2D getEnd()**

说明:

获得终节点值

#### **4.26.7 public void setInnerNodeNumber(short innerNodeNumber)**

说明:

设置网络结点的个数

参数:

innerNodeNumber: 网络结点的个数

#### **4.26.8 public short getInnerNodeNumber()**

说明:

获得网络结点的个数

#### **4.26.9 public void setInnerNodes(CPoint2D[] innerNodes)**

说明:

设置网络节点

参数:

innerNodes: 网络节点坐标

#### **4.26.10 public CPoint2D[] getInnerNodes()**

说明:

获得网络结点的值

#### **4.26.11 public void setPathNumber(short pathNumber)**

说明:

设置网络分析路径个数

参数:

pathNumber: 路径个数值

#### **4.26.12 public short getPathNumber()**

说明:

获得网络分析路径个数

#### 4.26.13 public void setPaths(CLine[] paths)

说明:

设置网络分析路径

参数:

Paths: 路径线数组

#### 4.26.14 public CLine[] getPaths()

说明:

获取网络分析路径的信息

#### 4.26.15 public CLayerSelectResult[] getLayerSelRlt()

说明:

获得网络分析结果信息

### 4.27 CPoint2D 类

二维点类

#### 4.27.1 public CPoint2D(double xPoint,double yPoint)

说明:

含参数的二维点构造函数

参数:

xPoint: 二维点 x 坐标值

yPoint: 二维点 y 坐标值

#### 4.27.2 public CPoint2D()

说明:

不含参数的二维点构造函数

#### 4.27.3 public void setPointX(double x)

说明:

设置二维点 x 坐标值

参数:

x: x 坐标值

#### 4.27.4 public double getPointX()

说明:

获取二维坐标点 x 的值

#### 4.27.5 public void setPointY(double y)

说明:

设置二维点 y 坐标值

**参数:**

y: y 坐标值

#### **4.27.6 public double getPointY()**

**说明:**

获取二维坐标点 y 的值

### **4.28 CPointSelectClass 类**

点击查询类

void viewport.SelectByHitting(CPointSelectClass pointSelect)函数的入口参数。

#### **4.28.1 public CPointSelectClass()**

**说明:**

不含参数的构造函数

#### **4.28.2 public CPointSelectClass(CPoint2D dot,double tolerance,long mapID)**

**说明:**

含参数的构造函数

**参数:**

dot: 点击位置

toleranc: 误差半径

mapID: 地图文档号

#### **4.28.3 public void setLayerIndex(long layerIndex)**

**说明:**

设置要查询图层的索引号

**参数:**

layerIndex: 图层索引号

#### **4.28.4 public long getLayerIndex()**

**说明:**

获取要查询图层的索引号

#### **4.28.5 public void setDot(CPoint2D dot)**

**说明:**

设置要查询点的信息

**参数:**

dot: 二维点坐标

#### **4.28.6 public CPoint2D getDot()**

说明:

获取要查询的二维点的信息

#### **4.28.7 public void setDistance(double distance)**

说明:

设置查询的误差半径

参数:

distance: 误差半径值

#### **4.28.8 public double getDistance()**

说明:

获取误差半径值

#### **4.28.9 public void setMapID(long mapID)**

说明:

设置要查询点属性所在地图文档 ID

参数:

mapID: 地图文档 ID

#### **4.28.10 public long getMapID()**

说明:

获取要查询点属性所在地图文档 ID

#### **4.28.11 public void setPageIndex(long pageIndex)**

说明:

设置查询结果当前页的第一条记录的索引

参数:

pageIndex: 起始索引号

#### **4.28.12 public long getPageIndex()**

说明:

获取查询结果当前页的第一条记录的索引

#### **4.28.14 public void setRevNum(int number)**

说明:

设置点击查询每次返回的最大查询结果条数

参数:

number: 每次返回的最大查询结果条数

#### 4.28.15 public int getRevOnce()

说明:

获取点击查询每次返回的最大查询结果条数

### 4.29 CPolygonSelectClass 类

多边形查询类

void viewport.SelectByPolygon(CPolygonSelectClass polygonSelect)函数的入口参数。

#### 4.29.1 public CPolygonSelectClass(CPoint2D[] points,short[] nums,long mapID)

说明:

含参数的构造函数

参数:

points: 所有多边形有序包含的所有顶点

nums: 各个多边形顶点个数

mapID: 地图文档号

#### 4.29.2 public CPolygonSelectClass()

说明:

不含参数的构造函数

#### 4.29.3 public short[] getPointNum()

说明:

获取各个多边形上的点的个数

#### 4.29.4 public void setPointNum(short[] value)

说明:

设置各个多边形上的点数数组

参数:

value: 点数数组值

#### 4.29.5 public void setPolygonNum(short value)

说明:

设置多边形的个数

参数:

Value: 多边形个数值

#### 4.29.6 public long getPolygonNum()

说明: 获取多边形个数值

**4.29.7 public void setLayerIndex(long layerIndex)**

说明:

设置要查询多边形图层的索引号

参数:

layerIndex: 所查询图层的索引号

**4.29.8 public long getLayerIndex()**

说明:

获取要查询多边形图层的索引号

**4.29.9 public void setPointNumber(short pointNumber)**

说明:

设置多边形上某一点的值

参数:

pointNumber: 所要设置点的点号

**4.29.10 public short getPointNumber()**

说明:

获取多边形上的点数值

**4.29.11 public void setPolygon(CPoint2D[] polygon)**

说明:

设置多边形上的坐标点对象数组

参数:

polygon: 坐标点对象数组值

**4.29.12 public CPoint2D[] getPolygon()**

说明:

获取多边形上的坐标点对象数组值

**4.29.13 public void setMapID(long mapID)**

说明:

设置要查询多边形所在地图文档 ID

参数:

mapID: 地图文档 ID

**4.29.14 public long getMapID()**

说明:

获取所要查询多边形所在地图文档 ID

#### 4.29.15 public void setPageIndex(long pageIndex)

说明:

设置查询记录的起始索引

参数:

pageIndex: 起始索引号

#### 4.29.16 public long getPageIndex()

说明:

获取查询记录的起始索引

#### 4.29.17 public void setRevNum(int number)

说明:

设置多边形查询每次返回的最大查询结果条数

参数:

number: 每次返回的最大查询结果条数

#### 4.29.18 public int getRevOnce()

说明:

获取每次返回的最大查询结果条数

### 4.30 CPolygonWithAttSelectStru 类

多边形条件查询类

short viewport.SelectByCondition (CPolygonWithAttSelectStru polygonAndAtt);函数的入口参数。

#### 4.30.1 Public CPolygonWithAttSelectStru(CPoint2D[] polygon, short[] pointNum,long mapID)

说明:

带参数的构造函数

参数:

polygon: 所有多边形有序包含的所有顶点

nums: 各个多边形顶点个数

mapID: 地图文档号

#### 4.30.2 public CPolygonWithAttSelectStru()

说明:

不带参数的构造函数

#### 4.30.3 public short[] getPointNum()

说明:

获取各个多边形上的点的个数

#### **4.30.4 public void setPointNum(short[] value)**

**说明:**

设置各个多边形上的点数数组

**参数:**

value: 点数数组值

#### **4.30.5 public void setPolygonNum(short value)**

**说明:**

设置多边形的个数

**参数:**

value: 多边形个数值

#### **4.30.6 public long getPolygonNum()**

**说明:** 获取多边形个数值

#### **4.30.7 public void setLayerIndex(long layerIndex)**

**说明:**

设置要查询多边形图层的索引号

**参数:**

layerIndex: 所查询图层的索引号

#### **4.30.8 public long getLayerIndex()**

**说明:**

获取要查询多边形图层的索引号

#### **4.30.9 public void setPointNumber(short pointNumber)**

**说明:**

设置多边形上的点数

**参数:**

pointNumber: 多边形上的点数值

#### **4.30.10 public short getPointNumber()**

**说明:**

获取多边形上的点的数值

#### **4.30.11 public void setPolygon(CPoint2D[] polygon)**

**说明:**

设置多边形上的坐标点对象数组

**参数:**

polygon: 坐标点对象数组值

**4.30.12 public CPoint2D[] getPolygon()**

说明:

获取多边形上的坐标点对象数组值

**4.30.13 public void setMapID(long mapID)**

说明:

设置要查询多边形所在地图文档 ID

参数:

mapID: 地图文档 ID

**4.30.14 public long getMapID()**

说明:

获取所要查询多边形所在地图文档 ID

**4.30.15 public void setPageIndex(long pageIndex)**

说明:

设置查询记录的起始索引

参数:

pageIndex: 起始索引号

**4.30.16 public long getPageIndex()**

说明:

获取查询记录的起始索引

**4.30.17 public void setAttCondition(java.lang.String sqlQuery)**

说明:

设置查询条件

参数:

sqlQuery: sql 查询语句

**4.30.18 public java.lang.String getAttCondition()**

说明:

获取所设置的查询条件

**4.30.19 public void setRevNum(int number)**

说明:

设置多边形条件查询每次返回的最大查询结果条数

参数:

number: 每次返回的最大查询结果条数

#### **4.30.20 public int getRevOnce()**

说明:

获取查询每次返回的最大查询结果条数

### **4.31 CRect 类**

简单矩形类，逻辑/窗口坐标均可。

#### **4.31.1 public CRect(CPoint2D min,CPoint2D max)**

说明:

带参数的构造函数

参数:

min: 矩形左上角点

max: 矩形右下角点

#### **4.31.1 public CRect(double xMin,double yMin,double xMax,double yMax)**

说明:

带参数的构造函数

参数:

xMin: 矩形上的最小 x 值

yMin: 矩形上的最小 y 值

xMax: 矩形上的最大 x 值

yMax: 矩形上的最大 y 值

#### **4.31.2 public void setXMin(double xMin)**

说明:

设置矩形的最小 x 值

参数:

xMin: x 的坐标值

#### **4.31.3 public double getXMin()**

说明:

获取矩形上的最小 x 值

#### **4.31.4 public void setYMin(double yMin)**

说明:

设置矩形的最小 y 值

参数:

yMin: y 的坐标值

#### 4.31.5 public double getYMin()

说明:

获取矩形上的最小 y 值

#### 4.31.6 public void setXMax(double xMax)

说明:

设置矩形的最大 x 值

参数:

xMax: x 的坐标值

#### 4.31.7 public double getXMax()

说明:

获取矩形上的最大 x 值

#### 4.31.8 public void setYMax(double yMax)

说明:

设置矩形的最大 y 值

参数:

yMax: y 的坐标值

#### 4.31.9 public double getYMax()

说明:

获取矩形上的最大 y 值

### 4.32 CRectangleSelectClass 类

拉框查询类

示例:

void viewport.SelectByRect(CRectangleSelectClass rectangleSelect);函数的入口参数。

#### 4.32.1 public CRectangleSelectClass()

说明:

不带参数的构造函数

#### 4.32.2 public CRectangleSelectClass(CRect rectangleArea,long mapID)

说明:

带参数的构造函数

参数:

rectangleArea: 矩形查询框

mapID: 地图文档号

#### 4.32.3 public void setLayerIndex(long layerIndex)

说明:

设置要查询图层的索引号

参数:

layerIndex: 图层索引号

#### 4.32.4 public long getLayerIndex()

说明:

获取要查询图层的索引号

#### 4.32.5 public void setRectangleArea(CRect rectangleArea)

说明:

设置矩形区信息

参数:

rectangleArea: 矩形对象

#### 4.32.6 public CRect getRectangleArea()

说明:

获得矩形区的的信息

#### 4.32.7 public void setMapID(long mapID)

说明:

设置要查询对象所在地图文档 ID

参数:

mapID: 地图文档 ID

#### 4.32.8 public long getMapID()

说明:

获取所要查询对象所在地图文档 ID

#### 4.32.9 public void setPageIndex(long pageIndex)

说明:

设置查询记录的起始索引

参数:

pageIndex: 起始索引号

#### 4.32.10 public long getPageIndex()

说明:

获取查询记录的起始索引

#### 4.32.11 public void setRevNum(int number)

说明:

设置矩形查询每次返回的最大查询结果条数

参数:

number: 每次返回的最大查询结果条数

#### 4.32.12 public int getRevOnce()

说明:

获取查询每次返回的最大查询结果条数

### 4.33 CRectWithAttSelectStru 类

矩形条件查询类

示例:

short viewport.SelectByCondition (CRectWithAttSelectStru rectAndAtt);函数的入口参数。

#### 4.33.1 public CRectWithAttSelectStru(CRect bound,long mapID)

说明:

带参数的构造函数

参数:

bound: 拉框查询范围

mapID: 地图文档号

#### 4.33.2 public CRectWithAttSelectStru()

说明:

不带参数的构造函数

#### 4.33.3 public void setLayerIndex(long layerIndex)

说明:

设置要查询图层的索引号

参数:

layerIndex: 图层索引号

#### 4.33.4 public long getLayerIndex()

说明:

获取要查询图层的索引号

#### 4.33.5 public void setLayerBound(CRect layerBound)

说明:

设置指定图层逻辑范围

参数:

layerBound: 逻辑矩形范围

#### **4.33.6 public CRect getLayerBound()**

说明:

获取指定图层逻辑范围

#### **4.33.7 public void setMapID(long mapID)**

说明:

设置要查询对象所在地图文档 ID

参数:

mapID: 地图文档 ID

#### **4.33.8 public long getMapID()**

说明:

获取所要查询对象所在地图文档 ID

#### **4.33.9 public void setAttCondition(java.lang.String sqlQuery)**

说明:

设置所要查询的条件

参数:

sqlQuery: sql 查询语句

#### **4.33.10 public java.lang.String getAttCondition()**

说明:

获取设置的查询条件

#### **4.33.11 public void setPageIndex(long pageIndex)**

说明:

设置查询记录的起始索引

参数:

pageIndex: 起始索引号

#### **4.33.12 public long getPageIndex()**

说明:

获取查询记录的起始索引

#### **4.33.13 public void setRevNum(int number)**

说明:

设置矩形条件查询每次返回的最大查询结果条数

参数:

number: 每次返回的最大查询结果条数

#### 4.33.14 public int getRevOnce()

说明:

获取查询每次返回的最大查询结果条数

### 4.34 CRegion2D 类

二维区域类，可获取指定区上弧段。

#### 4.34.1 public CRegion2D()

说明:

不带参数的构造函数

#### 4.34.2 public void setArcNumber(short arcNumber)

说明:

设置要素弧线个数

参数:

arcNumber: 弧线个数值

#### 4.34.3 public short getArcNumber()

说明:

获取要素的弧线个数

#### 4.34.4 public void setTheArcs(CLine[] theArcs)

说明:

设置线对象属性

参数:

theArcs: 线对象

#### 4.34.5 public CLine[] getTheArcs()

说明:

获取线对象属性

### 4.35 CSelectResult 类

查询结果类

示例:

```
CSelectResult selectResultObj = viewport.getLastSelectResult();
```

#### 4.35.1 public CSelectResult()

说明:

不带参数的构造函数

#### **4.35.2 public long[] getElementIDs(long layerIndex)**

说明:

获得要素类的 ID 号

参数:

layerIndex: 图层索引号

#### **4.35.3 public void setTotalFCount(long totalCount)**

说明:

设置查询到的所有要素的个数

参数:

totalCount: 要素个数值

#### **4.35.4 public long getTotalFCount()**

说明:

获取查询得到的所有要素的个数

#### **4.35.5 public void setLayerCount(short layerCount)**

说明:

设置子图层个数

参数:

layerCount: 子图层个数值

#### **4.35.6 public short getLayerCount()**

说明:

获取子图层个数值

#### **4.35.7 public void setlayerSelInfo(CLayerSelectResult[] layerSelectResults)**

说明:

设置查询结果图层的的信息

参数:

layerSelectResults: 查询结果图层对象

#### **4.35.8 public CLayerSelectResult[] getlayerSelInfo()**

说明:

获取结果图层的的信息

### **4.36 CSpatialBuffer 类**

缓冲区分析类

示例:

CSpatialOperResult viewport.SpatialAnalysisBuffer(CSpatialBuffer buffer); 函数的入口参数。

#### **4.36.1 public CSpatialBuffer(long layerIndex,long[] featureIDS, long mapID, CRect curBound)**

说明:

带参数的构造函数

参数:

LayerIndex: 图层索引号

FeatureIDS: 要素 ID 号

MapID: 地图文档 ID

CurBound: 当前地图的显示范围

#### **4.36.2 public CSpatialBuffer()**

说明:

不带参数的构造函数

#### **4.36.3 public void setLayerIndex(long layerIndex)**

说明:

设置要查询图层的索引号

参数:

layerIndex: 图层索引号

#### **4.36.4 public long getLayerIndex()**

说明:

获取要查询图层的索引号

#### **4.36.5 public void setFeatureNumber(short featureNumber)**

说明:

设置缓冲区分析的要素个数

参数:

featureNumber: 要素个数值

#### **4.36.6 public short getFeatureNumber()**

说明:

获取缓冲区要素的个数值

#### **4.36.7 public void setFeatureIDS(long[] featureIDS)**

说明:

设置要素 ID 号

参数:

featureIDS: 要素 ID 号

#### **4.36.8 public long[] getFeatureIDS()**

说明:

获取要素 ID 号

#### **4.36.9 public void setRadius(double radius)**

说明:

设置缓冲区半径

参数:

Radius: 缓冲区半径值

#### **4.36.10 public double getRadius()**

说明:

获取缓冲区半径值

#### **4.36.11 public void setFeatureClassName(java.lang.String featureClassName)**

说明:

设置缓冲区分析文件名称

参数:

featureClassName: 缓冲区的文件名称

#### **4.36.12 public java.lang.String getFeatureClassName()**

说明:

获取缓冲区分析文件名称

#### **4.36.13 public void setMapID(long mapID)**

说明:

设置缓冲区对象所在地图文档 ID

参数:

mapID: 地图文档 ID

#### **4.36.14 public long getMapID()**

说明:

获取缓冲区对象所在地图文档 ID

#### **4.36.15 public void setMapWidth(int width)**

说明:

设置缓冲区图像宽度

参数:

Width: 宽度值

#### 4.36.16 public void setMapHeight(int height)

说明:

设置缓冲区图像高度

参数:

Height: 高度值

#### 4.36.17 public void setCurrentBound(CRect displayRectangle)

说明:

设置当前对象的显示范围

参数:

displayRectangle: 地图的显示范围

#### 4.36.18 public int getMapWidth()

说明:

获取缓冲区图像宽度值

#### 4.36.19 public int getMapHeight()

说明:

获取缓冲区图像高度值

#### 4.36.20 public CRect getCurrentBound()

说明:

获取当前对象的显示范围

### 4.37 CSpatialCircleClip 类

圆裁剪类

示例:

CSpatialOperResult viewport.SpatialAnalysisCircleClip(CSpatialCircleClip circleClip); 函数的入口参数。

#### 4.37.1 public CSpatialCircleClip()

说明:

不带参数的构造函数

#### 4.37.2 public int getCurMapWidth()

说明:

获取当前地图的宽度值

#### 4.37.3 public void setCurMapWidth(int curMapWidth)

说明:

设置当前地图的宽度

**参数:**

curMapWidth: 地图宽度值

#### **4.37.4 public int getCurMapHeight()**

**说明:**

获取当前地图的高度值

#### **4.37.5 public void setCurMapHeight(int curMapHeight)**

**说明:**

设置当前地图的高度值

**参数:**

curMapHeight: 地图高度值

#### **4.37.6 public CRect getCurBound()**

**说明:**

获取当前对象的显示范围

#### **4.37.7 public void setCurBound(CRect curBound)**

**说明:**

设置当前对象的显示范围

**参数:**

curBound: 地图的显示范围

#### **4.37.8 public void setLayerIndex(long layerIndex)**

**说明:**

设置要查询图层的索引号

**参数:**

layerIndex: 图层索引号

#### **4.37.9 public long getLayerIndex()**

**说明:**

获取要查询图层的索引号

#### **4.37.10 public void setCenter(CPoint2D center)**

**说明:**

设置中心值

**参数:**

Center: 中心点对象

#### **4.37.11 public CPoint2D getCenter()**

**说明:**

获取中心值

#### **4.37.12 public void setRadius(double radius)**

说明:

设置剪裁半径

参数:

Radius: 剪裁半径值

#### **4.37.13 public double getRadius()**

说明:

获取剪裁半径值

#### **4.37.14 public void setClipType(short clipType)**

说明:

设置圆剪裁类型

参数:

clipType: 圆剪裁类型值

#### **4.37.15 public short getClipType()**

说明:

获取圆剪裁类型值

#### **4.37.16 public void setFeatureClassName(java.lang.String featureClassName)**

说明:

设置裁减后生成的文件名

参数:

featureClassName: 圆剪裁类型名

#### **4.37.17 public java.lang.String getFeatureClassName()**

说明:

获取裁减后生成的文件名

#### **4.37.18 public void setMapID(long mapID)**

说明:

设置圆剪裁对象所在地图文档 ID

参数:

mapID: 地图文档 ID

#### **4.37.19 public long getMapID()**

说明:

获取圆剪裁对象所在地图文档 ID

#### 4.37.20 public void setLayerIndexLst(java.lang.String layerIndexLst)

说明:

设置圆裁减对象的图层索引号

参数:

layerIndexLst: 图层索引号

#### 4.37.21 public java.lang.String getLayerIndexLst()

说明:

获取圆裁减对象的图层索引号

#### 4.37.22 public void setDisRadius(double disRadius)

说明:

设置裁剪容差半径

参数:

disRadius: 圆裁减容差半径值

#### 4.37.23 public double getDisRadius()

说明:

获取裁剪容差半径

### 4.38 CSpatialOperResult 类

空间分析返回结果类

#### 4.38.1 public CSpatialOperResult()

说明:

不带参数的构造函数

#### 4.38.2 public java.lang.String getRltHDFName()

说明:

获取空间分析结果 HDF 名称

#### 4.38.3 public void setRltHDFName(java.lang.String rltHDFName)

说明:

设置空间分析结果 HDF 名称

参数:

rltHDFName: 空间分析结果 HDF 名称

#### 4.38.4 public java.lang.String getRltFeattrueName()

说明:

获取空间分析结果要素类名称

#### 4.38.5 public void setRltFeattrueName(java.lang.String rltFeattrueName)

说明:

设置空间分析要素类名称

参数:

rltFeattrueName: 要素类名称

#### 4.38.6 public void setZipFileName(java.lang.String zipFileName)

说明:

设置压缩文件名称

参数:

zipFileName: 压缩文件名称

#### 4.38.7 public java.lang.String getZipFileName()

说明:

获取压缩文件名称

#### 4.38.8 public void setFileSize(int nFileSize)

说明:

设置文件大小

参数:

nFileSize: 文件大小值

#### 4.38.9 public int getFileSize()

说明:

获取文件大小值

### 4.39 CSpatialOverlay 类

叠加分析类

示例:

```
CSpatialOperResult SpatialOperRlt = viewport.SpatialAnalysisOverlay(CSpatialOverlay  
overlay);
```

#### 4.39.1 public CSpatialOverlay()

说明:

不带参数的构造函数

#### 4.39.2 public int getCurMapWidth()

说明:

获取当前地图的宽度值

**4.39.3 public void setCurMapWidth(int curMapWidth)**

说明:

设置当前地图的宽度

参数:

curMapWidth: 地图宽度值

**4.39.4 public int getCurMapHeight()**

说明:

获取当前地图的高度值

**4.39.5 public void setCurMapHeight(int curMapHeight)**

说明:

设置当前地图的高度值

参数:

curMapHeight: 地图高度值

**4.39.6 public CRect getCurBound()**

说明:

获取当前对象的显示范围

**4.39.7 public void setCurBound(CRect curBound)**

说明:

设置当前对象的显示范围

参数:

curBound: 地图的显示范围

**4.39.8 public void setLayerIndex1(long layerIndex1)**

说明:

设置图层 1 序号

参数:

layerIndex1: 图层 1 序列号

**4.39.9 public long getLayerIndex1()**

说明:

获取图层 1 序号

**4.39.10 public void setLayerIndex2(long layerIndex2)**

说明:

设置图层 2 序号

参数:

layerIndex2: 图层 2 序列号

#### **4.39.11 public long getLayerIndex2()**

说明:

获取图层 2 序号

#### **4.39.12 public void setOverlayType(short overlayType)**

说明:

设置叠加类型

参数:

overlayType: 叠加类型值

#### **4.39.13 public void setFeatureClassName(java.lang.String featureClassName)**

说明:

设置叠加后生成的文件名称

参数:

featureClassName: 文件名称

#### **4.39.14 public java.lang.String getFeatureClassName()**

说明:

获取叠加后生成的文件名称

#### **4.39.15 public void setMapID(long mapID)**

说明:

设置叠加后生成对象所在地图文档 ID

参数:

mapID: 地图文档 ID

#### **4.39.16 public long getMapID()**

说明:

获取叠加后生成对象所在地图文档 ID

#### **4.39.17 public void setGDBDataSourceName1(java.lang.String gdbDataSourceName1)**

说明:

设置叠加所需图层 1 数据源名称

参数:

gdbDataSourceName1: 数据源 1 名称

#### **4.39.18 public java.lang.String getGDBDataSourceName1()**

说明:

获取叠加所需图层 1 数据源名称

#### **4.39.19 public void setGDBDataSourceName2(java.lang.String gdbDataSourceName2)**

**说明:**

设置叠加所需图层 2 数据源名称

**参数:**

gdbDataSourceName2: 数据源 2 名称

#### **4.39.20 public java.lang.String getGDBDataSourceName2()**

**说明:**

设置叠加所需图层 2 数据源名称

#### **4.39.21 public void setGDBName1(java.lang.String gdbName1)**

**说明:**

设置叠加所需图层 1GDB 名

**参数:**

gdbName1: 图层 1 的 GDB 名

#### **4.39.22 public java.lang.String getGDBName1()**

**说明:**

获取叠加所需图层 1GDB 名

#### **4.39.23 public void setGDBName2(java.lang.String gdbName2)**

**说明:**

设置叠加所需图层 2GDB 名

**参数:**

gdbName2: 图层 2 的 GDB 名

#### **4.39.24 public java.lang.String getGDBName2()**

**说明:**

获取叠加所需图层 2GDB 名

#### **4.39.25 public void setFeatureIDSLst1(java.lang.String featureIDSLst1)**

**说明:**

设置第一个数据库的图层名称

**参数:**

featureIDSLst1: 数据库图层 1 名称

#### **4.39.26 public java.lang.String getFeatureIDSLst1()**

**说明:**

获取第一个数据库的图层名称

#### **4.39.27 public void setFeatureIDSLst2(java.lang.String featureIDSLst2)**

说明:

设置第二个数据库的图层名称

参数:

featureIDSLst2: 数据库图层 2 名称

#### **4.39.28 public java.lang.String getFeatureIDSLst2()**

说明:

获取第二个数据库的图层名称

#### **4.39.29 public void setUsername1(java.lang.String userName)**

说明:

设置网络数据用户 1 名字

参数:

userName: 用户名

#### **4.39.30 public java.lang.String getUsername1()**

说明:

获取网络数据用户 1 名字

#### **4.39.31 public void setPwd1(java.lang.String pwd)**

说明:

网络用户 1 数据密码

参数:

pwd: 密码一

#### **4.39.32 public java.lang.String getPwd1()**

说明:

获取网络用户 2 数据密码

#### **4.39.33 public void setUsername2(java.lang.String userName)**

说明:

设置网络数据用户 1 名字

参数:

userName: 用户名

#### **4.39.34 public java.lang.String getUsername2()**

说明:

获取网络数据用户 2 名字

#### 4.39.35 public void setPwd2(java.lang.String pwd)

说明:

网络用户 2 数据密码

参数:

pwd: 密码二

#### 4.39.36 public java.lang.String getPwd2()

说明:

获取网络用户 2 数据密码

#### 4.39.37 public void setRadius(double radius)

说明:

设置叠加分析误差半径

参数:

radius: 误差半径值

#### 4.39.38 public double getRadius()

说明:

获取叠加分析容差半径

### 4.40 CSpatialPolygonClip 类

多边形裁剪类

示例:

```
CSpatialOperResult SpatialOperRlt = viewport.SpatialAnalysisPolygonClip  
(CSpatialPolygonClip polygonClip);
```

#### 4.40.1 public CSpatialPolygonClip()

说明:

不带参数的构造函数

#### 4.40.2 public int getCurMapWidth()

说明:

获取当前地图的宽度值

#### 4.40.3 public void setCurMapWidth(int curMapWidth)

说明:

设置当前地图的宽度

参数:

curMapWidth: 地图宽度值

**4.40.4 public int getCurMapHeight()**

说明:

获取当前地图的高度值

**4.40.5 public void setCurMapHeight(int curMapHeight)**

说明:

设置当前地图的高度值

参数:

curMapHeight: 地图高度值

**4.40.6 public CRect getCurBound()**

说明:

获取当前对象的显示范围

**4.40.7 public void setCurBound(CRect curBound)**

说明:

设置当前对象的显示范围

参数:

curBound: 地图的显示范围

**4.40.8 public int[] getPointNum()**

说明:

获取各个多边形点的个数

**4.40.9 public void setPointNum(int[] value)**

说明:

设置各个多边形点的个数

参数:

Value: 多边形的点的个数

**4.40.10 public void setPolygonNum(int value)**

说明:

设置多边形个数

参数:

value: 多边形的个数

**4.40.11 public int getPolygonNum()**

说明:

获取多边形个数

#### 4.40.12 public void setLayerIndex(long layerIndex)

说明:

设置要裁减图层的索引号

参数:

layerIndex: 图层索引号

#### 4.40.13 public long getLayerIndex()

说明:

获取要裁减图层的索引号

#### 4.40.14 public void setPointNumber(int pointNumber)

说明:

设置多边形上的各点的值

参数:

pointNumber: 所要设置的点号

#### 4.40.15 public int getPointNumber()

说明:

获取多边形上某一点的值

#### 4.40.16 public void setPoints(CPoint2D[] points)

说明:

设置坐标点序列

参数:

points: 坐标点序列数组

#### 4.40.27 public CPoint2D[] getPoints()

说明:

获取坐标点序列

#### 4.40.18 public void setClipType(int clipType)

说明:

设置多边形裁减类型

参数:

clipType: 多边形剪裁类型值

#### 4.40.19 public short getClipType()

说明:

获取多边形剪裁类型值

**4.40.20 public void setFeatureClassName(java.lang.String featureClassName)**

说明:

设置裁减后生成的文件名称

参数:

featureClassName: 文件名称

**4.40.21 public java.lang.String getFeatureClassName()**

说明:

获取裁减后生成的文件名称

**4.40.22 public void setMapID(long mapID)**

说明:

设置裁减后生成对象所在地图文档 ID

参数:

mapID: 地图文档 ID

**4.40.23 public long getMapID()**

说明:

获取裁减后生成对象所在地图文档 ID

**4.40.24 public void setLayerIndexLst(java.lang.String layerIndexLst)**

说明:

设置多边形裁减对象的图层索引号

参数:

layerIndexLst: 图层索引号

**4.40.25 public java.lang.String getLayerIndexLst()**

说明:

获取多边形裁减对象的图层索引号

**4.40.26 public void setDisRadius(double disRadius)**

说明:

设置裁剪容差半径

参数:

disRadius: 多边形裁减容差半径值

**4.40.27 public double getDisRadius()**

说明:

获取裁剪容差半径

## 4.41 CUniformResult 类

添加、删除、更新要素统一返回值类

示例:

CUniformResult uniformRlt = viewport.GetGDBInfo(GDBInfo);函数的返回值。

### 4.41.1 public CUniformResult()

说明:

不带参数的构造函数

### 4.41.2 public void setOperCode(short operCode)

说明:

设置操作代码

参数:

operCode: 操作代码值

### 4.41.3 public short getOperCode()

说明:

获取操作代码值

### 4.41.4 public void setFlag(short flag)

说明:

设置返回标志值

参数:

Flag: 返回标志

### 4.41.5 public short getFlag()

说明:

获取返回标志

### 4.41.6 public void setMessage(java.lang.String message)

说明:

设置信息值

参数:

Message: 信息值

### 4.41.7 public java.lang.String getMessage()

说明:

获取信息值

## 4.42 CUpdateFeature 类

更新点、线、区要素类

示例:

```
CUniformResult updateSpatialRlt = viewport.UpdateSpatialFeature(CUpdateFeature  
newSpatialInfo);函数入口参数。
```

### 4.42.1 public CUpdateFeature()

说明:

不带参数的构造函数

### 4.42.2 public void setLayerIndex(long layerIndex)

说明:

设置更新后的图层索引号

参数:

layerIndexLst: 图层索引号

### 4.42.3 public long getLayerIndex()

说明:

获取更新后的图层索引号

### 4.42.4 public void setFeatureID(long featureID)

说明:

设置要素 ID 值

参数:

featureID: 要素 ID 号

### 4.42.5 public long getFeatureID()

说明:

获取要素 ID 号

### 4.42.6 public void setFeatureGeometry(CFeatureGeometry fGeometry)

说明:

设置要素几何信息

参数:

fGeometry: 要素几何信息

### 4.42.7 public CFeatureGeometry getFeatureGeometry()

说明:

获取要素的几何信息

#### 4.42.8 public void setMapID(long mapID)

说明:

设置更新后生成对象所在地图文档 ID

参数:

mapID: 地图文档 ID

#### 4.42.9 public long getMapID()

说明:

获取更新后生成对象所在地图文档 ID

### 4.43 CUpdateFeatureAttribute 类

更新点、线、区要素类

示例:

CUniformResult updateRlt = viewport.UpdateFeature(CUpdateFeatureAttribute newAttribute);函数的入口参数。

#### 4.43.1 public CUpdateFeatureAttribute()

说明:

不带参数的构造函数

#### 4.43.2 public void setLayerIndex(long layerIndex)

说明:

设置更新后的图层索引号

参数:

layerIndexLst: 图层索引号

#### 4.43.3 public long getLayerIndex()

说明:

获取更新后的图层索引号

#### 4.43.4 public void setFeatureID(long featureID)

说明:

设置要素 ID 值

参数:

featureID: 要素 ID 号

#### 4.43.5 public long getFeatureID()

说明:

获取要素 ID 号

#### 4.43.6 public void setFieldNumber(short fieldNumber)

说明:

设置字段个数

参数:

fieldNumber: 字段个数

#### 4.43.7 public short getFieldNumber()

说明:

获取字段个数

#### 4.43.8 public void setNewFieldValues(java.lang.String[] newFieldValues)

说明:

设置新字段值

参数:

newFieldValues: 新字段值

#### 4.43.9 public java.lang.String[] getNewFieldValues()

说明:

获取新字段值

#### 4.43.10 public void setMapID(long mapID)

说明:

设置更新后生成对象所在地图文档 ID

参数:

mapID: 地图文档 ID

#### 4.43.11 public long getMapID()

说明:

获取更新后生成对象所在地图文档 ID

### 4.44 CUpdateFeatureStyleInfo 类

更新要素样式属性类

示例:

```
CUniformResult updateFeatureStyleInfo =  
viewport.updateFeatureStyleInfo(CUpdateFeatureStyleInfo updatePatter);函数的入口参数。
```

#### 4.44.1 public CUpdateFeatureStyleInfo(CFeatureStyleInfo info, long layerIndex, long mapID, long factorID)

说明:

带参数的构造函数

**参数:**

info: 要素样式信息对象

layerIndex: 图层索引

mapID: 地图文档号

factorIDinfo: 对应的要素 id

#### **4.44.2 public CUpdateFeatureStyleInfo()**

**说明:**

不带参数的构造函数

#### **4.44.3 public float getPicAngle()**

**说明:**

获取子图的角度

#### **4.44.4 public void setPicAngle(float picAngle)**

**说明:**

设置子图的角度

**参数:**

picAngle: 子图的角度值

#### **4.44.5 public long getMapID()**

**说明:**

获取更新后生成对象所在地图文档 ID

#### **4.44.6 public void setMapID(long mapID)**

**说明:**

设置更新后生成对象所在地图文档 ID

**参数:**

mapID: 地图文档 ID

#### **4.44.7 public long getLayerIndex()**

**说明:**

获取更新后的图层索引号

#### **4.44.8 public void setLayerIndex(long layerIndex)**

**说明:**

设置更新后的图层索引号

**参数:**

layerIndexLst: 图层索引号

#### **4.44.9 public long getFeatureID()**

说明:

获取要素 ID 号

#### **4.44.10 public void setFeatureID(long featureID)**

说明:

设置要素 ID 值

参数:

featureID: 要素 ID 号

#### **4.44.11 public int getPatternIndex()**

说明:

获取图案号

#### **4.44.12 public void setPatternIndex(int patternIndex)**

说明:

设置图案号

参数:

patternIndex: 图案号

#### **4.44.13 public float getMarginWidth()**

说明:

获取宽度值(点: 子图宽; 线: 外部笔宽; 区: 填充图案宽)

#### **4.44.14 public void setMarginWidth(float marginWidth)**

说明:

设置要素的宽度值(点: 子图宽; 线: 外部笔宽; 区: 填充图案宽)

参数:

marginWidth: 宽度值

#### **4.44.15 public float getMarginHeight()**

说明:

获取要素高度值

#### **4.44.16 public void setMarginHeight(float marginHeight)**

说明:

设置要素高度值

参数:

marginHeight: 高度值

#### 4.44.17 public int getColor()

说明:

获取要素的颜色值

#### 4.44.18 public void setColor(int color)

说明:

设置要素的颜色值

参数:

color: 颜色值

#### 4.44.19 public int getFeatureType()

说明:

获取要素类型

#### 4.44.20 public void setFeatureType(int featureType)

说明:

设置要素类型

参数:

featureType: 要素类型值

### 4.45 CUpdateLayerLst 类

更新多个图层信息类

示例:

```
short sFlg = viewport.UpdateLayerLst(CUpdateLayerLst updateLayerLst);
```

#### 4.45.1 public CUpdateLayerLst(int layerNum)

说明:

带参数的构造函数

参数:

layerNum: 图层个数

#### 4.45.2 public void setMapID(long mapID)

说明:

设置更新后生成对象所在地图文档 ID

参数:

mapID: 地图文档 ID

#### 4.45.3 public long getMapID()

说明:

获取更新后生成对象所在地图文档 ID

#### 4.45.4 public void setLayerNum(int layerNum)

说明:

设置图层个数

参数:

layerNum: 图层个数

#### 4.45.5 public long getLayerNum()

说明:

获取图层个数

#### 4.45.6 public void setLayerInfo(CLayerInformation[] layerInfo)

说明:

设置各个图层信息数组值

参数:

layerInfo: 各个图层信息数组

#### 4.45.7 public CLayerInformation[] getLayerInfo()

说明:

获取各个图层信息数组值

## 5.com.mapgis.ims.pro.chart 包

### 5.1 Chart 类（旧版本使用）

绘制统计分析图表类

示例:

```

if(drawType.equalsIgnoreCase("bar")){//如果直方图
    new Chart(Arrays.asList(names), Arrays.asList(values), filePath + "/" + fileName +
".htm").drawBar();
} else{//画饼图
    new Chart(Arrays.asList(names), Arrays.asList(values), filePath + "/" + fileName +
".htm").drawPie();
}

```

#### 5.1.1 Chart(List<String> recordesName, List<Double> recordesValue, String path)

说明:

图标类构造函数

参数:

recordesName: 记录名称数组, 如{"China","American","England","France"}

recordsValue: 记录的数值数组, 如{1,2,3,4,5}

path: 生成的 HTML 在磁盘上存放的位置, 如"D: \ChartFiles\"

### 5.1.2 public void drawBar()

说明:

绘制直方图

### 5.1.3 public void drawPie()

说明:

绘制饼图

### 5.1.4 public void drawCurve()

说明:

绘制折线图

## 6.com.mapgis.ims.pro.layer 包

### 6.1 CMapConfig 类

获取配置文件参数类

#### 6.1.1 public CMapConfig(String propFileName)

说明:

配置类构造函数

参数:

propFileName: 配置文件的名称, 默认是: webconfig.properties 文件

示例:

```
CMapConfig webconfig = new CMapConfig("webconfig1.properties");
```

#### 6.1.2 public CMapConfig()

说明:

配置类构造函数

参数:

无

示例:

```
CMapConfig webconfig = new CMapConfig();
```

#### 6.1.3 public String getInitParameter(String parameterName)

说明:

获取配置文件中的参数

**参数:**

parameterName: 需要获取的参数名称

**返回值:**

返回配置参数

**示例:**

```
//取得地图文档名称
CMapConfig webconfig = new CMapConfig();
String mapName = myconfig.getInitParameter("mapName");
```

#### **6.1.4 public void setInitParameter(String parameterName)**

**说明:**

设置配置文件中的参数

**参数:**

parameterName: 需要修改的参数名称

**返回值:**

无

**示例:**

```
//设置地图文档名称
CMapConfig webconfig = new CMapConfig();
String mapName = myconfig.setInitParameter("mapName","China.Map");
```

#### **6.1.5 public void removeInitParameter(String parameterName)**

**说明:**

移除配置文件中的参数

**参数:**

parameterName: 需要移除的参数名称

**返回值:**

无

**示例:**

```
CMapConfig webconfig = new CMapConfig();
String mapName = myconfig.removeInitParameter("mapName");
```

#### **6.1.6 public void storeInitParameter()**

**说明:**

保存对配置文件的修改

**参数:**

无

**返回值:**

无

示例:

```
CMapConfig webconfig = new CMapConfig();
mapconfig.setInitParameter("gisSvrIP",configInfo.getGisIP());
mapconfig.setInitParameter("MapName",mapName);
mapconfig.setInitParameter("MapType",String.valueOf(mapType));
mapconfig.storeInitParameter();
```

## 6.2 CMapDisplay 类

地图加载类

### 6.2.1 public CMapDisplay()

说明:

地图加载类构造函数

参数:

无

示例:

```
CMapDisplay mapDisplay = new CMapDisplay();
```

### 6.2.2 public MapViewPort LoadMapDisplay(String mapName)

说明:

地图加载函数

参数:

mapName: 地图文档名称

返回值:

返回 MapViewPort 类对象, 参见 4.1

示例:

```
MapViewPort viewport = null;
CMapDisplay mapDisplay = new CMapDisplay();
//加载地图
try{
    viewport = mapDisplay.LoadMapDisplay(mapName);
}catch(Exception ex){
    out.println("<script language='javascript'>alert('加载地图失败, 请确认是否有相应的地图文档。');</script>");
    return;
}
```

## 6.3 CMapLayerSetting 类

地图图层控制类

### 6.3.1 public CMapLayerSetting ()

**说明:**

地图图层类构造函数

**参数:**

无

**示例:**

```
CMapLayerSetting mapLayer = new CMapLayerSetting();
```

### 6.3.2 public String getLayerListByJSON(String HDFName CLayerInformation[] informations, int iMapType)

**说明:**

获取图层列表

**参数:**

HDFName: 所要获取图层的栅格流名称

Informations: 图层信息

iMapType: 所要加载的图层类型

**返回值:**

返回图层列表

**示例:**

```
//获取图层信息列表
```

```
CMapLayerSetting mapLayer = new CMapLayerSetting();
```

```
//获取图层类型
```

```
iMapType=Integer.parseInt(webconfig.getInitParameter("MapType").toString());
```

```
CLayerInformation[] layerInfo = null;
```

```
String HDFName = String.valueOf(webconfig.getInitParameter("HDFName"));
```

```
if(iMapType != 0){
```

```
    viewport = (MapViewPort)session.getAttribute("viewport");
```

```
    if(viewport == null){
```

```
        out.println("<script type='language'>alert('工程加载失败。')</script>");
```

```
        return;
```

```
    }
```

```
    try{
```

```
        layerInfo = viewport.getMapObj().getLayerInformationArr();
```

```
    }catch(Exception ex){
```

```
        out.println("<script type='language'>alert('获取图层信息失败。')</script>");
```

```
        return;
```

```
    }
```

```
}
```

```
//获取图层列表
```

```
res = mapLayer.getLayerListByJSON(HDFName,layerInfo,iMapType);
```

### 6.3.3 public void updateMap(String[] requestStrArray, MapViewPort viewport)

说明:

更新图层列表

参数:

requestStrArray: 状态请求的字符串

viewport: 地图视图类对象

返回值:

无

示例:

```
CMapLayerSetting mapLayer = new CMapLayerSetting();
viewport = (MapViewPort)session.getAttribute("viewport");
if (viewport == null) {
    out.println("<script type='language'>alert('工程加载失败。')</script>");
    return;
}

int selectcount = 0; //图层可查状态的个数
String strSelectIndex = "";
int iLayernum = viewport.getMapObj().getLayerNumber();
//获取图层状态
String[] strLayerStatus = new String[iLayernum];
for(int i=0; i<iLayernum; i++){
    strLayerStatus[i] = request.getParameter("layer"+i);
    if(strLayerStatus[i].equalsIgnoreCase("selectable")||strLayerStatus[i].equalsIgnoreCase("
    editable")){
        strSelectIndex += String.valueOf(i) + ",";
        selectcount++;
    }
}
//更新图层状态信息
mapLayer.updateMap(strLayerStatus,viewport);
```

## B. 客户端函数库参考

### 1. IMSMapType

地图类型枚举

```
var IMSMapType={  
Grid: 0,  
Vector: 1,  
GridAndVector: 2  
};
```

**说明:**

Grid, 栅格地图;  
Vector, 矢量地图;  
GridAndVector, 栅格叠矢量的地图

**示例:**

```
var map = new IMSMaps("zdims", IMSMapType.GridAndVector,IMSMapConfig);  
或  
var map = new IMSMaps("zdims",2,IMSMapConfig);
```

## 2. IMSOperType

地图操作类型枚举

```
var IMSOperType={  
ZoomIn: 1001,  
ZoomOut: 1002,  
Drag: 1003,  
Restore: 1004,  
Erase: 1005,  
Refresh: 1006,  
SelectByHitting: 2001,  
SelectByRect: 2002,  
SelectByCircle: 2003,  
SelectByPolygon: 2004,  
SelectByLine: 2005,  
MeasureDistance: 3001,  
MeasureArea: 3002,  
OpenMagnifier: 3003,  
AddLabel: 3004  
};
```

**说明:**

ZoomIn, 放大  
ZoomOut, 缩小  
Drag, 拖动  
Restore, 复位  
Erase, 清除

Refresh, 刷新  
SelectByHitting, 点击选择  
SelectByRect, 拉框选择  
SelectByCircle, 画圆选择  
SelectByPolygon, 多边形选择  
SelectByLine, 画线选择  
MeasureDistance, 测距  
MeasureArea, 测面积  
OpenMagnifier, 放大镜  
AddLabel, 添加标注

示例:

```
var map = new IMSMaps("zdims",2,IMSMapConfig);  
map.setCurOper (IMSOperType. ZoomIn);  
或  
var map = new IMSMaps("zdims",2,IMSMapConfig);  
map.setCurOper (1001);
```

## 3. ZDEvent

静态类，使用时无需实例化，直接调用。主要用于将方法与事件进行关联。

### 3.1 addListener

addListener (o, evType, f, capture)

**说明:**

添加 evType 事件监听方法 f 到对象 o 中去。

**参数:**

**o:** 必选项。对象。需要监听事件的对象，如 DIV、IMG 等

**evType:** 必选项。字符串。事件名称。如 mousedown、mouseup 等。

**f:** 必选项。字符串。需要绑定的事件监听方法名称。

**capture:** 可选项。布尔值。默认为 false，即只捕捉冒泡阶段的事件。如果为 true，则捕捉起泡泡阶段的事件

**返回值:**

无

**备注:**

此方法兼容 IE、FireFox 等浏览器的事件模型。如果要监听 IE 的 onmousedown 事件，则 evType 取值“mousedown”。

**示例:**

```
ZDEvent.addListener(window, "load", _initMap, false);
```

## 3.2 removeListener

removeListener (o, evType, f, capture)

**说明:**

移除对象 o 中 evType 事件监听方法 f

**参数:**

**o:** 必选项。对象。需要监听事件的对象，如 DIV、IMG 等

**evType:** 必选项。字符串。事件名称。如 mousedown、mouseup 等。

**f:** 必选项。字符串。需要绑定的事件监听方法名称。

**capture:** 可选项。布尔值。默认为 false，即只捕捉冒泡阶段的事件。如果为 true，则捕捉起泡阶段的事件

**返回值:**

无

**备注:**

此方法兼容 IE、FireFox 等浏览器的事件模型。如果要卸载在 IE 中添加的 onmousedown 事件，则 evType 取值“mousedown”。

**示例:**

```
ZDEvent.removeListener(window, "load", _initMap, false);
```

## 3.3 clearEvent

clearEvent (o, evType)

**说明:**

清除对象 o 的 evType 事件中所有的监听方法

**参数:**

**o:** 必选项。对象。需要监听事件的对象，如 DIV、IMG 等

**evType:** 必选项。字符串。事件名称。如 mousedown、mouseup 等。

**返回值:**

无

**备注:**

此方法兼容 IE、FireFox 等浏览器的事件模型。如果要监听 IE 的 onmousedown 事件，则 evType 取值“mousedown”。

**示例:**

```
ZDEvent.clearEvent(window, "load");
```

## 4. IMSMapConfig

此类用于地图的参数配置。类名可根据实际开发需要而定，但其成员变量应包含如下一个或多个。

## 4.1 IMSMapConfig

```
function IMSMapConfig()
{
this.xMinMap = 114.125602229914;           //整图的最小逻辑坐标
this.yMinMap = 30.4539323507469;
this.xMaxMap = 114.500788705197;         //整图的最大逻辑坐标
this.yMaxMap = 30.8291188260299;
this.levelNum = 6;                       //栅格地图级数
this.imgSize = 256;                      //图片大小
this.moveSpeed = 20;                     //地图移动时的速度
this.statusBarFlag=true;                 //是否在状态栏显示当前鼠标所在的
坐标
this.zoomRate = 1.5;                    //放大镜的放大比例
this.backgroundImage = "url(images/mapgis.jpg)"; //css 属性值。地图背景图片
this.errorImage = "images/transparence.gif"; //图片下载出错时的替换图片
this.mapout = "images/mapgis.jpg";      //无地图时显示的图片
this.setCoordPrec = 8;                   //设定坐标精度
this.magnifierSize = [200,150];          //放大镜的大小
this.vectImageFormat = "gif";            //矢量图格式。可取 gif、jpg、png
this.gridImageUrl = ["GetIMG.ashx"];     //栅格图请求页面
this.vectImageUrl = ["GetMap.aspx"];     //矢量图请求页面
this.getVectCoorPage = "LoadMap.aspx";  //获取矢量地图坐标范围的页面
}

```

另外，请求页面提交到服务器端的参数说明如下：

## 4.2 this.gridImageUrl

```
var url = this.gridImageUrl[layerNo]+"?layerNo="+layerNo+"&a="+rowNo+"&b="+lineNo+"&c="+levelF+"

```

其中：

layerNo: 图层序号，从 0 开始；

a: 图片所在的行号；

b: 图片所在的列号；

c: 图片所在的级数，从第一级开始。

## 4.3 this.vectImageUrl

```
var url = this.vectImageUrl[i] + "?layerNo="+i+"&LogicMinX=" + tmp[0] +

```

```
"&LogicMinY=" + tmp[1] + "&LogicMaxX=" + tmp[2] + "&LogicMaxY=" +
tmp[3]+ "&ImageWidth="+this.div.clientWidth+"&ImageHeight="+this.div.clientHeight+"&I
mageFormat=gif&rnd=" + Math.random());
```

其中:

layerNo: 图层序号, 从 0 开始;

LogicMinX、LogicMinY、LogicMaxX、LogicMaxY: 所请求的地图矩形范围, 为矩形左下右上的坐标;

ImageHeight、ImageWidth: 所请求地图图片的高宽;

ImageFormat: 所请求地图图片的格式。

rnd: 随机数

IsEagle: 如果为鹰眼取图, 则包含此参数, 取值为 true, 表示是鹰眼取图。

## 4.4 this.getVectCoorPage

```
var url =
this.getVectCoorPage+"?w="+this.div.offsetWidth+"&h="+this.div.offsetHeight+"&rnd="+M
ath.random();
```

其中:

w: 所请求地图图片的宽度;

h: 所请求地图图片的高度;

rnd: 随机数

## 5. IMSMaps

核心对象, 每个实例代表页面上的一幅地图。

### 5.1 IMSMaps

IMSMaps(pDiv,mapType, mapConfig)

**说明:**

IMSMaps 类的构造函数

**参数:**

pDiv: 必选项。DIV 层对象或 DIV 层的 ID 名。需附加地图的 DIV 层。

mapType: 必选项。IMSMapType 枚举值或数字。地图类型。

0: 栅格; 1: 矢量; 2: 栅格叠矢量。

mapConfig: 必选项。配置类名, 类结构与 IMSMapConfig 相同。

注: 此处为类名, 非实例化后的对象名。

**返回值:**

无

备注:

无

示例:

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
```

## 5.2 setRestoreParam

setRestoreParam(level)

说明:

设置地图复位时的级数

参数:

level: 必选项。数字。地图复位时的级数

返回值:

无

备注:

无

示例:

```
var map = new IMSMaps("zdims",2,IMSMapConfig);  
map.setRestoreParam(114.2015,14.5698,2);
```

## 5.3 setRestoreParam

setRestoreParam(x,y,level)

说明:

设置地图复位时窗口中心点的逻辑坐标和级数参数

参数:

x,y: 必选项。浮点型数字。地图复位时窗口中心点对应的逻辑坐标。

level: 必选项。数字。地图复位时的级数

返回值:

无

备注:

无

示例:

```
var map = new IMSMaps("zdims",2,IMSMapConfig);  
map.setRestoreParam(114.2015,14.5698,2);
```

## 5.4 restoreMap

restoreMap()

说明:

刷新复位地图

**参数:**

无

**返回值:**

无

**备注:**

地图根据设定的复位参数，刷新地图，如未指定复位参数，则地图将以整幅地图的中心点作为复位后的窗口中心，并以第 1 级地图作为复位后地图的显示级别。

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
map.restoreMap();
```

## 5.5 jumpMap

jumpMap(x,y,level)

**说明:**

根据坐标和级数跳转地图

**参数:**

x,y: 必选项。浮点型数字。地图跳转后显示窗口中心对应的逻辑坐标。

level: 必选项。整型数字。跳转显示的地图级数

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
map.jumpMap(114.252,115.2469,5);
```

## 5.6 jumpMapByLev

jumpMapByLev(count)

**说明:**

直接放大到固定级数

**参数:**

count: 必选项。整型数字。地图跳转后的级数

**返回值:**

无

**备注:**

该操作保证地图放大后窗口中心对应的逻辑坐标不变。

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
map.jumpMapByLev(5);
```

## 5.7 pntLocation

pntLocation (x,y)

**说明:**

根据坐标跳转地图

**参数:**

x,y: 必选项。数字。地图将要跳转的逻辑坐标位置

**返回值:**

无

**备注:**

该方法用于地图在同一级数内进行跳转，根据指定的逻辑坐标将此处地图移至窗口中心。

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
map.pntLocation (114.256,113.2564);
```

## 5.8 setCurOper

setCurOper(operType)

**说明:**

设置当前操作状态

**参数:**

operType: 必选项。对应 IMSOperType 枚举类型或数字。

操作类型如下所示:

操作说明	枚举常量名称	枚举常量序数
地图放大	ZoomIn	1001
地图缩小	ZoomOut	1002
地图拖拽	Drag	1003
复位地图	Restore	1004
清除绘图	Erase	1005
刷新地图	Refresh	1006
点击查询	SelectByHitting	2001
拉框查询	SelectByRect	2002
画圆查询	SelectByCircle	2003
多边形查询	SelectByPolygon	2004
线查询	SelectByLine	2005
距离测量	MeasureDistance	3001
面积测量	MeasureArea	3002
放大镜	OpenMagnifier	3003
添加标注	AddLabel	3004

返回值:

无

备注:

无

示例:

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
map.setCurOper(IMSOperType.ZoomIn);
```

## 5.9 screenToLogic

screenToLogic(x,y)

说明:

窗口坐标转逻辑坐标

参数:

x,y: 必选项。整形数字。此点代表地图显示层上的像素坐标位置。

返回值:

数组。[lx,ly], x,y 在地图上对应的逻辑坐标。

备注:

示例:

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var lxy=map.screenToLogic(200,300);
```

## 5.10 logicToScreen

logicToScreen(x,y)

说明:

逻辑转窗口坐标

参数:

x,y: 必选项。数字。某点的逻辑坐标值。

返回值:

数组。[wx,wy], 指定逻辑位置在当前地图显示层上对应的像素坐标位置。

备注:

无

示例:

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var wxy=map.logicToScreen(114.256,113.2564);
```

## 5.11 addControl

addControl(obj,isCanMove)

说明:

添加控件

**参数:**

obj: 必选项。控件对象, 一般为 div 层。

isCanMove: 必选项。布尔类型。控件对象是否需要随地图的移动而移动, true: 跟随; false: 不跟随。

**返回值:**

无

**备注:**

添加控件主要是指将用于一定功能的 div 层 (如: 导航类、比例尺等), 根据其是否需要跟随地图移动 (如: 点位信息层需要跟随移动), 从而将其附加到不同的地图功能层上。

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var marker = new IMSMarker(false);
map.addControl(marker,true);
```

## 5.12 removeControl

removeControl(obj,isCanMove)

**说明:**

移除控件

**参数:**

obj: 必选项。控件对象, 一般为 div 层。

isCanMove: 必选项。bool 类型。控件对象是否需要随地图的移动而移动, true: 跟随; false: 不跟随。

**返回值:**

无

**备注:**

将控件从地图功能层上卸载下来, 即将控件对应的 div 从父节点上去除。

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var marker = new IMSMarker(false);
map.addControl(marker,true);
map.removeControl(marker,true);
```

## 5.13 loadAllCtl

loadAllCtl()

**说明:**

添加所有地图控件, 包括中心点、标尺、比例尺、Logo、测量、鹰眼等控件。

**参数:**

无

**返回值:**

无

**备注:**

当加载所有地图控件后，地图内容将变得十分丰富。

示例：

```
var map = new IMSMaps("zdims",2,IMSMapConfig);  
map.loadAllCtl();
```

## 5.14 getCurLev

getCurLev()

说明：

获得当前地图显示级数

参数：

无

返回值：

数字型。当前地图的显示级数。

备注：

无

示例：

```
map = new IMSMaps("zdims",2,IMSMapConfig);  
var Level = map.getCurLev();
```

## 5.15 refresh

说明：

刷新地图

参数：

无

返回值：

无

备注：

执行刷新操作后，客户端将向服务端重新发出图片请求，重绘地图，如果存在矢量地图则从新获取矢量地图与栅格地图一同显示。

示例：

```
var map = new IMSMaps("zdims",2,IMSMapConfig);  
map.refresh();
```

## 5.16 moveTo

moveTo(lx,ly)

说明：

移动地图，直到指定逻辑位置位于窗口中心。

参数：

lx,ly: 必选项。数字。窗口中心将要显示的逻辑坐标。

返回值:

无

备注:

pntLocation 是直接跳转地图，无平滑过渡效果。moveTo 是平滑移动地图到指定的位置

示例:

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
map.moveTo(114.235,113.236);
```

## 5.17 getClickCoor

getClickCoor()

说明:

获取点击处的逻辑坐标

参数:

无

返回值:

[lx,ly]: 数组类型；当前鼠标点击处的逻辑坐标。

备注:

无

示例:

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var lxy = map.getClickCoor ();
```

## 5.18 getCurWinLog

getCurWinLog()

说明:

获得当前地图显示窗口的逻辑范围

参数:

无

返回值:

[xMinMap,yMinMap,xMaxMap,yMaxMap]: 数组型。

备注:

窗口的逻辑范围通过记录窗口左下角的逻辑坐标以及右上角的逻辑坐标来表示的。

示例:

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var range = map.getCurWinLog();
xmin = range[0]; ymin = range[1]; xmax= range[2]; ymax=range[3];
```

## 5.19 getCenterPnt

getCenterPnt()

**说明:**

获取地图显示窗口中心位置的逻辑坐标。

**参数:**

无

**返回值:**

[centerX,centerY]: 数组型; 返回窗口中心点的逻辑坐标。

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
map.getCenterPnt();
```

## 5.20 getMapViewSize

getMapViewSize()

**说明:**

获取地图可视范围, 为浏览器窗口像素值

**参数:**

无

**返回值:**

[width,height]: 数组类型; 该值记录地图显示窗口的宽、高。

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
map.getMapViewSize ();
```

## 5.21 closeGridLayer

closeGridLayer(index)

**说明:**

关闭某层栅格地图。

**参数:**

index: 必选项。数字。栅格数据的图层序号。

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
for(var i=0;i<map.gridImageUrl.length;i++)
{
    map.closeGridLayer(i);
}
```

```
}
```

以上代码表示循环关闭所有栅格图层。

## 5.22 openGridLayer

```
openGridLayer(index,url)
```

**说明:**

开启某一栅格图层进行显示

**参数:**

**index:** 必选项。数字。栅格数据的图层序号。

**url:** 可选项。字符串。请求地图的页面网址，如: "GetIMG.ashx"。

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);  
map.openGridLayer(2);
```

## 5.23 openVectMap

```
openVectMap(index)
```

**说明:**

开启矢量地图文档，显示文档中的图层。

**参数:**

**index:** 必选项。数字。地图文档序号。

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);  
map.openVectMap(2);
```

## 5.24 closeVectLayer

```
closeVectLayer(index)
```

**说明:**

关闭地图文档，不显示其相应的图层。

**参数:**

**index:** 必选项。数字。地图文档序号。

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);  
map.closeVectLayer (2);
```

## 5.25 getGridCode

getGridCode(x,y)

**说明:**

根据 x,y 坐标，获在当前地图显示级数下，点所在的网格号

**参数:**

x,y: 必选项。数字。指定点的逻辑坐标。

**返回值:**

gridcode: 数字。记录 x,y 坐标在当前级数下对应的网格号。

**备注:**

每一级的网格号排列是以左下角为起始点，按从左到右、从下到上的顺序进行编号。每层的起始网格编号为 0。获得网格号，便于数据检索。

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);  
var gridcode = map.getGridCode(114.2563,113.589);
```

## 5.26 getGridCodes

getGridCodes (sx,sy,ex,ey)

**说明:**

获取当前级数下指定范围的网格编号

**参数:**

sx,sy,ex,ey: 必选项。数字。sx,sy 为指定范围左下角的逻辑坐标； ex,ey 为指定范围右上角的逻辑坐标。

**返回值:**

gridCodes: 字符串型；由多个网格编号组成,且网格编号间用逗号隔开，

如：gridcode1,girdcode2,...,gridcode10

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);  
var gridCodes = map.getGridCodes(114.2563,113.589,114.369,115.264);
```

## 5.27 gridCodeConvert

gridCodeConvert(index,baseLevel,currentLevel)

**说明:**

级数变换时图片所在网格互算

**参数:**

**index:** 必选项。字符串型。当前网格编号(图片或图片所在的单元格TD的ID号),形式为:行号&列号&级数。

**baseLevel:** 必选项。数字。转换之后的级数。

**currentLevel:** 必选项,数字。转换之前的级数。

**返回值:**

**gridCodes:** 级数由大变小,为网格号数字;由小变大,为网格号数组。

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var id = rowNo+"&"+lineNo+"&"+parseInt(level+1);
var gridCodes = map.gridCodeConvert(id,thisObj.levelNum,(level+1));
```

## 5.28 addEventListener

addEventListener(evType, f, capture)

**说明:**

添加地图事件监听方法

**参数:**

**evType:** 必选项。字符串。事件名称。如 mousedown、mouseup 等。

**f:** 必选项。字符串。需要添加的事件监听方法名称。

**capture:** 可选项。布尔值。默认为 false,即只捕捉冒泡阶段的事件。如果为 true,则捕捉起泡泡阶段的事件

**返回值:**

无

**备注:**

此方法兼容 IE、FireFox 等浏览器的事件模型。如果要监听 IE 的 onmousedown 事件,则 evType 取值“mousedown”。

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
map.addEventListener("mouseup",SelectByHitting, false);
```

## 5.29 removeEventListener

removeEventListener(evType, f, capture)

**说明:**

移除地图事件监听方法

**参数:**

evType: 必选项。字符串。事件名称。如 mousedown、mouseup 等。

f: 必选项。字符串。需要添加的事件监听方法名称。

capture: 可选项。布尔值。默认为 false，即只捕捉冒泡阶段的事件。如果为 true，则捕捉起泡泡阶段的事件。

**返回值:**

无

**备注:**

此方法兼容 IE、FireFox 等浏览器的事件模型。如果要移除添加在 IE 中的 onmousedown 事件，则 evType 取值“mousedown”。

**示例:**

```
map.removeEventListener ("mouseup",thisObj.so.selectByHitting , false);
```

## 5.30 setRectDrawStyle

setRectDrawStyle(lineColor,lineWidth,fillColor,fillOpacity)

**说明:**

设置矩形绘图风格

**参数:**

lineColor: 必选项。字符串。矩形框边线颜色（如“red”、“#f06”）。

lineWidth: 必选项。字符串。矩形框边线宽度（如“1px”）。

fillColor: 必选项。字符串。矩形内部填充颜色（如“red”、“#f06”）。

fillOpacity : 必选项。字符串。填充颜色透明度（如“0.6”）。

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
```

```
map.setRectDrawStyle("#f06","1px ","#119 ","0.3 ");
```

## 5.31 setCircleDrawStyle

setCircleDrawStyle(lineColor,lineWidth,fillColor,fillOpacity)

**说明:**

设置圆绘图风格

**参数:**

lineColor: 必选项。字符串。圆边线颜色（如“red”、“#f06”）。

lineWidth: 必选项。字符串。圆边线宽度（如“1px”）。

fillColor: 必选项。字符串。圆内部填充颜色（如“red”、“#f06”）。

fillOpacity: 必选项。字符串。填充颜色透明度（如“0.6”）。

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
map.setCircleDrawStyle("#f06","1px","#119","0.3");
```

## 5.32 setLineDrawStyle

setLineDrawStyle(lineColor,lineWidth)

**说明:**

设置线绘图风格

**参数:**

lineColor: 必选项。字符串。线颜色（如”red”,”#f06”）。

lineWidth: 必选项。字符串。线宽度（如”2px”）。

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
map.setLineDrawStyle("#f06","1px");
```

## 5.33 setPolygonDrawStyle

setPolygonDrawStyle(lineColor,lineWidth,fillColor,fillOpacity)

**说明:**

设置多边形绘图风格

**参数:**

lineColor: 必选项。字符串。多边形边线颜色（如”red”,”#f06”）。

lineWidth: 必选项。字符串。多边形边线宽度（如”2px”）。

fillColor: 必选项。字符串。多边形内部填充颜色（如”red”,”#f06”）。

fillOpacity: 必选项。字符串。填充颜色透明度（如”0.6”）。

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
map.setCircleDrawStyle("#f06","1px","#119","0.3");
```

## 5.34 clearOper

clearOper()

**说明:**

清楚操作遗留在地图上的痕迹

**参数:**

无

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
map.clearOper();
```

## 事件

IMSMaps 类支持如下事件。

事件名称	说明
mousedown	鼠标在地图上按下时触发
mousemove	鼠标在地图上移动时触发
mouseup	鼠标在地图上弹起时触发
dblclick	鼠标在地图上双击时触发
mouseout	鼠标移出地图范围时触发
zoom	地图缩放时触发
resize	地图大小变化时触发
keypress	在地图上按键时触发

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
map.addEventListener("zoom",MdotLocation,false);
function MdotLocation(){...};
```

## 6. IMSNavigation

地图导航类，主要涉及到地图的按级数缩放、地图的快速移动、复位等功能。

**注:** 仅适用于 IMSMapType.GridAndVector、IMSMapType.Grid

## 6.1 IMSNavigation

IMSNavigation(barType)

**说明:**

导航类

**参数:**

barType: 导航类型; 0: 带级数条的导航标尺;1: 不带级数条的迷你导航标尺; 默认情况下 barType 的值为 0。

**返回值:**

无

**备注:**

在使用地图导航功能前必须对导航类进行实例化, 并将其作为控件添加到 map 对象中。

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var nav= new IMSNavigation();
map.addControl(nav,false);
```

## 6.2 setImage

setImage(up\_url,right\_url,down\_url,left\_url,restore\_url,zoom\_in\_url,  
zoom\_out\_url,cursor\_url,node\_url)

**说明:**

设置导航条的图片

**参数:**

up\_url、right\_url、down\_url、left\_url: 可选项。字符串。分别为上、右、下、左四方向快速移动地图的图片地址。

restore\_url: 可选项。字符串。刷新地图的图片地址。

zoom\_in\_url、zoom\_out\_url: 可选项。字符串。放大、缩小地图操作的图片地址。

cursor\_url: 可选项。字符串。级数条图片地址。

node\_url: 可选项。字符串。级数条背景图片地址。

**返回值:**

无

**备注:**

创建导航类对象后再进行导航条图片设置。

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var nav= new IMSNavigation();
map.addControl(nav,false);
nav.setImage("image/up.gif","image/rigth.gif","image/down.gif","image/left.gif");
```

## 6.3 setControlPos

setControlPos(left,top)

**说明:**

设置导航条的位置

**参数:**

left: 必选项。字符串。导航条距地图显示窗口左边的距离如，如 (“20px”)。

top: 必选项。字符串。导航条地图显示窗口上边界的距离，如 (“30px”)。

**返回值:**

无

**备注:**

无

**示例:**

无

## 6.4 moveSpeed

moveSpeed(speed)

**说明:**

设置方向导航地图移动的速度

**参数:**

speed: 必选项。数字。方向导航地图移动的速度。speed 值越大地图移动越快，默认为 30。

**返回值:**

无

**备注:**

无

**示例:**

无

## 6.5 setMapShade

setMapShade (flag)

**说明:**

设置拖动滑块地图是否有渐变效果

**参数:**

flag: 必选项。布尔值。默认为 true。true, 有渐变效果; false, 无渐变效果。

**返回值:**

无

**备注:**

无

**示例:**

无

## 7. IMSMarker

### 7.1 IMSMarker

IMSMarker (saveFlag)

**说明:**

IMSMarker 类的构造函数

**参数:**

saveFlag: 可选。标志是否将标注的点提交到服务器端保存, true 为提交, false 为不提交。默认为不提交到服务器端保存。

**返回值:**

无

**备注:** 标注类, 使用时需要先实例化。

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var marker = new IMSMarker(false);
map.addControl(marker,true);
```

### 7.2 setMarkerImg

setMarkerImg(url)

**说明:**

设置标记使用的图片

**参数:**

url: 必选项。标记使用图片的路径

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var marker = new IMSMarker(false);
var url = "images/markerImg.jpg"
marker.setMarkerImg(url);
map.addControl(marker,true);
```

### 7.3 setURL

setURL(url)

**说明:**

设置服务器处理页面

**参数:**

**url:** 必选项。服务器处理页面的路径

**返回值:**

无

**备注:**

该方法在需要将标注点提交到服务器端保存，即 saveFlag 为 true 时才用到。

如果提交，则服务器端获得的参数为：

```
var data = "_method=SendLable+"&lableName="+name+"&lableDetail="+
details+"&lableID="+dotIndex+"&levelNo="+levelNo+"&posX="+XLogic+"&posY="+YLo
gic;
```

其中：

**\_method:** 提交请求的名称

**lableName:** 标注名称

**lableDetail:** 标注的详细信息

**lableID:** 标注 ID

**levelNo:** 标注所在的级数

**posX, posY:** 标注点所在的 X、Y 坐标

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var marker = new IMSMarker(true);
var url = " Maps.ashx"
marker.setURL(url);
map.addControl(marker,true);
```

## 7.4 setStyle

setStyle(style)

**说明:**

设置标注对话框的样式

**参数:**

**style:** 必选项。标注对话框的样式，为一段 html 代码。

**返回值:**

无

**备注:**

调用该方法设置自定义对话框样式后，需要编写相应的方法，不推荐初级用户使用

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var marker = new IMSMarker(false);
var style = "<div style='.....'></div>"
marker.setURL(style);
map.addControl(marker,true);
```

## 7.5 setMarkerTitle

setMarkerTitle(text)

**说明:**

设置标注对话框的标题

**参数:**

text: 必选项。标注对话框的标题，为任意长度的字符串。

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var marker = new IMSMarker(false);
var text = " 中地数码"
marker.setMarkerTitle(text);
map.addControl(marker,true);
```

## 7.6 getMarkerPoint

getMarkerPoint(type)

**说明:**

获取标注点的点坐标

**参数:**

type: 可选项。标志返回逻辑坐标还是窗口坐标，true 返回逻辑坐标，false 返回窗口坐标。默认返回窗口坐标。

**返回值:**

返回一个数组，第一个元素即 array[0]为横坐标，第二个元素即 array[1]为纵坐标。

**备注:**

调用该方法前，必须已经在图上标识了点，否则没有返回值。

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var marker = new IMSMarker(false);
map.addControl(marker,true);
marker.getMarkerPoint();
```

## 8. IMSScale

### 8.1 IMSScale

IMSScale()

**说明:**

比例尺类，用于显示地图在不同比例下的比例尺。

**参数:**

无

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var sc = new IMSScale();
sc.setUnits([103133.845,1000],["米","千米"]);
map.addControl(sc,false);
```

## 8.2 setUnits

setUnits (valueArr,unitArr)

**说明:**

设定每像素所代表的长度。

**参数:**

valueArr: 必选项。值数组。

valueArr[0]记录当前坐标系下的比例度量，例如经纬度的地图 1 度代表的长度约为 103133.845 米，那么 valueArr[0] 值为 103133.845。如果地图为大地坐标系那么 valueArr[0] 值为 1。

valueArr[1]记录比例尺的比例，如果是米到千米的转换那么 valueArr[1] 值为 1000；

unitArr: 必选项。单位数组，是 valueArr 中值之间的转换度量的汉字表示，如“米”到“千米”或则“平方米”到“平方千米”；

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var sc = new IMSScale();
sc.setUnits([103133.845,1000],["米","千米"]);
map.addControl(sc,false);
```

## 8.3 setPos

setPos(left,top)

**说明:**

设定比例尺位置

**参数:**

left: 必选项。字符串。比例尺距地图显示窗口左边的距离, 如(“300px”)。

top: 必选项。字符串。比例尺距地图显示窗口上边界的距离,如(“300px”)。

返回值:

无

备注:

无

示例:

无

## 8.4 setFontColor

setFontColor(color)

说明:

设定比例尺字体颜色

参数:

color: 必选项。字符串。字体颜色值如 (“#e97575”)。

返回值:

无

备注:

无

## 9. IMSLogo

### 9.1 IMSLogo

IMSlogo()

说明:

logo 标识类。加版权标识到地图上。

参数:

无

返回值:

无

备注:

无

示例:

```
var map = new IMSMaps("zdims",2,IMSMapConfig);  
var l = new IMSLogo();  
map.addControl(l,false);
```

### 9.2 setlogo

setlogo (link,tip,imgSrc)

**说明:**

设置 Logo

**参数:**

link: 可选项。字符串。点击 Logo, 弹出的链接地址。

tip: 可选项。字符串。当鼠标移动到 Logo 上的提示。

imgSrc: 可选项。字符串。背景图片链接地址。

**返回值:**

无

**备注:**

无

## 9.3 setPos

setPos(left,top)

**说明:**

设置 logo 的位置

**参数:**

left: 必选项。字符串。logo 距地图显示窗口左边的距离。

top: 必选项。字符串。logo 地图显示窗口上边界的距离。

**返回值:**

无

**备注:**

无

## 10. IMSSearch

该类可用于矩形查询, 圆查询, 多边形查询, 点查询等图形搜索相关功能。此类与服务端页面相关联。应用时, 一般在服务器端添加 asp.net 自定义处理程序“Maps.ashx”, 用于处理客户端的请求。也可用于其他的图形操作功能。

### 10.1 IMSSearch

IMSSearch()

**说明:**

查询类构造函数

**参数:**

无

**返回值:**

无

**备注:**

在使用查询功能前需要对查询类进行实例化, 并作为控件添加到 map 对象中。

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var search = new IMSSearch(map);
search.setSelectByHitting(3,rect);
search.setSelectByRect(rect);
map.addControl(search,false);
```

```
function rect(para)
{
    alert(para);
}
```

## 10.2 setSelectByHitting

setSelectByHitting(ioffset,dealFun)

**说明:**

设置点击查询的功能函数

**参数:**

ioffset: 必选项, 数字。点击处向外扩展的半径, 其大小根据地图逻辑范围而定。

dealFun: 必选项, 点击查询参数处理函数, 用于传送点击查询的各项参数到指定页面。

**返回值:**

无

**备注:**

dealFun 函数可模拟示例站点中的写法也可以根据需求另外编写。

setSelectByHitting 函数的设置在初始化查询类后进行。

其返回给 dealFun 的值的格式为:

“centerX=...&centerY=...&radius=...&GridCodes=...&lev=...&sType=SelectByHitting”

其中:

centerX, centerY: 点击出的坐标;

radius: 扩展半径;

GridCodes: 点所在的网格;

lev: 当前的级数;

sType: 操作类型

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
search = new IMSSearch(map);
search.setSelectByHitting(3,rect);
map.addControl(search,false);
```

## 10.3 setSelectByRect

setSelectByRect (dealFun)

**说明:**

设置拉框查询的参数处理函数

**参数:**

dealFun: 必选项, 参数处理函数, 参数包括拖拽矩形框时鼠标按下与弹起时的逻辑坐标、矩形框所占的网格号、当前显示级数以及操作类型。

**返回值:**

无

**备注:**

其返回给 dealFun 的值的格式为:

```
"startX =...& startY =...& endX =...&endY= ...&GridCodes=...&lev=...&sType=
SelectByRect "
```

其中:

startX、startY、endX、endY: 拉框矩形的左下右上的坐标;

GridCodes: 矩形所占的网格编号字符串;

lev: 当前的级数;

sType: 操作类型

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
search = new IMSSearch(map);
search.setSelectByRect(rect);
map.addControl(search,false);
```

## 10.4 setSelectByCircle

setSelectByCircle(dealFun)

**说明:**

设置画圆查询的参数处理函数

**参数:**

dealFun: 必选项, 参数处理函数, 参数包括画圆时鼠标按下点的逻辑坐标、圆的半径值、圆所占的网格号、当前显示级数以及操作类型。

**返回值:**

无

**备注:**

其返回给 dealFun 的值的格式为:

```
"centerX=...&centerY=...&radius=...&GridCodes=...&lev=...&sType= SelectByCircle "
```

其中:

centerX, centerY: 圆心的坐标;

radius: 圆的半径;

GridCodes: 圆所在的网格编号字符串;

lev: 当前的级数;

sType: 操作类型

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
search = new IMSSearch(map);
search.setSelectByCircle(rect);
map.addControl(search,false);
```

## 10.5 setSelectByPolygon

setSelectByPolygon(dealFun)

**说明:**

设置多边形查询的参数处理函数

**参数:**

**dealFun:** 必选项, 参数处理函数, 参数包括多边形各顶点的逻辑坐标、多边形所占的网格号、当前显示级数以及操作类型。

**返回值:**

无

**备注:**

其返回给 dealFun 的值的格式为:

```
"XArray =...& YArray =...&GridCodes=...&lev=...&sType= SelectByPolygon "
```

其中:

XArray, YArray: 多边形 xy 坐标串;

GridCodes: 圆所在的网格编号字符串;

lev: 当前的级数;

sType: 操作类型

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
search = new IMSSearch(map);
search.setSelectByPolygon(rect);
map.addControl(search,false);
```

## 10.6 setSelectByLine

setSelectByLine(ioffset,dealFun)

**说明:**

设置线查询的参数处理函数

**参数:**

**ioffset:** 必选项, 数字。线查询半径;

**dealFun:** 必选项, 参数处理函数, 参数包括组成折线的各点的逻辑坐标、折线所占的网格号、当前显示级数以及操作类型。

**返回值:**

无

**备注:**

其返回给 dealFun 的值的格式为:

```
"XArray =...& YArray =...&GridCodes=...&lev=...&sType= SelectByLine"
```

其中:

XArray, YArray: 多边形 xy 坐标串;

GridCodes: 圆所在的网格编号字符串;

lev: 当前的级数;

sType: 操作类型

示例:

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
search = new IMSSearch(map);
search.setSelectByLine(rect);
map.addControl(search,false);
```

## 11. IMSVIPnts

重要信息点类。添加总要信息点至地图上，且随地图一起放大缩小。

仅适用于：IMSMapType.Grid、IMSMapType.GridAndVector

### 11.1 IMSVIPnts

说明:

重要信息点类构造函数

参数:

无

返回值:

无

备注:

示例:

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var vip = new IMSVIPnts();
map.addControl(vip,true);
```

### 11.2 setImgSrc

setImgSrc (src,imgTooltip,onclickFun)

说明:

设置图标的相关配置

参数:

src : 必选项，字符型。Vip 点图标的链接;

imgTooltip: 必选项，字符型。鼠标移到图标上的提示;

onclickFun: 必选项，函数句柄。鼠标点击图标时触发执行的函数;

返回值:

无

备注:

无

示例:

无

## 11.3 loadData

loadData(requestPage)

**说明:**

载入数据，返回的数据为 JSON 格式

**参数:**

requestPage: 必选项，字符型。数据请求的页面；

**返回值:**

无

**备注:**

服务器端返回的数据应该为 JSON 的写法。且对应的每个信息点的结构至少包含如下结构。

```
function VipPnt(x,y,name)
```

```
{  
this.lx= x;  
this.ly=y;  
this.name=name;  
}
```

如返回的数据为:

```
[{"lx",114.2966;"ly",30.5811;"name","武汉客运港"}, {"lx",114.2992;"ly",  
30.5468;"name","黄鹤楼公园"}]
```

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);  
var vo = new IMSVIPPnts();  
vo.setImgSrc("images/p1.gif","asdafafd",voFun);  
map.addControl(vo,true);  
vo.loadData("Maps.ashx");  
vo.showPnts();
```

## 11.4 showPnts

showPnts()

**说明:**

显示信息点

**参数:**

无

**返回值:**

无

**备注:**

无

**示例:**

无

## 11.5 hidePnts

hidePnts ()

**说明:**

隐藏信息点

**参数:**

无

**返回值:**

无

**备注:**

无

**示例:**

无

## 11.6 clearPnts

clearPnts()

**说明:**

清除信息点

**参数:**

无

**返回值:**

无

**备注:**

无

**示例:**

无

## 12. IMSMeasure

测量类，主要负责距离测量与面积测量

### 12.1 IMSMeasure

**说明:**

测量初始化类

**参数:**

无

**返回值:**

无

**备注:**

在使用测量功能前需要对测量类进行实例化，并作为控件添加到 `map` 对象中。

示例：

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var mo = new IMSMeasure();
mo.setDistanceUnits([1,1000],["米","千米"]);
mo.setAreaUnits([10636589908.557,1000000],["平方米","平方千米"]);
map.addControl(mo,false);
```

## 12.2 setDistanceUnits

`setDistanceUnits(valueArr,unitArr)`

**说明：**

设定测距的单位

**参数：**

`valueArr`：必选项。值数组。

`valueArr[0]`记录当前坐标系下的比例度量，例如经纬度的地图 1 度代表的长度为 103133.845 米，那么 `valueArr[0]`值为 103133.845。如果地图为大地坐标系那么 `valueArr[0]`值为 1。

`valueArr[1]`记录比例尺的比例，如果是米到千米的转换那么 `valueArr[1]`值为 1000；  
`unitArr`：必选项。单位数组，是 `valueArr` 中值之间的转换度量的汉字表示，如“米”到“千米”或则“平方米”到“平方千米”；

**返回值：**

无

**备注：**

无

**示例：**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var mo = new IMSMeasure();
mo.setDistanceUnits([1,1000],["米","千米"]);
map.addControl(mo,false);
```

## 12.3 setAreaUnits

`setAreaUnits(valueArr,unitArr)`

**说明：**

设定测面积的单位

**参数：**

`valueArr`：必选项。值数组。

`valueArr[0]`记录当前坐标系下的比例度量，例如经纬度的地图 1 度代表的长度为 103133.845 米，那么 `valueArr[0]`值为  $103133.845 * 103133.845$ 。如果地图为大地坐标系那么 `valueArr[0]`值为  $1 * 1$ 。

`valueArr[1]`记录比例尺的比例，如果是“平方米”到“平方千米”的转换那么 `valueArr[1]`值为 1000000；

unitArr: 必选项。单位数组, 是 valueArr 中值之间的转换度量的汉字表示, 如“米”到“千米”或则“平方米”到“平方千米”;

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",2,IMSMapConfig);
var mo = new IMSMeasure();
mo.setAreaUnits([10636589908.557,1000000],[ "平方米","平方千米"]);
map.addControl(mo,false);
```

## 12.4 setTooltipStyle

setTooltipStyle(border,backgroundColor,fontColor)

**说明:**

设定提示框的风格

**参数:**

border: 必选项。字符串。边框风格, 如"1px outset #ddd";

backgroundColor: 必选项。字符串。提示框的背景颜色, 如"#FFFFFF";

fontColor: 必选项, 字符串。字体颜色, 如" black";

**返回值:**

无

**备注:**

无

**示例:**

无

## 13. IMSPalette

画图类, 主要用于线、框、圆、区等图形的绘画

### 13.1 IMSPalette

IMSPalette(pElement,drawType)

**说明:**

画图类。

**参数:**

pElement: 必选项。被画的对象, 比如在某个 DIV 层上画图, 则此 DIV 层即为 pElement。

drawType: 必选项。字符串。画图类型。有矩形"rect", 圆" circle", 线" polyline", 区" area" 四种。

**返回值:**

画图对象

**备注:**

无。

**示例:**

```
var drawDiv = document.createElement("DIV")
drawDiv.id = "draw";
var Obj = new IMSPalette(drawDiv,"rect");
```

## 13.2 setStyle

setStyle(lineColor,lineWidth,fillColor,fillOpacity)

**说明:**

设置画图表现样式

**参数:**

lineColor: 可选项。字符串。线的颜色（如"red","#f06"）。

lineWidth: 可选项。字符串。线的宽度。（如"3px"）

fillColor: 可选项。字符串。区的填充色（如"red","#f06"）。

fillOpacity : 可选项。字符串。区的透明度（如"0.6"）。

**返回值:**

画图对象。

**备注:** 无。

**示例:**

```
var drawDiv = document.createElement("DIV");
drawDiv.id = "draw";
var lineVml = new IMSPalette(drawDiv,"polyline");
lineVml.setStyle("red",3px,"#88ff44","0.5");
```

## 13.3 draw

draw(dotInfo)

**说明:**

根据传入的坐标数组画图。一次性的根据坐标数组画出成形的图形。

**参数:**

dotInfo: 必选项。由(x,y)对组成的坐标数组，其中 x,y 为窗口坐标

**返回值:**

无

**备注:**

无

**示例:**

```
var drawDiv = document.createElement("DIV")
drawDiv.id = "draw";
var lineDotInfo = new Array();
```

```
lineDotInfo.push(x1+",""+y1);  
lineDotInfo.push(x2+",""+y2);  
var lineVml = new IMSPalette(drawDiv,"polyline");//vml 对象（画道路曲线）  
lineVml.draw(lineDotInfo);
```

## 13.4 setStartPnt

```
setStartPnt(x,y)
```

### 说明：

设置画图起点坐标

### 参数：

x：必选项。数字类型或字符串类型。窗口 X 坐标。

y：必选项。数字类型或字符串类型。窗口 Y 坐标。

### 返回值：

无

### 备注：

无

### 示例：

```
var drawDiv = document.createElement("DIV")  
drawDiv.id = "draw";  
var lineVml = new IMSPalette(drawDiv,"polyline");  
lineVml.setStartPnt(x1,y1);
```

## 13.5 drawing

```
drawing(x,y)
```

### 说明：

根据传入的坐标画图。适用于边点击边画图类型，如测量中的画线操作。

### 参数：

x：必选项。数字类型或字符串类型。窗口 X 坐标。

y：必选项。数字类型或字符串类型。窗口 Y 坐标。

### 返回值：

无。

### 备注：

调用前必须先调用 setStartPnt()或 appendPoint()设置起点。

### 示例：

```
var drawDiv = document.createElement("DIV")  
drawDiv.id = "draw";  
var lineVml = new IMSPalette(drawDiv,"polyline");  
lineVml.setStartPnt(x1,y1);  
lineVml.drawing(x2,y2);
```

## 13.6 appendPoint

appendPoint(x,y)

**说明:**

将确定后的坐标加入已保存的坐标数组中，主要用于画线、画区的操作，比如测距操作。

**参数:**

x：必选项。数字类型或字符串类型。窗口 X 坐标。

y：必选项。数字类型或字符串类型。窗口 Y 坐标。

**返回值:**

无。

**备注:**

无。

**示例:**

```
var drawDiv = document.createElement("DIV")
drawDiv.id = "draw";
var lineVml = new IMSPalette(drawDiv,"polyline");
lineVml.appendPoint(x1,y1);
```

## 13.7 dispose

dispose()

**说明:**

清除画图对象。

**参数:**

无。

**返回值:**

无。

**备注:**

无。

**示例:**

```
var drawDiv = document.createElement("DIV")
drawDiv.id = "draw";
var lineVml = new IMSPalette(drawDiv,"polyline");
lineVml.dispose();
```

## 13.8 flash

flash()

**说明:**

闪烁画图对象。

**参数:**

无。

**返回值：**

无。

**备注：**

无。

**示例：**

```
var drawDiv = document.createElement("DIV")
drawDiv.id = "draw";
var lineVml = new IMSPalette(drawDiv,"polyline");
lineVml.flash();
```

## 14. IMSEagleMap

实现鹰眼功能。（注意：必须在 IMSMaps 初始化之后再初始化该类。）

### 14.1 IMSEagleMap

IMSEagleMap(size, zoomAdd, imgSrc, hidePosition, cSpeed)

**说明：**

鹰眼类

**参数：**

size：可选项。数组。指的是鹰眼的宽与高。默认值[150,120]

zooAdd：可选项。数字。指的是鹰眼地图与主图的放缩比例。默认为 3

imgSrc：可选项。数组。指的是显示和隐藏鹰眼开关图片的路径。默认值  
["images/h\_arrow.gif", "images/d\_arrow.gif"]

hidePosition：可选项。数字。指的是鹰眼收缩方向，有四个方向 1 为左上角,2 右上角,3 为右下角,4 为左下角。默认是 3。

cSpeed：可选项。数字。指的是鹰眼动画收缩速度，默认为 30。

**返回值：**

无

**备注：**

必须在 IMSMaps 初始化之后再初始化该类

**示例：**

```
var map = new IMSMaps("zdims",0,null,3IMSMMapConfig);
var eo = new IMSEagleMap();
map.addControl(eo,false);
```

### 14.2 eagleJumpingMap

eagleJumpingMap (x,y,level)

**说明：**

根据输入逻辑坐标和级数来放大、缩小、定位或跳转鹰眼地图（坐标是逻辑坐标）。

**参数:**

x : 必选项。数字。逻辑 X 坐标。

y : 必选项。数字。逻辑 Y 坐标。

level : 必选项。数字。地图级数。

**返回值:**

无

**备注:**

无

**示例:**

```
var map = new IMSMaps("zdims",0,IMSMapConfig);
var eo = new IMSEagleMap();
map.addControl(eo,false);
eo.EagleJumpingMap(114.184225, 30.60855,4);
```

## 14.3 setButtonImage

setButtonImage(img1,img2)

**说明:**

设置鹰眼开关图片。

**参数:**

img1: 必选项。字符串。显示鹰眼图片所在的路径。比如“images/h\_arrow.gif”。

img2: 必选项。字符串。隐藏鹰眼图片所在的路径。比如“images/d\_arrow.gif”。

**返回值:**

无。

**备注:**

无。

**示例:**

```
var map = new IMSMaps("zdims",0,IMSMapConfig);
var eo = new IMSEagleMap();
eo.setButtonImage("img1.gif", "img2.gif");
map.addControl(eo,false);
```

## 14.4 setBorderColor

setBorderColor(color)

**说明:**

设置鹰眼外框颜色。

**参数:**

color: 必选项。字符串。边框的颜色（如“red”,”#f06”）。

**返回值:**

无

**备注:**

无。

示例:

```
var map = new IMSMaps("zdims",0,IMSMapConfig);
var eo = new IMSEagleMap();
eo.setBorderColor ("red");
map.addControl(eo,false);
```

## 14.5 setRectBorderColor

setRectBorderColor (color)

说明:

设置鹰眼中心框颜色。

参数:

color: 必选项。字符串。边框的颜色（如"red", "#f06"）。

返回值:

无

备注:

无。

示例:

```
var map = new IMSMaps("zdims",0,IMSMapConfig);
var eo = new IMSEagleMap();
eo.setRectBorderColor ("red");
map.addControl(eo,false);
```

## 14.6 setRectBackColor

setRectBackColor (color)

说明:

设置鹰眼中心框背景颜色。

参数:

color: 必选项。字符串。边框的颜色（如"red", "#f06"）。

返回值:

无

备注:

无。

示例:

```
var map = new IMSMaps("zdims",0,IMSMapConfig);
var eo = new IMSEagleMap();
eo.setRectBackColor ("red");
map.addControl(eo,false);
```

## 14.7 setViewStatus

setViewStatus(flag)

**说明:**

鹰眼显示开关。

**参数:**

**flag:** 必选项。布尔类型。鹰眼显示开关，为 true 时显示鹰眼，为 false 时隐藏鹰眼。

**返回值:**

无。

**备注:**

无。

**示例:**

```
var map = new IMSMaps("zdims",0,IMSMapConfig);
var eo = new IMSEagleMap();
map.addControl(eo,false);
eo.setViewStatus(true);
```

## 14.8 isOpen

isOpen ()

**说明:**

获得鹰眼是否打开的状态（当返回值为 true 时表示鹰眼是打开，false 为关闭）。

**参数:**

无。

**返回值:**

无。

**备注:**

无。

**示例:**

```
var map = new IMSMaps("zdims",0,IMSMapConfig);
var eo = new IMSEagleMap();
map.addControl(eo,false);
var flag =eo.isOpen();
```

## 14.9 changeViewStatus

changeViewStatus()

**说明:**

改变鹰眼的显示状态，当鹰眼关闭时调用此函数将显示鹰眼，反之将隐藏鹰眼。

**参数:**

无。

**返回值:**

无。

**备注：**

无。

**示例：**

```
var map = new IMSMaps("zdims",0,IMSMapConfig);
var eo = new IMSEagleMap();
map.addControl(eo,false);
eo.changView ();
```

## 15. IMSHotSpots

该类的主要功能是实现为三维地图中的建筑显示名称信息。此类与服务器端页面相关联。应用时，一般在服务器端添加自定义处理程序，用于处理客户端的请求。

使用该类应注意：

- (1) 保证图片和数据库信息相匹配。
- (2) 保证图片文件夹中有一张透明图片。示例中为 `transparence.gif`。

### 15.1 IMSHotSpots

IMSHotSpots (lev,dataRequestPage)

**说明：**

三维热点类。

**参数：**

lev：必选项。数字。开始显示热点的级数。

dataRequestPage：可选项。字符串。服务器处理页面,默认页面为 Maps.ashx。

**返回值：**

无。

**备注：**

无。

**示例：**

```
var hs=new IMSHotSpots(1,"Maps.ashx");//热点对象
map.addControl(hs,false);
```

### 15.2 getHsInfo

getHsInfo (rowNo,lineNo,level)

**说明：**

根据行列号和当前级数，向服务器端请求数据。

**参数：**

rowNo：必选项。数字。向 GIS 服务器取图时图片的行号。

lineNo：必选项。数字。向 GIS 服务器取图时图片的列号。

level：必选项。数字。当前级数。

返回值:

无。

备注:

无

示例:

```
var rowNo = 1 ;
var lineNo = 2 ;
var level = 3 ;
if(this.hotSpotsObj!=null&&level>=this.showHsLev-1 && layerNo==0)
{
this.hotSpotsObj.getHsInfo (rowNo,lineNo,level);
}
```

## 16. IMSCarMove

该小车运动类它主要是实现公交换乘、指路服务的路线显示功能。

### 16.1 IMSCarMove

IMSCarMove(color,width)

说明:

小车运动类初始化函数

参数:

color: 可选项。字符串。线路显示的颜色(默认为“red”)。

width : 可选项。字符串。线路显示的宽度(默认为”3px”)。

返回值:

无

备注:

无

示例:

```
var newCar = new IMSCarMove(“red”,”2px”);
```

### 16.2 setLineInfo

setLineInfo(strPos)

说明:

获取线路的坐标信息，为相关坐标数组赋值

参数:

strPos: 必选项。字符串。表示线路上的所有坐标集，格式为 (x0,y0,x1,y1,x2,y2,……)。

其中 xi,yi 均为逻辑坐标

返回值:

无

**备注:**

无

**示例:**

```
var str = "114.39540,30.51896,114.39100,30.50976";  
var newCar = new IMSCarMove("red",3);  
newCar.setLineInfo(str);
```

## 16.3 drawLine

drawLine()

**说明:**

显示道路曲线。

**参数:**

无。

**返回值:**

无。

**备注:**

调用此函数前必须先调用 setLinInfo 函数来取得道路坐标。

**示例:**

```
var str = "114.39540,30.51896,114.39100,30.50976";  
var newCar = new IMSCarMove("red",3);  
newCar.setLineInfo(str);  
newCar.drawLine();
```

## 16.4 openNormalNodeDiv

openNormalNodeDiv ()

**说明:**

显示道路普通站点信息。

**参数:**

无。

**返回值:**

无。

**备注:**

调用此函数前必须先给 normalNodeInfoArray 赋值。

**示例:**

```
var str = "114.39540,30.51896,114.39100,30.50976";  
var newCar = new IMSCarMove("red",3);  
newCar.setLineInfo(str);  
var nodeInfoArray = new Array();  
nodeInfoArray.push(new Node("中国地质大学", 114.39540,30.51896,1,1));  
newCar.normalNodeInfoArray = nodeInfoArray;  
newCar.openNormalNodeDiv ();
```

## 16.5 closeNormalNodeDiv

closeNormalNodeDiv()

**说明:**

关闭道路普通站点信息的显示。

**参数:**

无。

**返回值:**

无。

**备注:**

无

**示例:**

```
var newCar = new IMSCarMove("red",3);  
var nodeInfoArray =new Array() ;  
newCar.closeNormalNodeDiv();
```

## 16.6 openSpecialNodeDiv

openSpecialNodeDiv ()

**说明:**

显示线路的特殊站点信息，如 起点、步行点、换乘点、终点等。

**参数:**

无

**返回值:**

无

**备注:**

无

**示例:**

```
var str = "114.39540,30.51896,114.39100,30.50976";  
var newCar = new IMSCarMove("red",3);  
newCar.setLineInfo(str);  
newCar.openSpecialNodeDiv ();
```

## 16.7 closeSpecialNodeDiv

closeSpecialNodeDiv ()

**说明:**

关闭线路的特殊站点信息。

**参数:**

无。

**返回值:**

无。

**备注:**

无。

**示例:**

```
var newCar = new IMSCarMove("red",3);  
var nodeInfoArray =new Array() ;  
newCar. closeSpecialNodeDiv ();
```

## 16.8 move

move (carMoveMode)

**说明:**

小车移动

**参数:**

carMoveMode : 必选项。数字。 小车移动方式 (1 为小车移动底图不动, 2 为底图移动小车居中不动)

**返回值:**

无

**备注:**

无

**示例:**

```
var str = "114.39540,30.51896,114.39100,30.50976";  
var newCar = new IMSCarMove("red",3);  
newCar. setLineInfo(str);  
newCar.move(1);
```

## 16.9 pause

pause ()

**说明:**

暂停小车移动。

**参数:**

无。

**返回值:**

无。

**备注:**

无。

**示例:**

```
var str = "114.39540,30.51896,114.39100,30.50976";  
var newCar = new IMSCarMove("red",3);  
newCar. setLineInfo(str);  
newCar.move(1);  
newCar.pause();
```

## 16.10 stop

stop ()

**说明:**

停止小车移动。

**参数:**

无。

**返回值:**

无。

**备注:**

无。

**示例:**

```
var str = "114.39540,30.51896,114.39100,30.50976";
var newCar = new IMSCarMove("red",3);
newCar.setLineInfo(str);
newCar.move(1);
newCar.stop();
```

## 16.11 clearVar

clearVar ()

**说明:**

清除路线与小车。

**参数:**

无。

**返回值:**

无。

**备注:**

无。

**示例:**

```
var str = "114.39540,30.51896,114.39100,30.50976";
var newCar = new IMSCarMove("red",3);
newCar.setLineInfo(str);
newCar.move(1);
newCar.stop();
newCar.ClearVar();
```

## 17. RtMenu

该右键菜单类主要包括了地图上点击右键时触发的操作。在地图显示页面引用。

## 17.1 RtMenu

RtMenu(parentObj,iconList,menuName,commandList,menuId,iHeight,iWidth)

**说明:**

右键类

**参数:**

parentObj: 必选项。要响应事件的对象，比如某个 DIV 层。

iconList: 必选项。数组。指保存右键菜单中各菜单项图标地址的数组。

menuName: 必选项。数组。指保存右键菜单中各菜单项名称的数组。

commandList: 必选项。数组。指的是保存右键菜单项响应事件名称的数组。另当存在子菜单时，相应的响应事件名称为子菜单各项组成的数组。

menuId: 可选项。字符串。右键菜单的 ID。默认值为”menuTable”。

iHeight: 可选项。数字。右键菜单进行显示时的高度。

iWidth: 可选项。数字。右键菜单进行显示时的宽度。

**返回值:**

无

**备注:**

无

**示例:**

一级菜单示例:

```
var iconList = new Array("images/1.gif","images/2.gif","images/4.gif",
"images/3.gif","images/5.gif","", "images/7.gif","images/6.gif");
var menuName = new Array("放 大","缩 小","移 动","复 位","鹰 眼","hr","显示信息点",
"隐藏信息点");
var commandList = new Array("map.setCurOper(IMSOperType.ZoomIn)","
map.setCurOper(IMSOperType.ZoomOut)","
map.setCurOper(IMSOperType.Drag)","
map.setCurOper(IMSOperType.Restore)","EagleDisplay()
","hr","hiddenPnt(false)","hiddenPnt(true)");
var o = new RtMenu($("#zdims"),iconList,menuName,commandList);
o.ShowMenu();
```

二级菜单示例 :

```
var iconList = new Array("images/zin.gif","images/zout.gif","
images/mov.gif","images/resetmap.gif","",
"images/clearall.gif","images/mark.gif"); // icon 数组
var menuName = new Array("子菜单","缩 小","移 动","全 图","hr","添加 vip
点","删除 vip 点");
var childMenu = new Array() ;
var tempCommand1 = new Array("setOpeMode('ZoomIn','Rect')","
map.setCurOper(alert("子菜单一"),"alert("子菜单二"),"alert("子
菜单三"),"alert("子菜单四)");
var menuName1 = new Array("子菜单一","子菜单二","子菜单三","子菜单四");
childMenu.push(menuName1);
```

```
childMenu.push(iconList) ;
childMenu.push(tempCommand1) ;
var commandList = new Array(childMenu,"setOpeMode('ZoomOut','Rect')",
    setOpeMode('Pan','Pan')",childMenu,"hr","","");
var o = new RtMenu$("zdims"),iconList,menuName,commandList);
o.ShowMenu();
```

## 17.2 ShowMenu

ShowMenu()

**说明:**

显示菜单

**参数:**

无

**返回值:**

无

**备注:**

无

**示例:**

无

## 18. 全局方法

### 18.1 String.prototype.trim

**说明:**

对 Array 对象方法的扩展。去除字符串前后的空白字符

**参数:**

无

**返回值:**

返回去除了 String 对象中前后空格的字符串

**备注:**

无

**示例:**

```
var s = " hello  ";
ss = s.trim(); // 返回“hello”
```

### 18.2 Array.prototype.max

**说明:**

对 JavaScript 中 Array 对象的扩展，用于获取数组中值最大的元素。

**参数:**

无

**返回值:**

返回数组中值最大的元素

**备注:**

无

**示例:**

```
var a = new Array(0,1,2,3,4);  
var l = a.max(); //l 为 4
```

## 18.3 Array.prototype.min

**说明:**

对 JavaScript 中 Array 对象的扩展，用于获取数组中值最小的元素。

**参数:**

返回数组中值最小的元素

**返回值:**

无

**备注:**

无

**示例:**

```
var a = new Array(0,1,2,3,4);  
var l = a.min(); //l 为 0
```

## 18.4 StopBubble

StopBubble(e)

**说明:**

停止事件冒泡，中断事件的传播。

**参数:**

e: 必选项。为 event 对象。

**返回值:**

无

**备注:**

无

**示例:**

无

## 18.5 StopDefault

StopDefault(e)

**说明:**

阻止默认浏览器动作，该方法将通知 Web 浏览器不要执行与事件关联的默认动作（如果存在这样的动作）。例如，如果 type 属性是 "submit"，在事件传播的任意阶段可以调用任意的事件句柄，通过调用该方法，可以阻止提交表单。

**参数：**

e: 必选项。event 对象

**返回值：**

无

**备注：**

如果 Event 对象的 cancelable 属性是 false，那么就没有默认动作，或者不能阻止默认动作。无论哪种情况，调用该方法都没有作用。

**示例：**

无

## 18.6 StopDivBubble

StopDivBubble(id)

**说明：**

根据 DIV 层的 ID 号取消其 mousedown、mousemove、mouseup、mouseout、dblclick、click、mousewheel、contextmenu 事件冒泡

**参数：**

Id: 必选项。元素的 ID 号

**返回值：**

无

**备注：**

无

**示例：**

无

## 18.7 GetImgSize

GetImgSize(url)

**说明：**

获取图片的高宽

**参数：**

url: 必选项。字符串。图片的地址。

**返回值：**

存储着图片宽、高值的数组[width , height]，例： [500, 400]。

**备注：**

无

**示例：**

```
var imgSize = GetImgSize("images/p1.gif");
```

## 18.8 CreatDiv

CreatDiv(pElement,id,left,top,position,zIndex,visibility,style,content)

**说明:**

创建 DIV 层

**参数:**

pElement: 必选项。对象类型。div 层依附的父节点

id: 必选项。字符串。div 层 id 号。

position: 可选项。字符串。与父节点的位置关系。“absolute”或则“relative”

left、top: 可选项。字符串。div 层的位置如“10px”。

visibility: 可选项。字符串。是否显示 div 层。Visibility 值为“hidden”时表示 DIV 不可见, “visible”表示 DIV 可见。

style: 可选项。字符串。DIV 层其他样式。

content: div 层嵌入的 innerHTML 字符串

**返回值:**

返回 Div 层对象

**备注:**

用户可以根据自己的需求利用该方法创建出相应的 DIV 层。

**示例:**

```
var div= CreatDiv(divObj,"pntDiv",0,0,"absolute",1,"visible","margin: 0;padding: 0;");
```

## 18.9 GetAbsolutePos

GetAbsolutePos(obj)

**说明:**

获取元素 obj 的绝对位置。

**参数:**

obj: 必选项。需要获取位置的对象。

**返回值:**

存储 obj 绝对位置的数组[left,top], 例如[100,120]

**备注:**

该位置是对象相对于浏览器可视窗口左上角 (0, 0) 的窗口位置。

无

**示例:**

无

## 18.10 AJAXRequest

AJAXRequest(url,isAjax,method,data,callback,charSet)

**说明:**

AJAX 通讯请求方式。

**参数:**

**isAjax:** 必选项。Bool 型。是否为 Ajax 异步请求。True: 异步请求; false: 同步请求;

**url:** 必选项。字符串。处理请求的服务器端页面。

**method:** 必选项。字符串。发送方式。可取值“POST”、“GET”等

**data:** 可选项。字符串。发送的参数名与对应值。

**callback:** 可选项。函数方法。指定返回接收数据的回调函数。

**charSet:** 可选项。字符串。传输编码方式一般有: utf-8、gb2312 等

**返回值:**

无

**备注:**

无

**示例:**

```
var data = "_method=QueryByGraph";
```

```
var url= "Maps.ashx? ";
```

```
AJAXRequest(url,true,"POST",data,cbQuery);
```

其中 cbQuery 为必带一个参数的自定义函数，用于查询返回结果。可模仿 function.js 中的 cbQuery 函数进行自定义改写。