MapX

开发人员指南

MapInfo Corporation

Troy, NY

本文档中的信息如有更改，恕不另行通知，并且本文档中的信息不代表供应商或其代表的任何承诺。未经 MapInfo Corporation（One Global View, Troy, New York 12180-8399) 的书面许可，不得以任何形式、采用任何方式 （无论是电子的还是机械的，包括影印）复制或传播本文档的任何部分。

©1992-2002 MapInfo Corporation。保留所有权利。 MapInfo Help ©1992-2002 MapInfo Corporation。保留所有权利。

MapInfo 和 MapInfo “彩虹”徽标是 MapInfo Corporation 或其附属公司的商标。此处提及的其它产品可能是其各自制造商的商标。

MapInfo、 MapInfo Professional、 MapBasic、 MapXtreme、 MapInfo MapX 和 MapInfo 徽标是 MapInfo Corporation 的注册商标。此处使用的所有其它标记均为其各自所有者所有。

请与 MapInfo Corporation 按以下因特网地址联系：http://www.mapinfo.com

MapInfo 公司总部：

电话：(518) 285-6000

传真：(518) 285-6060

销售信息热线：(800) 327-8627

美国政府销售电话：(800) 619-2333

技术支持热线：(518) 285-7283
技术支持传真：(518) 285-6080

MapInfo 欧洲总部：

电话：+44 (0)1753.848.200

传真：+44 (0)1753.621.140

电子邮件：uk@mapinfo.com

在美国和加拿大提供免费电话支持。有关详细信息，请与您的 MapInfo 销售代表联系。对于国际客户，请使用技术支持传真号码。

德国：

电话：+49 (0)6142-203-400

传真：+49 (0)6142-203-444

电子邮件：germany@mapinfo.com

警告：本软件使用已获专利的 LZW 技术来压缩和 / 或解压缩 .GIF 图像。（Unisys 在美国的专利号为 4,558,302，它在加拿大、法国、德国、意大利、日本和英国也同样获得了专利）。不得出于商业目的而出售或许可使用经过压缩和 / 或解压缩 （以便通过因特网或其他联机通信方式进行传输）的 .GIF 图像。因特网服务提供商 (ISP) 或商业广告也不得使用这种 .GIF 图像，但使用方与 Unisys 事先已达成书面许可协议的情况除外。有关许可方面的信息，请致函以下地址：Unisys Corporation Welch Licensing Department C1SW19 Township Line & Union Meeting Roads P.O. Box 500 Blue Bell PA 19424 Fax: 215-986-3090

Adobe Acrobat ® 是 Adobe Systems Incorporated 在美国的注册商标。

ECW by ER mapper © 1998-2002。

LEAD Technologies, Inc. © 1996。

VMGrid by Northwood Geoscience, Ltd. © 1995-2002。

部分数据是 Roadnet Technologies, Inc., a United Parcel Service Company 拥有所有权的信息，版权所有 1993. Roadnet Technologies, Inc. 部分软件是从 *Standard C Library* 导出的，版权所有 1992, by P.J. Plauger，由 Prentice-Hall 发布并且许可使用。

本文档凝聚了在 MapInfo Corporation 工作的几乎所有员工的心血。它是由 Max Morton、编辑 Colleen Cox 及编辑 Juliette Funiciello 专门撰写的。文档部的这些成员十分感谢 MapInfo 质量保证部门的大力协助，并且向参与此项目的工程团队的所有成员致以谢意。

MapInfo 欢迎您提出宝贵的意见和建议。

MapInfo MapX
2002

目 录

欢迎使用 ... 27
MapInfo MapX 的新增功能 ... 27
关于 MapInfo ... 31

第 1 章 : MapX 简介 ... 33
地图绘制简介 ... 34
使 MapX 适合您工作 ... 34
主要功能概述 ... 34
学习 MapX ... 35

第 2 章 : MapX 基础 ... 37
Map 对象 ... 38
属性页 ... 38
图层 ... 40
GeoSet ... 41
Dataset ... 42
注释 ... 42
MapX 中的可创建对象 ... 43

第 3 章 : 地图绘制概念 ... 47
组织您的数据和地图：表概述 ... 48
什么是 GeoSet? ... 48
地图图元 ... 49
将数据放置于地图上 ... 49
MapX 的强大之处 ... 50

第 4 章 : 设计阶段 MapX 速成教学 ... 51
何为 MapX? ... 52
安装 MapX 之前的注意事项 ... 53
安装 MapX ... 53
添加地图控件 ... 55
从较早的 MapX 版本升级 Visual Basic 应用程序 ... 55
从初期的 MapX 版本升级 C++ 应用程序 ... 56
Visual Basic 速成教学 ... 56
Visual C++ 速成教学 ... 61

Delphi 速成教学 61
MapX 文档集 61
后续内容 63

第 5 章 : 在图层中绘制地图 65
作为图层的地图 66
Layer 集合：地图的构建模块 66
Layer 集合的某些属性 68
Layer 集合的某些方法 70
使用 LayerInfo 对象创建图层 72
Layer 对象 74
图层顺序 75
检查图层 75
检查图元类型 76
缩放图层 76
生成用于图层的标签 78
注释 80
光栅图像 80
动画图层 81
绘制图层 82

第 6 章 : 将数据加入地图 85
什么是数据绑定? 86
将数据加入地图的作用 86
如何将数据加入地图 87
使用 Dataset 对象 87
使用 Datasets 集合 87
使用 Datasets.Add 方法 88
使用 Fields 集合 91
使用 Fields.Add 方法 92
将数据显示为点图层 (BindLayer) 92
使新的点图层成为永久图层 95
数据绑定如何使用 GeoDictionary 96
不同类型的数据源 97
XML 支持 99

第 7 章 : 图元和选择集合 105

什么是地图图元? 106
什么是 Features 集合? 106
什么是 Selection 集合? 106
使用 Features 集合. 106
了解 Selection 集合 108
图元编辑 109

第 8 章 : 主题地图绘制和分析 111
什么是主题地图绘制? 112
规划主题地图 112
主题地图绘制的类型 114
处理主题地图 122
自定义主题图例 123
标签主题 123

第 9 章 : 查找地图上的图元 125
使用 Find 对象. 126
使用 FindFeature 对象. 126

第 10 章 : 使用 Geodictionary 和 Geoset Manager 129
使用 GeodictionaryManager 130
GeoDictionary Manager 的用户界面 131
使用 Geoset Manager. 136

第 11 章 : 使用逐层细化图层 147
什么是逐层细化图层? 148
应当了解的术语和概念 149
如何开发逐层细化应用程序 150
准备逐层细化图层 150
创建逐层细化工具 152
重置逐层细化图层 154
逐层细化图层的限制和要求 154
详细信息 154

第 12 章 : 访问 DBMS 中的数据 157
访问远程空间数据 158
通过 LayerInfo 对象使用 Layers.Add 方法 158

通过 .tab 文件访问远程表 159
通过 X/Y 列对 DBMS 数据进行地图绘制 159
Oracle 支持 159
DBMS LayerInfo 参数 162
DBMS 连接字符串格式 162
MapX DBMS 图层查询 164
访问属性数据 165
性能问题 166
使用缓存 166
MapInfo 地图目录. 168
每个记录的样式 170
使 DBMS 表成为可以进行地图绘制的表 172
故障排除 175

第 13 章 : 使用坐标系统. 177
坐标系统的基本概念 178
获取坐标系统对象 178
查询 CoordSys 对象的属性 178
在不同 CoordSys 中显示地图 178
在不同 CoordSys 中指定 X-Y 坐标 179
显示 “选择投影”对话框. 180
概要介绍坐标系统所使用的参数 181
使用来自 MAPINFOW.PRJ 的设置 183
应用仿射转换 185
定义定制数据 185
数据转换 187
更多信息 187

第 14 章 : 使用 Visual C++ 189
理解示例应用程序 190
从早期的 MapX 版本升级 C++ 应用程序 190
访问 C++ 中的 MapX 属性和方法 191
在您的项目中包括 MapX.cpp 192
使用 C++ 创建 MapX 控件 192
使用 C++ 创建菜单项 194
使用 C++ 处理 MapX 事件 195
使用定制工具 （C++ 示例） 197

使用 C++ 进行数据绑定 198
使用 C++ 添加快捷菜单 200
使用 C++ 的内置帮助程序对话框 201
使用 C++ 处理 MapX 异常错误 202
在 C++ 对话框中创建地图 203

第 15 章 : MapX 工具 207
标准工具概述 208
对象编辑工具 209
节点选择和编辑 209
创建定制工具 212
创建多边形绘制工具 (Polytools) 214

第 16 章 : 导出地图 217
导出地图的方法 218
ExportSelection 属性 219
打印地图 219

第 17 章 : 分发您的 MapX 应用程序 221
MapX 客户安装 222
安装 MapX 223
空间服务器访问安装支持 226
安装光栅格式处理程序 227
安装地图和 Geoset 229
向 Windows 注册表添加注册表项 229
传入 MapX 许可证字符串中 230
重新通过 MrSID 分发您的 MapX 应用程序 230
重新通过 ECW 分发您的 MapX 应用程序 231

第 18 章 : MapX 对象模型 233
Affine Transform 对象 234
AffineTransform.A, B, C, D, E, F 属性 234
AffineTransform.Set 方法 235
AffineTransform.Units 方法 235
AllFeaturesConstraint 对象 236
AllFeaturesConstraint.Disable 方法 236
AllFeaturesConstraint.Enable 方法 237
AllFeaturesConstraint.Enabled 属性 237

Annotation 对象和 Annotations 集合 238
Annotation.Graphic 属性 238
Annotation.Type 属性 239
Annotations.ActiveAnnotation 方法 239
Annotations.AddSymbol 方法 239
Annotations.AddText 方法 239
Annotations.Count 属性 240
Annotations.Editable 属性 240
Annotations.Item 属性 240
Annotations.Remove 方法 241
Annotations.RemoveAll 方法 241
BindLayer 对象 242
BindLayer.CoordSys 属性 242
BindLayer.Filespec 属性 242
BindLayer.KeyLength 属性 243
BindLayer.LayerName 属性 243
BindLayer.LayerType 属性 243
BindLayer.OverwriteFile 属性 244
BindLayer.RefColumn1 属性 244
BindLayer.RefColumn2 属性 245
BindLayer.ReferenceLayer 属性 245
BindLayer.ReferenceLayerField 属性 245
BitmapSymbol 对象 和 BitmapSymbols 集合 247
BitmapSymbol.Name 属性 247
BitmapSymbols.Count 属性 247
BitmapSymbols.Item 属性 248
BitmapSymbols.Refresh 方法 248
BitmapSymbols.Unload 方法 248
BoundsConstraint 对象 249
BoundsConstraint.Bounds 属性 249
BoundsConstraint.Disabel 方法 250
BoundsConstraint.Enable 方法 250
BoundsConstraint.Enabled 属性 251
CoordSys 对象 252
CoordSys.AffineTransform 属性 252
CoordSys.Azimuth 属性 253
CoordSys.Bounds 属性 253
CoordSys.Clone 方法 253
CoordSys.Datum 属性 253
CoordSys.FalseEasting 和 CoordSys.FalseNorthing 属性 253
CoordSys.OriginLatitude、 OriginLongitude 属性 254
CoordSys.PickCoordSys 方法 254
CoordSys.Range 属性 254
CoordSys.ScaleFactor 属性 254
CoordSys.Set 方法 255

CoordSys.StandardParallelOne、 StandardParallelTwo 属性 256
CoordSys.Type 属性 256
CoordSys.Units 属性 256
Dataset 对象和 Datasets 集合 257
Dataset.AddField 方法 258
Datasets.BuildSourceRows 属性 259
Dataset.Fields 属性 259
Dataset.GeoField 属性 259
Dataset.Layer 属性 259
Dataset.Name 属性 259
Dataset.Refresh 方法 260
Dataset.RowValues 属性 260
Dataset.RowCount 属性 260
Dataset.ReadOnly 属性 261
Datasets.Restore 方法 261
Dataset.SecondaryGeoField 属性 262
Dataset.SourceRows 属性 262
Dataset.Themes 属性 262
Dataset.Type 属性 262
Dataset.Value 方法 262
Datasets.Add 方法 263
Datasets.Contains 方法 265
Datasets.Count 属性 265
Datasets.Item 属性 265
Datasets.Remove 方法 266
Datasets.RemoveAll 方法 266
Datum 对象 267
Datum.Eccentricity 属性 267
Datum.Ellipsoid 属性 267
Datum.Flattening 属性 267
Datum.PrimeMeridian 属性 268
Datum.RotateX、 RotateY、 RotateZ 属性 268
Datum.ScaleAdjust 属性 268
Datum.SemiMajorAxis, SemiMinorAxis 属性 268
Datum.Set 方法 269
Datum.SetFromList 方法 269
Datum.ShiftX、 ShiftY、 ShiftZ 属性 270
Feature 对象和 Features 集合 271
Feature.Area 属性 272
Feature.Attach 方法 272
Feature.Clone 方法 273
Feature.Bounds 属性 273
Feature.Caption 属性 273
Feature.CenterX 属性 , CenterY 属性 274
Feature.FeatureID 属性 274

Feature.FeatureKey 属性 274
Feature.KeyValue 属性 275
Feature.HasMultipoint 属性 275
Feature.HasPolyline 属性 276
Feature.HasRegion 属性 276
Feature.LabelPoint 属性 276
Feature.Layer 属性 276
Feature.Length 属性 276
Feature.Multipoint 属性. 277
Feature.Name 属性 277
Feature.Nodes 属性 277
Feature.Offset 方法 278
Feature.Parts 属性 278
Feature.Perimeter 属性 278
Feature.Point 属性 279
Feature.Polyline 属性 279
Feature.Region 属性 279
Feature.Smooth 属性 279
Feature.Style 属性 280
Feature.Type 属性 280
Feature.Update 方法 280
Features.Add 方法 281
Features.AddByID 方法 281
Features.Bounds 属性 281
Features.Clone 方法. 282
Features.Common 方法 282
Features.Count 属性 282
Features.Item 属性. 283
Features.Remove 方法. 283
Features.RemoveByID 方法 283
Features.Replace 方法 283
FeaturesConstraint 对象 285
FeaturesConstraint.Disable 方法 285
FeaturesConstraint.Enabled 属性 286
FeaturesConstraint.Features 属性 286
FeaturesConstraint.Enable 方法 286
FeatureFactory 对象 288
FeatureFactory.BufferFeatures 方法 288
FeatureFactory.CombineFeatures 方法 289
FeatureFactory.CreateArc 方法 290
FeatureFactory.CreateCircularRegion 方法 293
FeatureFactory.CreateCollectionFeature 方法 294
FeatureFactory.CreateEllipticalRegion 方法. 294
FeatureFactory.CreateLine 方法 295
FeatureFactory.CreateMultipoint 方法 295
FeatureFactory.CreateRegion 方法 296

FeatureFactory.CreateSymbol 方法 296
FeatureFactory.CreateText 方法 296
FeatureFactory.EraseFeature 方法 297
FeatureFactory.IntersectFeatures 方法 299
FeatureFactory.IntersectionPoints 方法 300
FeatureFactory.IntersectionTest 方法 301
Field 对象和 Fields 集合 302
Field.AggregationFunction 属性 303
Field.Decimals 属性 303
Field.Indexed 属性 303
Field.Name 属性 303
Field.Precision 属性 303
Field.Width 属性 303
Field.Type 属性 304
Field.TypeEx 属性 304
Fields.Add 方法 305
Fields.AddDateField 方法 306
Fields.AddFloatField 306
Fields.AddIntegerField 306
Fields.AddLogicalField 方法 307
Fields.AddNumericField 方法 307
Fields.AddSmallIntField 方法 308
Fields.Count 属性 309
Fields.Item 属性 309
Fields.Remove 方法 309
Fields.RemoveAll 方法 310
Find 对象 311
Find.Abbreviations 属性 311
Find.CloseMatchMax 属性 311
Find.ClosestAddr 属性 312
Find.FindDataset 属性 312
Find.FindField 属性 312
Find.SearchEx 方法 312
Find.OtherBoundary 属性 313
Find.RefineDataset 属性 313
Find.RefineField 属性 313
Find.RefineLayer 属性 313
Find.Search 方法 314
FindFeature 对象 315
FindFeature.FindRC 属性 315
FindMatch 对象和 FindMatches 集合 317
FindMatch.Name 属性 317
FindMatch.FeatureID 属性 317
FindMatch.FeatureKey 属性 317
FindMatches.Count 属性 318

FindMatches.Item 属性 318
FindResult 对象 319
FindResult.AddressOutOfRange 属性. 319
FindResult.ExactMatch 属性. 319
FindResult.FindRC 属性 320
FindResult.IntersectionNotFound 属性. 320
FindResult.MatchedFeature 属性 320
FindResult.Matches 属性. 320
FindResult.MultipleMatches 属性. 320
FindResult.RefineRegion 属性 321
FindResult.Substitute 属性 321
Geoset 对象和 Geoset 集合 322
Geoset.Centroid 属性. 322
Geoset.PathName 属性 322
Geoset.UserName 属性 322
Geosets.Count 属性 322
Geosets.Item 属性 323
Graphic 对象 324
Graphic.Caption 属性 324
Graphic.Position 属性 324
Graphic.Style 属性. 324
Graphic.X 属性 325
Graphic.Y 属性 325
IndividualValueCategory 对象和 IndividualValueCategories 集合 326
IndividualValueCategories.AllOthersCategory 属性 326
IndividualValueCategories.Count 属性. 326
IndividualValueCategories.Item 属性 326
IndividualValueCategory.NumItems 属性 327
IndividualValueCategory.Style 属性. 327
IndividualValueCategory.Value 属性 327
Label 对象和 Labels 集合 328
Label.AnchorX 属性 329
Label.AnchorY 属性 329
Label.Angle 属性. 329
Label.Caption 属性 329
Label.Drawn 属性 329
Label.Edited 属性 329
Label.EditedAnchor 属性 330
Label.EditedAngle 属性 330
Label.EditedCaption 属性 330
Label.EditedLineStyle 属性. 330
Label.EditedLineType 属性. 330
Label.EditedMultiSegment 属性 331
Label.EditedOffset 属性 331
Label.EditedPosition 属性 331

Label.EditedTextStyle 属性 331
Label.EditedVisibile 属性 332
Label.FeatureKey 332
Label.Layer 属性 332
Label.LineType 属性 332
Label.MultiSegment 属性 332
Label.Offset 属性 333
Label.Position 属性 333
Label.Selected 属性 333
Label.Style 属性 333
Labels.Count 属性 333
Labels.Item 属性 334
LabelProperties 对象 335
LabelProperties.DataField 属性 335
LabelProperties.Dataset 属性 336
LabelProperties.Duplicate 属性 336
LabelProperties.LabelAlong 属性 336
LabelProperties.LabelMax 属性 336
LabelProperties.LabelPartialObjects 属性 337
LabelProperties.LabelZoom 属性 337
LabelProperties.LabelZoomMax 属性 337
LabelProperties.LabelZoomMin 属性 337
LabelProperties.LineType 属性 338
LabelProperties.Offset 属性 338
LabelProperties.Overlap 属性 338
LabelProperties.Parallel 属性 338
LabelProperties.PartialSegments 属性 339
LabelProperties.Position 属性 339
LabelProperties.Style 属性 339
LabelProperties.Visible 属性 339
Layer 对象和 Layers 集合 340
Layer.AddFeature 方法 342
Layer.AllFeatures 方法 343
Layer.AutoLabel 属性 343
Layer.BeginAccess 方法 344
Layer.Bounds 属性 344
Layer.CreateAllFeaturesConstraint 方法 344
Layer.CreateBoundsConstraint 方法 345
Layer.CreateFeaturesConstraint 方法 345
Layer.ClippedBounds 属性 346
Layer.CoordSys 属性 346
Layer.ClearCustomLabels 方法 347
Layer.DeleteFeature 方法 347
Layer.DrawLabelsAfter 属性 348
Layer.DrilldownAddFeatures 方法 348
Layer.DrilldownRemoveFeatures 方法 349

Layer.DrilldownReset 方法 349
Layer.Editable 属性 350
Layer.FeatureIDFromFeatureName 方法 350
Layer.FeatureKeyFromFeatureName 方法 351
Layer.FileSpec 属性 351
Layer.Find 属性 351
Layer.GetDrilldownFeaturesByID 方法 351
Layer.GetFeatureByID 方法 352
Layer.GetFeatureByKey 方法 352
Layer.Invalidate 方法 353
Layer.KeyField 属性 353
Layer.Labels 属性 354
Layer.LabelAtPoint 方法 354
Layer.LabelProperties 属性 354
Layer.Name 属性 355
Layer.NoFeatures 方法 355
Layer.OverrideStyle 属性 355
Layer.Pack 方法 355
Layer.PredominantFeatureType 属性 356
Layer.Refresh 方法 357
Layer.Search 方法 357
Layer.SearchAtPoint 方法 358
Layer.SearchWithinDistance 方法 359
Layer.SearchWithinFeature 方法 359
Layer.SearchWithinRectangle 方法 360
Layer.Selectable 属性 361
Layer.Selection 属性 361
Layer.ShowNodes 属性 361
Layer.ShowCentroids 属性 361
Layer.ShowLineDirection 属性 361
Layer.Style 属性 362
Layer.SupportsPack 方法 362
Layer.Type 属性 362
Layer.UpdateFeature 方法 363
Layer.Visible 属性 364
Layer.ZoomLayer 属性 364
Layer.ZoomMax 属性 364
Layer.ZoomMin 属性 364
Layers.Add 方法 365
Layers.AddGeosetLayers 方法 365
Layers.AddServerLayer 方法 366
Layers.AddUserDrawLayer 方法 367
Layers.AnimationLayer 属性 368
Layers.Bounds 方法 368
Layers.ClippedBounds 属性 368
Layers.ClearSelection 方法 369

Layers.Count 属性 369
Layers.CreateLayer 方法 369
Layer.InsertionLayer 370
Layers.Item 属性 370
Layers.LayersDlg 方法 370
Layers.Move 方法 371
Layers.Remove 方法 372
Layers.RemoveAll 方法. 372
LayerInfo 对象 373
LayerInfo.Type 属性 377
LayerInfo.AddParameter 属性. 377
LayerInfo.Parameter 属性 378
Legend 对象 379
Legend.BodyTextStyle 属性 379
Legend.Compact 属性 380
Legend.CompactTitle 属性 380
Legend.CompactTitleStyle 属性 380
Legend.CurrencyFormat 属性 380
Legend.ExportLegend 方法 380
Legend.Height 属性 381
Legend.Left 属性 381
Legend.LegendDlg 方法 381
Legend.LegendTexts 属性 382
Legend.PaperHeight 属性 382
Legend.PaperWidth 属性. 382
Legend.PrintLegend 方法 383
Legend.ShowCount 属性 383
Legend.ShowEmptyRanges 属性. 383
Legend.SubTitle 属性 383
Legend.SubTitleStyle 属性 384
Legend.Title 属性. 384
Legend.TitleStyle 属性 384
Legend.Top 属性 384
Legend.Visible 属性. 384
Legend.Width 属性 385
LegendText 对象和 LegendTexts 收集 386
LegendTexts.AllOthersText 属性 386
LegendText.Text 属性 386
LegendText.Visible 属性 387
LegendText.AutoGenerate 属性 387
LegendTexts.Count 属性 387
LegendTexts.Item 属性 387
Map 对象 388
Map.AboutBox 方法 390
Map.Annotations 属性 390

Map.AreaUnit 属性 390
Map.AutoRedraw 属性 390
Map.BackColor 属性 391
Map.Bounds 属性 391
Map.CenterX 属性 391
Map.CenterY 属性 391
Map.ClipLine 方法 392
Map.CliplineV 方法 393
Map.ConvertCoord 方法 393
Map.ConvertCoordP 方法 394
Map.ConvertCoordV 方法 394
Map.CreateCustomTool 方法 395
Map.CurrentTool 属性 396
Map.Dataset 属性 396
Map.DatasetGeoField 属性 397
Map.Datasets 属性 397
Map.DatasetTheme 属性 397
Map.DefaultConversionResolution 属性 398
Map.DefaultStyle 属性 399
Map.DisplayCoordSys 属性 399
Map.Distance 方法 399
Map.DynamicSelectionSupport 属性 400
Map.EditableLabels 属性 400
Map.ExportMap 方法 401
Map.ExportSelection 属性 402
Map.FeatureEditMode 属性 402
Map.FeatureFactory 属性 402
Map.GeoDictionary 属性 402
Map.GeoSet 属性 403
Map.GeoSets 属性 404
Map.GeoSetWidth 属性 404
Map.hWnd 属性 404
Map.InfotipSupport 属性 404
Map.InfotipPopupDelay 属性 405
Map.IsPointVisible 方法 405
Map.Layers 属性 405
Map.MapUnit 属性 405
Map.MapPaperHeight 属性 406
Map.MapPaperWidth 属性 406
Map.MapScreenHeight 属性 406
Map.MapScreenWidth 属性 406
Map.MatchNumericFields 属性 406
Map.MatchThreshold 属性 406
Map.MaxSearchTime 属性 407
Map.MilitraryGridReferenceFromPoint 方法 407
Map.MilitaryGridReferenceToPointV 方法 408

Map.MilitaryGridReferenceToPoint 方法 409
Map.MouseIcon 属性 410
Map.MousePointer 属性 411
Map.MouseWheelSupport 属性. 411
Map.NumericCoordSys 属性. 411
Map.Pan 方法. 411
Map.PanAnimationLayer 属性 412
Map.PaperUnit 属性 412
Map.PreferCompactLegends 属性. 412
Map.PrintMap 方法 412
Map.PropertyPage 方法 413
Map.RedrawInterval 属性 413
Map.Refresh 方法 414
Map.ReuseEquivalentOnRestore 属性. 414
Map.Rotation 属性. 415
Map.SaveMapAsGeoset 方法 415
Map.SearchPath 属性. 416
Map.SelectionStyle 属性 416
Map.SetSize 方法. 416
Map.SnapTolerance 属性 417
Map.SnapToNodeSupport 属性. 417
Map.Title 属性. 418
Map.TitleText 属性 418
Map.Version 属性 418
Map.WaitCursorEnabled 属性. 418
Map.Zoom 属性. 419
Map.ZoomTo 方法. 419
MultivarCategory 对象和 MultivarCategories 集合 420
MultivarCategories.Count 属性. 420
MultivarCategories.Item 属性 420
MultivarCategory.Style 属性. 420
NotesQueryInfo 对象. 421
NotesQueryInfo.BeginDate 属性 （NotesQueryInfo 对象） 421
NotesQueryInfo.Database 属性 （NotesQueryInfo 对象） 421
NotesQueryInfo.DefaultNumericValue 属性 （NotesQueryInfo 对象）. 421
NotesQueryInfo.DefaultStringValue 属性 （NotesQueryInfo 对象） 422
NotesQueryInfo.EndDate 属性 （NotesQueryInfo 对象）. 422
NotesQueryInfo.FullTextSearch 属性 （NotesQueryInfo 对象） 422
NotesQueryInfo.MaxNumDocs 属性 （NotesQueryInfo 对象）. 422
NotesQueryInfo.Query 属性 （NotesQueryInfo 对象）. 422
NotesQueryInfo.Server 属性 （NotesQueryInfo 对象）. 423
NotesViewInfo 对象 424
NotesViewInfo.Database 属性 （NotesViewInfo 对象）. 424
NotesViewInfo.Server （NotesViewInfo 对象） 424
NotesViewInfo.View （NotesViewInfo 对象） 424

OCIQueryInfo 对象 425
OCIQueryInfo.ConnectString 属性 （OCIQueryInfo 对象） 425
OCIQueryInfo.Query 属性 （OCIQueryInfo 对象） 425
ODBCQueryInfo 对象 426
ODBCQueryInfo.ConnectString 属性 （ODBCQueryInfo 对象） 426
ODBCQueryInfo.DataSource 属性 （ODBCQueryInfo 对象） 426
ODBCQueryInfo.SqlQuery 属性 （ODBCQueryInfo 对象） 426
Parts 集合 427
Parts.Add 方法 427
Parts.Count 属性 427
Parts.Item 属性 427
Parts.Remove 方法 428
Parts.RemoveAll 方法 428
Point 对象和 Points 集合 429
Point.Offset 方法 429
Point.Set 方法 430
Point.X 属性 430
Point.Y 属性 430
Points.Add 方法 430
Points.Count 属性 431
Points.Item 属性 431
Points.GetXY 方法 431
Points.GetXYV 方法 432
Points.X 方法 432
Points.Y 方法 433
Points.Remove 方法 433
Points.RemoveAll 方法 433
RangeCategory 对象和 RangeCategories 集合 434
RangeCategories.AllOthersCategory 属性 434
RangeCategories.Count 属性 434
RangeCategories.Item 属性 434
RangeCategory.Max 属性 435
RangeCategory.Min 属性 435
RangeCategory.NumItems 属性 435
RangeCategory.Style 属性 436
Rectangle 对象 437
Rectangle.Height 属性 437
Rectangle.Offset 方法 437
Rectangle.Set 方法 438
Rectangle.Width 属性 438
Rectangle.XMax 属性 438
Rectangle.XMin 属性 438
Rectangle.YMax 属性 438
Rectangle.YMin 属性 439
ResolveObject 对象和 ResolveObjects 集合 440

ResolveObject.TableName 属性 440
ResolveObject.SourceMatch 属性 440
ResolveObject.TableMatch 属性 440
ResolveObjects.Count 属性 440
ResolveObjects.Item 属性 441
RowValue 对象和 RowValues 集合 442
RowValue.ReadOnly 属性 442
RowValue.Dataset 属性 442
RowValue.Field 属性 443
RowValue.Value 属性 443
RowValues.Count 属性 443
RowValues.ReadOnly 属性 443
RowValues.Item 属性 （ 443
RowValues.Remove 方法 444
RowValues.Add 方法 444
RowValues.RemoveAll 方法 444
RowValues.Clone 方法 445
Selection 集合 446
Selection.Count 属性 446
Selection.Item 属性 447
Selection.Add 方法 447
Selection.ClearSelection 方法 447
Selection.Clone 方法 448
Selection.Common 方法 448
Selection.Remove 方法 448
Selection.Replace 方法 449
Selection.SelectAll 方法 449
Selection.SelectByID 方法 449
Selection.SelectByPoint 方法 450
Selection.SelectByRadius 方法 451
Selection.SelectByRectangle 方法 451
Selection.SelectByRegion 方法 452
SourceRow 对象和 SourceRows 集合 454
SourceRow.Row 属性 454
SourceRows.Count 属性 454
SourceRows.Item 属性 454
State 对象 456
State.ReadFromFile 方法 458
State.Reset 方法 458
State.Restore 方法 459
State.Save 方法 460
State.Stream 属性 461
State.WriteToFile 方法 462
Style 对象 463
Style.Clone 方法 464

Style.DrawLineSample 方法 465
Style.DrawRegionSample 方法 465
Style.DrawSymbolSample 方法 465
Style.DrawTextSample 方法 466
Style.ExportLineSample 方法 466
Style.ExportRegionSample 方法 467
Style.ExportSymbolSample 方法 467
Style.ExportTextSample 方法 468
Style.LineColor 属性 469
Style.LineInterleaved 属性 469
Style.LineStyle 属性 469
Style.LineStyleCount 属性 470
Style.LineSupportsInterleave 属性 471
Style.LineWidth 属性 471
Style.LineWidthUnit 属性 471
Style.PickLine 方法 471
Style.PickRegion 方法 472
Style.PickSymbol 方法 472
Style.PickText 方法 472
Style.RegionBackColor 属性 472
Style.RegionBorderColor 属性 473
Style.RegionBorderStyle 属性 473
Style.RegionBorderWidth 属性 473
Style.RegionBorderWidthUnit 属性 473
Style.RegionColor 属性 473
Style.RegionPattern 属性 473
Style.RegionTransparent 属性 474
Style.SupportsBitmapSymbols 属性 474
Style.SymbolBitmapColor 属性 474
Style.SymbolBitmapName 属性 475
Style.SymbolBitmapOverrideColor 属性 475
Style.SymbolBitmapSize 属性 475
Style.SymbolBitmapTransparent 属性 476
Style.SymbolCharacter 属性 476
Style.SymbolFont 属性 476
Style.SymbolFontBackColor 属性 477
Style.SymbolFontColor 属性 477
Style.SymbolFontHalo 属性 477
Style.SymbolFontOpaque 属性 477
Style.SymbolFontRotation 属性 477
Style.SymbolFontShadow 属性 478
Style.MinVectorSymbolCharacter 属性 478
Style.MaxVectorSymbolCharacter 属性 478
Style.SymbolVectorColor 属性 478
Style.SymbolVectorSize 属性 478
Style.TextFont 属性 478

Style.TextFontAllCaps 属性 479
Style.TextFontBackColor 属性 479
Style.TextFontColor 属性 479
Style.TextFontDblSpace 属性 479
Style.TextFontHalo 属性 479
Style.TextFontOpaque 属性 479
Style.TextFontShadow 属性 480
Style.TextFontRotation 属性 480
Style.SymbolType 属性 480
Theme 对象和 Themes 集合 481
Theme.AutoRecompute 属性 481
Theme.ComputeTheme 属性 482
Theme.DataMax 属性 482
Theme.DataMin 属性 482
Theme.Fields 属性 483
Theme.Layer 属性 483
Theme.Legend 属性 483
Theme.Name 属性 483
Theme.Properties 属性 484
Theme.ThemeDlg 方法 484
Theme.ThemeProperties 属性 484
Theme.Type 属性 485
Theme.Visible 属性 485
Themes.Add 方法 485
Themes.Count 属性 486
Themes.Item 属性 486
Themes.Remove 方法 487
Themes.RemoveAll 方法 487
ThemeProperties 对象 488
ThemeProperties.AllowEmptyRanges 属性 489
ThemeProperties.DataValue 属性 489
ThemeProperties.DistMethod 属性 489
ThemeProperties.DotSize 属性 490
ThemeProperties.Graduated 属性 490
ThemeProperties.Independent 属性 490
ThemeProperties.IndividualValueCategories 属性 491
ThemeProperties.MultivarCategories 属性 491
ThemeProperties.NumRanges 属性 491
ThemeProperties.RangeCategories 属性 491
ThemeProperties.Size 属性 492
ThemeProperties.SpreadBy 属性 492
ThemeProperties.SymbolStyle 属性 492
ThemeProperties.ValuePerDot 属性 493
ThemeProperties.Width 属性 493
ThemeProperties.PieClockwise 属性 493
ThemeProperties.PieStartAngle 属性 493

ThemeProperties.PieHalfPies 属性 493
ThemeProperties.PieGraduated 属性. 494
ThemeProperties.BarStacked 属性 494
ThemeProperties.BarGraduatedStack 属性. 494
ThemeProperties.BarIndependentScale 属性 494
ThemeProperties.BarWidth 属性. 495
ThemeProperties.BarFramed 属性 495
ThemeProperties.BarFrameStyle 495
ThemeProperties.BorderStyle 属性. 495
ThemeProperties.ShowNegativeValues 属性 496
ThemeProperties.PositiveSymbolStyle 属性. 496
ThemeProperties.NegativeSymbolStyle 属性 496
ThemeProperties.GraduateSizeBy 属性 497
ThemeProperties.DotColor 属性 497
ThemeProperties.RoundRanges 属性 497
ThemeProperties.RoundBy 属性. 497
ThemeProperties.InflectRanges 属性 498
ThemeProperties.InflectionRange 属性. 498
ThemeProperties.InflectionColor 属性 498
ThemeProperties.ColorMethod 属性. 499
ThemeProperties.ApplyAttribute 属性 499
Title 对象 500
Title.Border 属性. 500
Title.Caption 属性 500
Title.Editable 属性. 500
Title.Position 属性. 500
Title.TextStyle 属性. 501
Title.Visible 属性. 501
Title.X 属性 501
Title.Y 属性 501
Variable 对象和 Variables 集合 502
Variable.Name 属性 502
Variable.Value 属性 503
Variables.Count 属性. 503
Variables.Add 方法 503
Variables.Clone 方法. 504
Variables.Item 方法. 504
Variables.Remove 方法. 504
Variables.RemoveAll 方法 505

附录 A: MapX 字段命名规范 507

附录 B: 调试器定制数据集支持 509

附录 C: 术语词汇表 519

附录 D: 使用 Geodictionary Manager 527

附录 E: MapX 事件 535

AddFeatureToolUsed 事件 536
AnnotationAdded 事件 537
AnnotationChanged 事件 537
Data 事件 538
DrawUserLayer 事件 539
LabelChanged 事件 539
MapDraw 事件 540
MapInitialized 事件 540
MapViewChanged 事件 540
MouseWheel 事件 541
PolyToolUsed 事件 542
RequestData 事件 542
ResolveDataBind 事件 543
ResolveDataBindEx 事件 543
SelectionChanged 事件 544
ThemeModifyRequested 事件 544
ToolUsed 事件 545

附录 F: MapX 错误代码 547

附录 G: 常数 557

AggregationFunctionConstants 557
AnnotationChangedTypeConstants 558
AnnotationTypeConstants 558
ApplyAttributeConstants 558
AreaUnitConstants 558
BindLayerTypeConstants 559
ColorConstants 560
ColorSpreadingMethodConstants 561
ConversionConstants 562
CoordSysTypeConstants 563
CursorConstants 564
DataSetTypeConstants 565
DistribMethod Constants 566
DotSizeConstants 566
ExportFormatConstants 566
FeatureEditModeConstants 566
FeatureTypeConstants 566
FieldTypeConstants 567
FillPatternConstants 567
GraduationConstants 567
IntersectionPointConstants 568

IntersectionTestConstants 568
LabelAlongConstants 568
LayerBeginAccessConstants 568
LayerEndAccessConstants 568
LayerInfoTypeConstants 568
LayerPackConstants 569
LayerSrvLayerOptions 569
LayerTypeConstants 569
LineTypeConstants 569
MapDrawConstants 569
MapUnitConstants 570
MouseWheelSupportConstants 570
PaperUnitConstants 570
PenStyleConstants 571
PolyToolFlagConstants 571
PositionConstants 571
ResolveDataBindConstants 572
SearchTypeConstants 572
SearchResultTypeConstants 572
SelectionTypeConstants 572
SpreadByConstants 572
StyleUnitConstants 573
SymbolTypeConstants 573
ThemeTypeConstants 573
ToolConstants 573
ToolFlagConstants 574
ToolTypeConstants 574

附录 H: 创建表达式 575

运算符 575
地理运算符 576
数学运算符 576
比较运算符 578
数字比较 578
字符串比较 578
日期比较 579
逻辑比较 579
逻辑运算符 579
字符串子句 580
日期子句 580
关键字 580
函数 582
Acos( ) 函数 584
Area( ) 函数 585
Atn( ) 函数 587

Buffer( ) 函数 588
Centroid( ) 函数 588
CentroidY( ) 函数 590
Chr$( ) 函数 591
CurDate( ) 函数 592
DeformatNumber$( ) 函数 593
Distance( ) 函数 593
Exp( ) 函数 594
LCase$( ) 函数 601
Left$( ) 函数 602
LTrim$( ) 函数 603
Maximum( ) 函数 604
MBR( ) 函数 604
Mid$( ) 函数 605
Minimum( ) 函数 605
ObjectLen( ) 函数 607
Perimeter( ) 函数 608
Proper$( ) 函数 609
Round( ) 函数 610
RTrim$( ) 函数 611
Sin( ) 函数 612
Sqr( ) 函数 613
StringCompare( ) 函数 614
StringCompareIntl( ) 函数 614
StringToDate( ) 函数 615
Str$( ) 函数 616
Tan( ) 函数 617
Val( ) 函数 618
Weekday( ) 函数 619

附录 I: Geoset 关键字 621

附录 J: 示例 Geoset 623

附录 K: 支持的关键字 627

附录 L: OLE_COLOR 值 631

附录 M: Idispatch 表 633

附录 N: 管理 MapX 数据 653

附录 O: 数据源 655

附录 P: 使用 Geoset Manager 665

索引 675

欢迎使用 ...

MapInfo MapX 开发人员指南！

本指南中的内容仅反映截至出版时可提供的信息，实质上与 “MapX 联机帮助系统”中的内容一致。 不过，对于信息时有更新的情况， MapX 联机帮助可随时反映更新内容，而不必等到 MapInfo MapX 新版真正发布。 请务必访问 www.MapX.com 下载最新版本的 “帮助系统”。

MapInfo 文档编写部门欢迎用户提出任何方面的意见和建议。您可以用电子邮件将意见和建议发送至 **MapXDocs@MapInfo.com**。

文档约定

MapInfo MapX 开发人员指南 中的语法说明所用的部分版式约定与 Visual Basic 文档中的约定类似。 请参考 “Microsoft Visual Basic 程序员指南”或 “Microsoft Visual Basic 语言参考”中的版式约定列表。

您会在方法条目中找到语法部分。 对于本版 MapInfo MapX，我们已对显示语法所遵循的约定进行标准化，如下所示：

[Return Value=]OBJECT.Method (parameter, [optional parameter])

部分	说明
[Return Value=]	属性的返回值 （如果确实返回一个值）将始终显示在语法行的左外侧，括在方括号中，右侧是赋值运算符。
OBJECT	表示方法应用到的对象。
方法	表示所用的方法。
(parameter)	参数需括在圆括号中。
[optional parameter]	可选参数需括在参数圆括号内的方括号中。

MapInfo MapX 的新增功能

这一节概括了 MapX 5.0 版的改进和增强功能。

- **图层压缩：**对图层中的数据进行压缩后将移除已删除的记录并重建索引和图形文件。为实现该功能， Layer 对象中添加了两种方法，分别是：
  - Layer.Pack 方法，第 355 页
  - Layer.SupportsPack 方法，第 362 页
- **将 “图层”另存为 .mdb 数据文件中的 MapInfo 本地表：**创建新表时， MapX 可提供两种保存格式 ó MapInfo 本地格式和 MS Access 格式。此外，您还可以为 MapX 创建的表选择 Access MDB 版本。
  通过指定 LayerInfo.Type 可创建新的永久表 （请参见第 373 页的 “LayerInfo 对象”参数表中的 TableStorageType 和 AccessVersion，该表位于 miLayerInfoTypeNewTable 之下）。
- **改进了用于远程数据库连接的高速缓存的性能：**MapX 的高速缓存能力得以增强，这是通过扩展 LayerInfo 对象的 Cache 参数、添加三个新对象以及将这三个新对象连接到 Layer 对象的三种新方

法来实现的。具体而言，“高速缓存”数据集可用来缓存来自远程数据库数据集内的记录，从而避免增加远程数据库服务器返回 MapX 最近提取数据的回程。以往在 MapX 中，高速缓存对每个图层仅保留一个 MBR 约束。而现今，用户 / 编程人员都有权决定高速缓存的存储内容、存储容量以及移除时间。请参考以下内容了解更多详细信息：

- LayerInfo 对象，第 373 页
- BoundsConstraint 对象，第 249 页
- FeaturesConstraint 对象，第 285 页
- AllFeaturesConstraint 对象，第 236 页
- Layer.CreateBoundsConstraint 方法，第 345 页
- Layer.CreateFeaturesConstraint 方法，第 345 页
- Layer.CreateAllFeaturesConstraint 方法，第 344 页

- **双平射投影：**对于双平射投影，椭圆数据首先通过正形投影绘制到正形球面上。然后，再一次利用正形投影将球面数据绘制到平面上即可完成整个绘图过程。加拿大的爱德华王子岛和新布伦兹维克的地图就是通过这种投影方法绘制的。请参考第 181 页的第 13 章：“概要介绍坐标系统所使用的参数”。
- **使用圆形和矩形选择工具动态选择对象：**在 Map 对象中，您会发现一个新的名为“Map.DynamicSelectionSupport 属性”的属性 （第 400 页）。以往在 MapX 中，如果您拖动所选工具的光标而后释放了鼠标按钮， MapX 将突出显示您圈定的橡皮圈形状内的所有对象。如果开启 “动态选择”模式，拖动鼠标的同时即可选中对象 （即：您不必再释放鼠标按钮，不必专门作出选择即可看到所选对象）。这是一个可选模式，通过将这个布尔型属性设置为 true 或 false 可相应启用或禁用该模式。该模式仅适用于矩形和圆形选择工具。
- **增强了标注功能：**以往在 MapX 中，编程人员可以控制标注，但只能按图层控制，而无法单独控制各个标注。直至现今仍无法单独定制各个标注的外观和内容。而自从添加了 Label 对象和集合、 LabelChanged 事件以及 LabelProperties 对象中的两个新属性后， MapInfo 为 MapX 编程人员扩展了这一功能。

  MapX 编程人员还可以允许用户使用 MapX 选择工具手动编辑标注。 通过 Map.EditableLabels 属性即可控制用户能否借助普通选择工具选择和移动标注。 （请参见 第 400 页的“Map.EditableLabels 属性”）

  新功能如下：

  - **LabelProperties.LabelPartialObjects 属性，第 337 页：**PartialSegment 属性的地位已大不如前 （但出于向后兼容的考虑，仍对其提供支持），而且将被新属性 LabelProperties.LabelPartialObjects 所取代。其操作方法适用于所有的地图对象 （但点除外），而不仅限于多义线。因此，如果要标注二维 GeoObject，但它们的中心不在地图中， MapX 仍将对其做出标注。其操作方法与标注多义线的方法相同。

- **LabelProperties.LabelAlong 属性，第 336 页：**LabelAlong 属性取代了原有的 Parallel 属性。虽然我们仍对 Parallel 属性提供支持，但已将其降格为隐藏属性。以前，由 Parallel 属性来决定是否让线条或多义线上的标注进行旋转，或者绘制这些标注时所使用的角度是否与其所锚定的线段或多义线段的绘制角度相同。现在，如果所要标注的对象为多义线，则使用 LabelAlong 属性就可表明该标注将沿被标注对象的轮廓绘制。LabelAlong 有三种状态，分别用不同的常数来表示。 （请参阅第 568 页的 “LabelAlongConstants”）。对于除多义线之外的其他对象 （例如，点对象和区域对象），如果将属性设置为 miLabelAlongMultiSegment，则该属性将被忽略。对于线条对象，如果将属性设置为 miLabelAlongMultiSegment，就将绘制自动标注，其效果与将属性设置为 miLabelAlongParallel 相同。
- **Label 对象和 Labels 集合，第 328 页：**有一个新对象可通过 “图层”对象 （请参阅第 354 页的 “Layer.Labels 属性”，该属性为新属性）来进行访问，它代表一个标注集。该标注集中既包括当前所绘制的标注，也包括通过用户界面或程序进行过编辑的标注。请参阅 “标注”对象的有关属性。
- **LabelChanged 事件，第 539 页：**只要有任何标注要进行修改、隐藏 （因为标注无法真正被删除）、标为隐藏或取消隐藏，则 MapX 就将触发此事件。MapX 编程人员可通过将 EnableDefault 的值设置为 False 来避免进行此操作。

- **军用方格参考系统 (MGRS) 坐标：**MapX 现在能够使用 “地图”对象的以下方法来与 MGRFS 坐标进行相互转换：
  - Map.MilitraryGridReferenceFromPoint 方法，第 407 页
  - Map.MilitaryGridReferenceToPointV 方法，第 408 页
  - Map.MilitaryGridReferenceToPoint 方法，第 409 页
- **匹配节点：**启用此功能之后，如果鼠标指针当前所在的矩形中有节点，则该功能将匹配距离该矩形中心最近的那个节点，然后绘制十字准线。在所有工具下都可打开此模式。请参阅下面有关新 “地图”对象的属性和方法：
  - Map.SnapToNodeSupport 属性，第 417 页
  - Map.SnapTolerance 属性，第 417 页
- **JPEG2000 支持：**已将此栅格格式添加到 MapX 所支持的栅格格式列表中。
- **Unicode 支持：**MapX 现在已是 Unicode 应用程序。
- **多点对象和集合对象：**MapInfo 为 MapX 5.0 中的两种新 GeoObject 类型 （多点对象和集合对象）添加了支持。多点对象是由许多点组成的单个对象。多点对象对应于表中的单条记录，而且该对象中的所有点都具有相同的符号。集合对象由许多对象组成，而这些对象又组合成单个对象。集合对象中可包含点、多义线和多边形。

  为多点对象和集合对象新添了六种 Feature 对象属性，以支持以下新的 GeoObject 类型：

- Feature.HasMultipoint 属性，第 275 页
- Feature.HasPolyline 属性，第 276 页
- Feature.HasRegion 属性，第 276 页
- Feature.Multipoint 属性，第 277 页
- Feature.Polyline 属性，第 279 页
- Feature.Region 属性，第 279 页

在此版本中， FeatureFactory 对象新添了两种方法：

- FeatureFactory.CreateMultipoint 方法，第 295 页
- FeatureFactory.CreateCollectionFeature 方法，第 294 页

- **支持标记主题：**此功能使得 MapX 开发人员能够向图层数据集中添加主题，以修改该图层的标注显示特点 （例如，标注的文本样式）。 有关详细信息，请参阅第 123 页的 “标签主题”。要了解新罗列的常数值，请参阅第 573 页的 “ThemeTypeConstants”。
- **坐标系更新：**MapX 中新添了五个坐标系 （请参阅第 181 页的 “概要介绍坐标系统所使用的参数”）。新的投影方法包括：
  - 麦卡脱投影 (Mercator Regional)
  - 多圆锥投影 (Polyconic)
  - Cassini - Soldner 投影
  - * 等距方位投影 (Azimuthal Equidistant) （此版本提供所有纬度原点）
  - * 朗伯等积方位投影 (Lambert Azimuthal Equal-Area) （此版本提供所有纬度原点）

  有关新罗列的常数值，请参阅**第 563 页的 “CoordSysTypeConstants”**。

注：*MapX 的以前版本中也提供了等距方位投影和朗伯等积方位投影，但只提供与极点有关的功能。

- **Oracle Call Interface(OCI) 数据源支持：**MapX 可使用 OCI 从 Oracle 数据源中检索数据。您需要使用 **OCIQueryInfo 对象**来指定所要执行的连接字符串和 SQL 字符串。

  有关详细信息，请参阅第 425 页的 “OCIQueryInfo 对象”。
- **优化点和组成部分的性能：**为 “点”集合新添了三种方法之后， MapX 开发人员就可迅速找出任何指定点的 X 轴 和 Y 轴。这三种方法及其在本开发人员指南中的位置分别如下：
  - Points.GetXY 方法，第 431 页
  - Points.X 方法，第 432 页
  - Points.Y 方法，第 433 页
  - Points.GetXYV 方法，第 432 页

- **“状态”对象：**使您能够保存任何项目 （例如对象或变量）的当前状态，而您以后就可将该项目的状态恢复为当前所保存的状态。可以保存和恢复 MapX 对象或变量 （例如，字符串或布尔值）。

  有关详细信息，请参阅第 456 页的 “State 对象”。

- **方格处理程序和栅格处理程序：**MapX 可支持其他的栅格文件格式和方格文件格式。

  其中，栅格格式包括：

  - MrSID （*.sid，由 Lizard Tech 公司开发）
  - ECW （*.ecw，由 ER Mapper 公司开发）
  - Vertical Mapper Continuous Grids （*.grd，由 Northwood Technologies 公司开发）
  - Vertical Mapper Classified Grids （*. grc，由 Northwood Technologies 公司开发）

  而方格格式则包括：

  - USGS DEM (*.dem)
  - GTOPO30 (*.dem)
  - DTED （*.dt0、 *.dt1、 *.dt2）
  - Vertical Mapper Continuous Grids （*.grd，由 Marconie （Northwood 是它的前身）公司开发）

  这些方格处理程序和栅格处理程序可支持您使用更多的格式来显示文件。这些处理程序是对以前版本中已有的处理程序的补充。

- **半透明栅格图像：**半透明图像可被透视。因此，可将栅格图像放置在其他图层的上面，而透过该图像还能看到下面图层中的部分内容。对于已使用 MapInfo Professional 6.5 启用了半透明功能的栅格表， MapXtreme 支持对其进行半透明处理。

关于 MapInfo

MapInfo MapX 是由 MapInfo Corporation 开发的产品。MapInfo 开发并销售种类齐全的地图绘制软件和数据产品。

如果您需要了解其他地图解决方案或数据产品，请通过电话或传真 （号码见后文）与我们联络，或访问我们的网站。

注：MapX 网站 (www.mapx.com) 为开发人员提供了最新的应用程序示例和 MapX 联机帮助系统的各个版本。此外，它还提供了一个公告栏，以便与其他 MapX 开发人员协作。

如果您在使用 MapInfo MapX 时需要帮助，我们的技术支持专家将为您排忧解难。MapX 的技术支持包括：查阅文档；帮助解决错误消息；帮助查找导致错误消息的原因。 （MapX 仅为已购买 MapX 的客户提供电话支持。）您还可安排参加 MapX 的有关培训，或从 MapInfo 的开发人员服务部门获得定制的服务。

MapInfo Corporate 总部的联络信息

- MapX 网站： **http://www.mapx.com**
- MapInfo 网站： **http://www.mapinfo.com**
- 主要电话： **518.285.6000**
- 主要传真： **518.285.6060**
- 销售信息预定热线： **800.327.8627**
- 联邦销售： **800.619.2333**
- 销售传真： **518.285.6070**
- MapX 销售电子邮件： **mapx.sales@mapinfo.com**
- 技术支持热线： **518.285.7283**
- 技术支持传真： **518.285.6080**
- 培训电子邮件： **training@mapinfo.com**
- 培训电话： **800.552.2511**
- 开发人员服务电子邮件： **devservices@mapinfo.com**
- 开发人员服务电话： **518.285.7175**

1 章

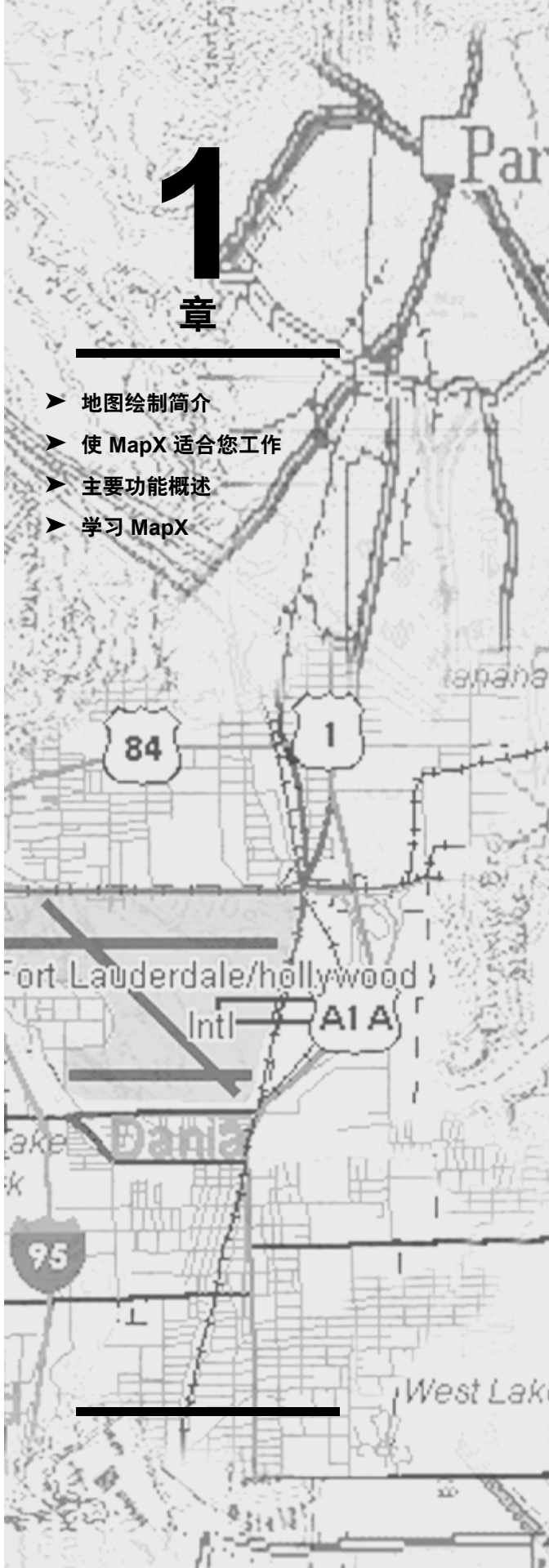
- 地图绘制简介
- 使 MapX 适合您工作
- 主要功能概述
- 学习 MapX

MapX 简介

欢迎使用 MapInfo 系列产品。随着企业地图绘制领域的发展壮大， MapInfo 继续致力于推出为满足用户的桌面和企业地图绘制需要而设计的各种新产品，包括从标志性的产品 MapInfo Professional 直到最为专业化的产品 MapMarker（我们的主要地址匹配产品）。

MapInfo MapX 是地图绘制控件，可以使您轻松地在您的应用程序中添加强大的地图绘制功能。使用地图，您可以通过每一人都很容易理解的格式显示信息。地图比简单的图表和图形更具信息性，并且比电子表格更快和更容易地加以解释。

MapX 基于与其它 MapInfo 产品 （例如 MapInfo Professional）所使用的相同的地图绘制技术。如果您创建或购买了 MapInfo 地图数据 （表）以用于 MapInfo Professional，则可以将同样这些文件用于 MapX。



地图绘制简介

当今可以利用数目庞大的信息，与以前相比远多得多。数据大量存在于电子表格、销售记录和市场宣传文件中。纸张和磁盘存储了大量与客户、店铺、人员、设备和资源有关的信息。

几乎所有这些信息都具有地理成分。估计约 85% 的数据库包含某些类型的地理信息，例如街道地址、城市、州、 ZIP 编码或者具有区号和分机号的电话号码。

MapX 可以帮助您对所有这些信息进行分类，并且使用数据中的地理成分在地图上显示结果。这使您可以简单、迅速地了解大量信息中蕴含的模式和关系，而不必详查您的数据库。

使 MapX 适合您工作

使用 MapX，您可以完全拥有向您的应用程序添加强大的地图绘制功能的能力。您可以将数据显示为点、按主题遮蔽的区域、饼图或条形图等。通过在特定的半径、矩形或特定点内组合和组织数据、执行搜索或选择地图图元，可以充分发挥 MapX 的分析功能。

例如， MapX 可以显示哪一分店距离您的最大客户最近。它可以计算客户和店铺之间的距离，可以向您显示去年哪一客户花费最多，可以按销售量对店铺符号进行颜色编码。所有这一切相结合，使数据可以在地图上直观地显示。

主要功能概述

使用 MapX，您可以分析和直观地表示您的业务数据，创建或编辑地图图元，从地理上显示结果。下面列出了 MapX 的主要功能。

- **主题地图绘制** 使用主题地图绘制直观地表示您的数据。将数据与地图上的每一图元相关联，然后使用颜色编码 （或其它样式）来直观地展现您的数据。使用主题地图绘制，您可以使用六种不同样式的任何一种 （颜色范围、点密度、单独值、分级符号、饼图或条形图）查看数据。
- **逐层细化地图绘制** 通过简单的点击即可详查您的数据。对于 OLAP/DSS，您可以通过点击令用户逐层细化到地图上的区域。
- **数据绑定** 您的地图可以并入来自在其中嵌入 MapX 的容器的数据、来自 ODBC 的数据或来自 DAO 数据源 （例如 MSAccess）的数据。 MapX 提供若干不同类型的数据绑定，包括 ZIP 编码层级 geocoding。
- **注释** 提供方位，突出显示特定数据，并且通过添加文本、符号和标签令您的地图更具信息性。
- **图层化** 显示和控制地图图层的显示，使其只在地图的缩放级别属于预设的距离内时显示。还使用或创建无缝的地图图层，可以像一个表一样处理一组基础表。通过特殊图层类型支持特殊的应用程序类型，这些特殊图层包括动画图层 （用于实时跟踪）和 UserDraw 图层 （用于在地图顶部绘制特殊地图元素，例如徽标）。

- **光栅图像** 包括光栅图像底层，为您的地图添加吸引人的、清晰的背景。
- **自动加标签** 自动向您的地图添加标签，以及控制其属性和显示。
- **选择** 通过对数据进行组合和组织，充分发挥 MapX 的分析功能。选择特定半径、矩形或特定点内的地图图元。
- **图元出厂值** FeatureFactory 对象使您可以创建、合并、存入缓冲区或清除点、线条和区域图元。
- **工具** 您的用户可以通过单击和拖放与地图直接交互。使用 MapX 的内置导航、选择和加标签工具，或者创建定制的工具以满足您自己的应用程序的要求。
- **地图编辑** 您可以使您的用户能够添加、修改或删除地图上的图元。
- **投影和坐标系统** 通过完全支持坐标系统和地图投影， MapX 允许您微调地图的显示并在本机坐标中处理 X-Y 数据。
- **远程空间服务器连接** 连接到在 Oracle、 Informix、 SQL Server 或其它支持的 SpatialWare 数据库上运行的 Oracle Spatial 和 MapInfo SpatialWare 中存储的活动数据。通过空间服务器，公司可以在企业数据库中存储它们的地图数据，以便集中管理和确保安全性。 SpatialWare 和 Oracle8i Spatial 等空间服务器提供了高级查询处理功能，提高了一个组织的空间数据在服务器上的性能。在 RDBMS 中存储空间数据对于要求大量地图编辑并处理大数据集的应用程序而言也是必需的。

学习 MapX

编写本文档的目的在于，向有关人士介绍如何使用 MapX 和面向对象的编程语言 （例如 Visual Basic、Visual C++ 或 Delphi）轻松地向其应用程序添加地图绘制功能。

如果您是第一次使用 MapX，下面是一些如何学习 MapX 的指导：

- 阅读第 37 页的第 2 章：“MapX 基础”。
- 查看示例应用程序并将它们用作创建您自己的应用程序的模板。在您安装 MapX 时，示例应用程序安装在您的计算机上，在 < 指向 MapX 的路径 >\ Samples50 路径下可以找到它。
- MapX CD 提供不自动安装在您的计算机上的附加示例程序。在 CD 上的 \Samples 目录下查找这些示例程序。您还可以在 MapX Web 站点 **<http://www.mapx.com>** 上找到最新的示例应用程序。

MapX 基础

在 “速成教学”一章中，我们使用 MapInfo MapX 控件创建了简单的 MapX 地图；在上一章中，我们介绍了一些 MapX 制图概念。本章将提供用于在应用程序中创建和控制 MapX 地图的主要组件概览。在后续章节中，我们将详细讨论以下选定主题。

2

章

- Map 对象
- 属性页
- 图层
- GeoSet
- Dataset
- 注释
- MapX 中的可创建对象

Map 对象

在 MapX 对象模型 （参见对象模型贴示或单击联机帮助窗口上的对象模型按钮）的 MapX 对象层次上，您将会看到该地图对象本身位于顶部。每个 MapX 对象、属性和方法均派生自 Map 对象。在该地图对象之下显示的每个属性和方法均有助于构建整体 Map 对象。各个 Map 对象主要由 Dataset、 Layer 和 Annotation 对象定义。

下表显示了由数字值表示的某些 Map 对象。这些属性既可在设计阶段更改，也可使用以下代码示例在运行时更改：

属性	说明	代码示例
缩放	设置在地图上显示的英里数 （默认测量单位）。	Map1.Zoom = 500
旋转	将地图旋转指定的度数。	Map1.Rotation = 179
CenterX	设置地图中心点的 X 坐标。 X 坐标和经度或东向位移有关，具体取决于地图投影。	Map1.CenterX = -79.4458
CenterY	设置地图中心点的 Y 坐标。 Y 坐标和纬度或北向位移有关，具体取决于地图投影。	Map1.CenterY = 44.9932

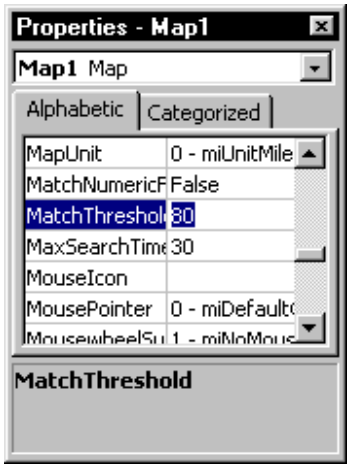
使用 Map 对象，可以通过控制地图的若干方法和属性，控制地图如何显示。此处有部分属性由其他对象表示。例如，看到 MapX 地图时，实际上正在看到的是各个图层的集合，由 Layer 集合表示。该 Layer 集合是 Map 对象的属性之一。

现在，在让我们来看一下有关更改地图属性的问题。“属性页”可用于控制地图对象的众多属性。

属性页

在设计和测试应用程序时如要更改地图的属性，“属性页”极其实用。要在 Visual Basic 项目中访问“属性页”，请执行以下操作：

从 “属性”窗口，选择 “视图” > “属性窗口”，然后单击 **(定制)**。



1. 单击 **(定制)** 行上的按钮。

此外在设计阶段，右键单击表单中的 “Map 对象”，然后选择 “属性 ...”，也可访问 “属性页”。

在运行时也可查看 “属性页”。为此，可向 Map.MouseUp 事件添加以下代码。

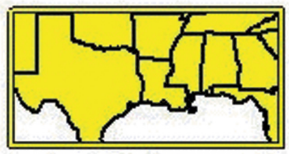
```
If Button = 2 Then Map1.PropertyPage
```

这将测试在用户单击鼠标右键时，是否在运行时显示 “属性页”。使用鼠标右键按钮，其与工具的正常使用没有关系 （左键）。

注：在设计阶段和运行时，只应该使用此代码，它将为用户提供更多的地图控件。但如果是要在产品交货时向最终用户显示对话框，则需要使用 Layers.LayersDlg method 方法取代 PropertyPage 方法，因为 LayersDlg 对话框更加友好。

图层

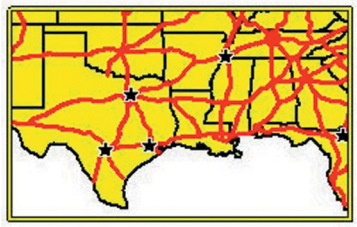
让我们将地图上由黑星代表的点考虑为一个数据库。这一地图本身并不是很实用，但是当将点图叠加在线图和区域图之上时，所得到地图将会非常实用。每个单独的地图均称为图层， MapX 将地图存储为图层的集合。



区域

线

点



这些图层既可在设计阶段通过 MapX 的 “属性”对话框更改，也可在运行时通过编程更改。使用 “属性”对话框，设计者只需更改设置即可控制图层。在第 65 页的第 5 章：“在图层中绘制地图”，我们将讨论可用于随时在程序中更改图层属性和方法的代码。添加新的图层、移除图层和更改图层的样式或可见性均属于可在图层上执行的操作。

Layer 集合

Layer 集合由 Layer 对象 （0 至 n 个）构成。 Layer 对象由图元集合构成，其中各个图元具有其各自的属性和样式。图元集合由 Feature 对象构成，相应于地图上图元，例如点、线或区域。在此既可以创建独

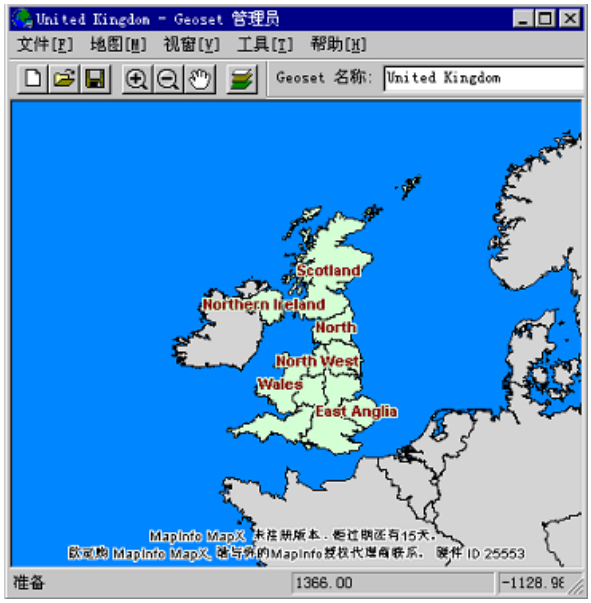
立的 Feature 对象，也可以获取 Feature 对象集合。我们将在第 105 页的第 7 章：“图元和选择集合”中讨论图元。

GeoSet

GeoSet 是图层及其设置的集合。 GeoSet 确定了在 Map 对象及其设置中使用的 MapInfo 表的集合。GeoSet 可在设计阶段指定。如果在运行时设定，则它将先移除所有加载的图层和 Dataset，然后加载新的 GeoSet。默认加载的 GeoSet 是 US.GST。如果熟悉 MapInfo 专业版，则相应的工作区和 MapX 中的 GeoSet 类似。

指定在运行时要添加的备选 GeoSet、向地图对象添加图层或控制图层的外观均可通过 “属性页”来完成。

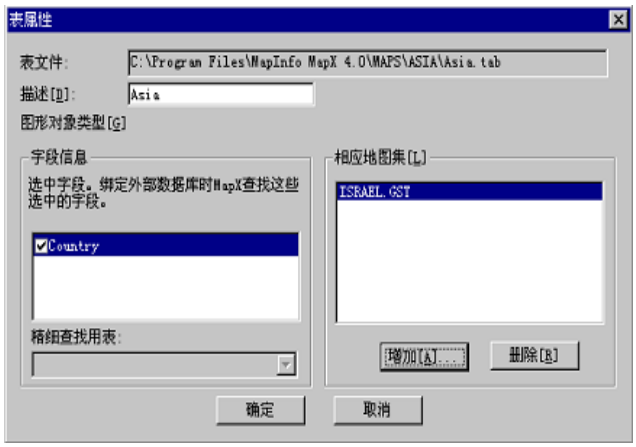
带有指定 GeoSet 的英国地图如下。



在对地图满意之后，可以保存它。这将在驱动器上写入 GeoSet (*.GST) 文件。打开该 GeoSet 文件时，所有的地图图层和设置都将呈现为其保存时的状态。 GeoSet Manager 可用于修改图层、管理缩放级别、表和其他属性。

Dataset

Dataset 可用于将数据绑定到地图。例如，如果使用按城镇划分的销售数据 MS Access 数据库，和可以将数据绑定到的相应城镇地图，以及按照城镇划分的销售模式的点趋势图无法在制表数据中轻松地突出显示。如上所示，如果需要 MapX 指定数据和地图之间的匹配，则匹配将通过称为自动匹配 / 自动绑定的进程来确定。您必须先将地图注册到 GeoDictionary 才可以利用自动匹配 / 自动绑定功能。



在将数据绑定到地图之后，可以按照地理方式来查看相关信息。借助于数据的可视表示，可以创建主题地图。主题是地图的颜色编码，用于在地理上表示数据趋势。

注释

使用 Annotation 集合可以将文本和符号轻松地置于地图之上。注释位于所有其他图层之上，并且不链接到任何数据。如果熟悉 MapInfo 软件，则注释的目的类似于地图的装饰图层。

以下列出了用于 Annotation 集合对象的方法和属性：

Annotation 集合方法

方法	说明	代码示例
AddSymbol	向 Annotation 集合添加符号。使用默认样式（正如在 Map.DefaultStyle 指定）。	Map1.Annotations.AddSymbol X, Y
AddText	向 Annotation 集合添加文本。第四个参数是文本相对于给定坐标的初始位置。	Map1.Annotations.AddText _ “开发人员服务”， _ 79.44, 46.8889, _ miPositionTL
移除	从集合移除特定的注释。	Map1.Annotations.Remove 1
RemoveAll	从集合移除全部注释。	Map1. Annotations.RemoveAll

Annotation 集合属性

属性	说明	代码示例
可编辑	指定注释是否可以编辑。	Map1.Annotations.Editable _ = True
类型	指定 Annotation 对象类型。	If Map1.Annotations(2).Type_ = miTextAnnotation Then _ Print “It is text”
图形	其中包含 Graphic 对象，具有用于 Annotation 的属性。	请参阅第 324 页的 “Graphic 对象”。

注意此处没有用于注释对象中的位置、符号样式或文本的属性。注释的图形属性包含图形对象，其中包含这一信息。要修改注释，可修改注释的图形对象。

以下代码向指定位置添加符号：

```
'Add a symbol at location
Map1.Annotations.AddSymbol X1, Y1
```

MapX 中的可创建对象

在 MapX 对象模型中，以下对象为可创建。换而言之，您可以使用这些对象类创建独立的对象。

AffineTransform	BindLayer	BitmapSymbols	CoordSys
Datum	Feature	Fields	LayerInfo
Map	ODBCQueryInfo	OCIQueryInfo	Parts
Point	Points	Rectangle	RowValue
RowValues	State	Style	Variables
NotesQueryInfo	NotesViewInfo		

以下示例显示了如何创建独立的样式对象，显式样式选择对话框，并使用新样式来设置图层的覆盖样式。

使用 Visual Basic、 Delphi 等编程语言时，您需要在对象名 （例如 MapX.Style.5）之后指定 MapX 版本号。

注：在 Visual Basic 中，只有在表单中没有嵌入 MapX 对象时需要指定 MapX 版本号。

C++ 示例

```
CMapXStyle style;
style.CreateDispatch(style.GetClsid());
// 还可使用 style.CreateDispatch("MapX.Style.5"),
// 但是上述示例在 MapX 的不同版本之间更容易移植
style.PickRegion();
m_ctrlMapX.GetLayers().Item(1).SetStyle(style);
```

Delphi 示例

```
Var
   s : variant;
begin
     s := CreateOleObject('MapX.Style.5');
     s.PickRegion;
     MapObject.Layers.Item(1).Style := s;
end
```

Visual Basic 示例

如果在 Visual Basic 表单中已经嵌入了 MapX 对象 …

```
Dim s as new MapXLib.Style
s.PickRegion
Map1.Layers(1).OverrideStyle = True
set Map1.Layers(1).Style = s
```

如果在表单中没有嵌入 Mapx 对象 …

```
Dim objStyle as new object
Set objStyle = CreateObject("MapX.Style.5)
Map1.Layers(1).OverrideStyle = True
objStyle.PickRegion
set objMap.Layers(1).Style = objStyle
```

3 章

- 组织您的数据和地图：表概述
- 什么是 GeoSet?
- 地图图元
- 将数据放置于地图上
- MapX 的强大之处

地图绘制概念

现在您已安装了 MapX 并被各式各样的图元和功能所吸引，可能十分渴望立刻开始地图绘制。但在开始前，应首先花几分钟时间来阅读本章，如果您是 MapX 的新手则更应如此。本章将加深您对一些概念的理解，以便成功使用 MapX 进行地图绘制。

组织您的数据和地图：表概述

若要使用 MapX，您需要包含来自 MapInfo 的记录和地图的文件。 MapX 将其所有基础信息以 MapInfo 表的形式组织起来；每一表都是一组 MapInfo 文件，用来在地图中建立一个图层。

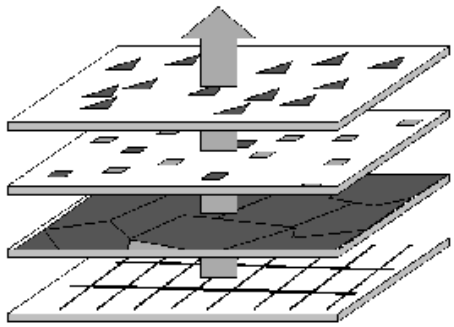
文件如何构成表

所有 MapInfo 表都将具有以下文件：

- <Somefile>.tab：该文件描述 MapInfo 表的结构。它是描述包含数据的文件的格式的小文本文件。
- <Somefile>.dat （.mdb、 .aid 或 .dbf）：这些文件包含表格数据。
- <Somefile>.map：该文件描述图形对象 （如果该表没有任何地图对象，则该文件将不存在）。
- <Somefile>.id：该文件是将数据与对象相链接的交叉引用文件 （如果该表没有如何地图对象，则该文件将不存在）。
- <Somefile>.ind：它是索引文件。通过该索引文件，您可以使用 Find 对象搜索地图对象。

MapInfo 表和 MapX 图层

每一可绘制地图的 MapInfo 表都可以在地图中显示为一个图层。例如，您可以显示顾客表、街道表以及国界表。



我们可以将这些图层视为透明的部分，每个图层包含地图的不同组成部分。这些图层依次堆叠在一起，用户即可看到地图的全貌。

什么是 GeoSet?

geoset 保留地图图层及其设置的集合，以便于您使用。 Geoset 是由同一地理区域的标准 MapInfo 格式地图文件 (.tab) 组成的数据集，因此命名为 geoset。 Geoset 可以帮助您避免在每次要作为示例地图处理图层时要分别打开和显示这些图层的耗时的工作。 geoset 的扩展名为 .gst。 .gst 是包含若干元数据关键字的文本文件，告诉 MapX 显示哪些表以及如何显示它们。

在打开一个 geoset 时，它自动向默认显示打开在 geoset 中包括的所有文件。开发人员可以更改该默认显示以满足自身的要求。 geoset 的设置包括投影、默认缩放、对象的自动加标签、缩放图层以及在打开时表是否可见。 MapX 也将打开开发人员指定的任何单个 (.tab) 地图文件。 Geoset 是出于方便目的提供的，不是 MapX 行使功能所必需的。

MapX 将不打开 MapInfo 工作区 （.wor 文件类型）。

地图图元

我们已在前面提到， MapX 中的地图是由多层地图对象构成的。这些地图对象是通过 Feature 对象在 MapX 中访问的。有四个基本图元类型：

- 区域：覆盖给定区域的闭合的对象。这些对象包括多边形、椭圆和矩形。例如，国界、邮政编码边界、销售地区。
- 点对象：表示单个数据位置。例如，客户位置、餐馆、停车计时器。
- 线条对象：覆盖给定距离的开放的对象。这些对象包括线条、折线和弧线。示例为街道、河流、电线。
- 文本对象：描述地图或其它对象的文本，例如标签和标题。

您可以在单独的图层中具有每一对象类型 （最常见），或者可以在同一图层中组合对象。 MapX 帮助您创建、编辑、定制和显示这些对象，使地图满足您的需要。

将数据放置于地图上

数据集可用于将数据绑定到地图。例如，如果使用按城镇划分的销售数据 Microsoft Access 数据库，则可以将数据绑定给它并在相应城镇地图上显示它，以用于无法单独在制表数据中轻松地标识的按照城镇划分的销售模式的点趋势图。

当今的商业中有许多不同类型的数据库；因此， MapX 使您可以绑定到多种不同类型的数据源。 **Datasets.Add** 方法的第一个参数可以指定 DatasetTypeConstants 值，该值指定要进行数据绑定的类型。有关您可以绑定的数据源的列表，请参见第 85 页的第 6 章：“将数据加入地图”。

用来完成工作的工具

大多数地图绘制应用程序提供各种工具栏按钮 （工具）来协助完成常见的绘图任务 （例如在地图上绘制线条）和导航任务 （例如放大）。 MapX 提供若干常见地图绘制工具，并且您还可以创建自己的定制工具。

标准工具

使用 MapX，您可以轻松地将常见工具并入您的应用程序中，而无需从头开始。 MapX 提供对若干常见地图绘制工具的内置支持，包括：

- 令用户更改地图的比例和位置的导航工具 （放大、缩小、平移、居中）。
- 让用户单击地图图元以给它加标签的加标签工具。

- 为用户提供各种方法来选择地图图元的一组选择工具。
- 注释 （符号和文本）。

定制工具

如果需要 MapX 未提供的工具栏按钮类型，只需通过使用 CreateCustomTool 方法创建一个定制工具。

MapX 的强大之处

现在您已对表格、图层、 geoset、地图图元、数据集和工具有了大致了解，可以准备付诸实施 MapX 的完整功能了。使用 MapX，您可以搜索地图中的图层并确定特定图元在该图层内的位置。例如，您可以使用它查找 Lackawaxen, PA. 的最近的经销商。或者，您可以计算保健供应商与其患者之间的距离，然后算出在特定医院的给定半径内居住着多少患者。或者，您可以根据每一区域的总客户数或根据在去年内购买了产品的客户数，遮蔽边界。 MapX 将此称作主题地图绘制。

随着您逐渐掌握 MapX，您会发现只有您的想像力的局限才会限制其应用。

设计阶段 MapX 速成教学

本章介绍成功安装 MapX 的必要信息，并提供了 MapX 的快速概览。初次使用 MapX 的用户应该阅读本章。

4 章

- 何为 MapX?
- 安装 MapX 之前的注意事项
- 安装 MapX
- 添加地图控件
- 从较早的 MapX 版本升级 Visual Basic 应用程序
- 从初期的 MapX 版本升级 C++ 应用程序
- Visual Basic 速成教学
- Visual C++ 速成教学
- Delphi 速成教学
- MapX 文档集
- 后续内容

何为 MapX?

MapX 是用于应用开发人员的工具。它为将绘图功能嵌入到新的和现有的应用程序中提供了最轻松和最高效的途径。 MapX 是一个可以使用面向对象的语言迅速集成到客户端应用程序的 DLL，例如 Visual Basic、 Delphi 和 Visual C++ 等语言。

系统要求

MapX 要求使用 32 位版本的 Windows 98 （第二版）、 2000 （服务包 2）、 Windows NT 4.0 （服务包 6A）或 Windows XP。

MapX 包含的内容

MapX 软件包包括以下组件。

注：如果在安装 MapX 时取消了某些选项，则部分下列文件将不会安装在系统中。默认情况下，文件均安装在以下文件夹之内或之下：Program Files\MapInfo\MapX 5.0

程序文件

MapX DLL 以及其他各种 DLL 和支持文件。

示例地图

全球不同地区的地图文件的集合。其格式包括：

- MapInfo TAB
- ImagePro
- GDT 2001
- StreetPro
- 军用网格参考系统 (Military Grid Reference System, MGRS)

示例数据

Microsoft Access 数据库包括示例的人口数据、 Oracle 数据库包括空间和非空间数据，以及 SQL Server 数据库。

示例应用程序

有关最新的示例程序，请访问 MapX Web 站点：**http://www.mapx.com**

实用程序

MapInfo Geodictionary Manager （GeoDictionaryManager50.exe），用于注册和 MapX 一起使用的表； GeosetManager，用于轻松创建 Geoset。

MapX 文档

购买 MapX 的客户所接收到的 **MapX 开发人员指南**。该指南中详细列出了有关制图概念和有效使用 MapX 的方法。（软件的 30 天评估版中不含此指南。）

MapX 全面的 “联机帮助” 系统也是一个电子版的 “MapX 开发人员指南”，其中包含了 “MapX 对象” 一书中的大量代码示例。

注：尽管 MapInfo 已经作出了大量努力，测试联机帮助中的示例代码，尽可能地确保精确和实用性，但是可以确定出于种种原因，部分代码示例可能无法按照设计意图在您的 MapX 应用程序中工作。为了能够使其正常工作，可能会需要对现有代码做出微小调整或 “修饰”。

安装 MapX 之前的注意事项

安装 MapX 之前，应该将序列号记录在易于记忆的地方，例如手册标题页。务必填写已经提供的付费之后的注册卡，并将其返回给 MapInfo Corporation。

注：我们强烈建议您移除以前版本的 MapX，并在安装之前退出所有 Windows 程序。

卸载旧版 MapX

卸载旧版 MapX 是一个简单易行的过程。具体步骤如下：

1. 选择 “开始” > “程序” > “MapInfo MapX v.X” > “卸载 MapX v.X”
2. 此时将出现 “卸载向导” 屏幕和一个对话框，用于确认是否要移 MapX 及其所有组件。单击 **“确定”**。

Unicode 版本的 MapX

MapX 是一个 Unicode 应用程序。但是，如果您正在运行 Windows 98， MapX 将删除此版本，并将安装一个非 Unicode 的版本。

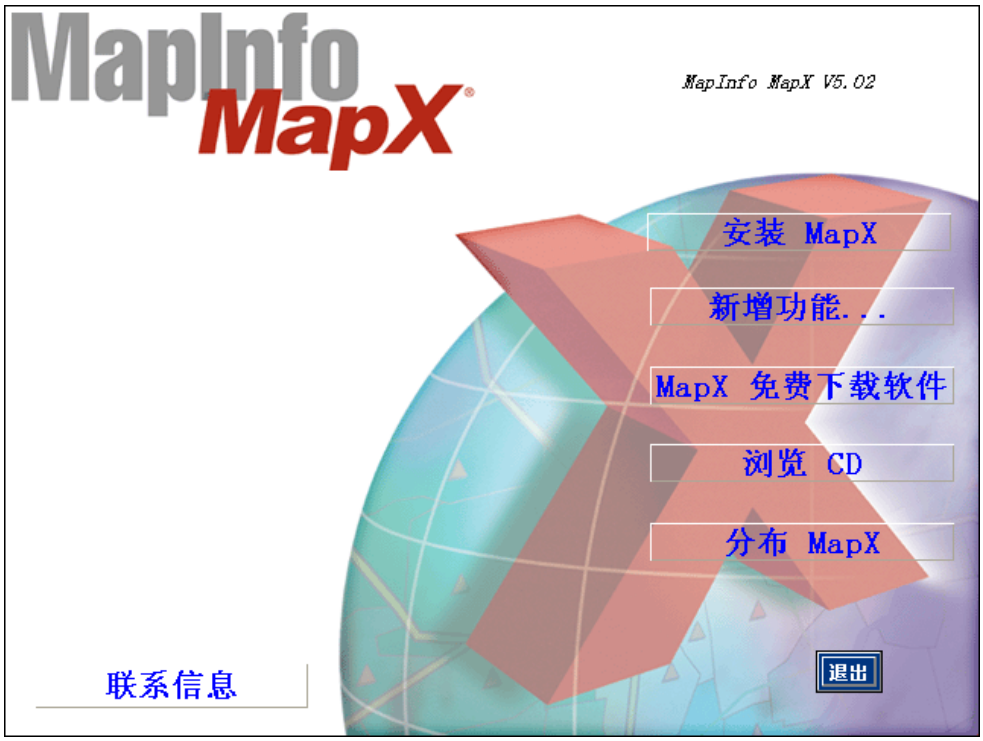
安装 MapX

MapX 安装步骤如下所示。

如果 Windows 配置中没有 MapX 程序群组，则安装进程将会创建相应群组。如果 Windows 配置中已有 MapX 程序群组，则安装进程将在该群组之中创建新的 MapX 图标。

要安装 MapX，请执行以下操作：

1. 将 MapX CD 置于 CD 驱动器中 （例如 D:）。CD 应该自动运行，否则可单击 Windows 的 “开始” 按钮，然后选择 “运行”，在打开的下拉菜单中键入或选择 [CD 驱动器盘符]\Setup.exe（例如 D:\Setup.exe），然后单击 **“确定”**。
2. 此时将出现一个欢迎屏幕，提供若干选项。单击 **“安装 MapX”** 按钮。



3. 此时将显示 “欢迎” 屏幕。选择 “下一步”，继续进行安装进程。
4. 此时将显示 “软件许可” 屏幕。选择 **“是”**，接受协议条款，然后继续安装进程。
5. 此时将显示 “选择目标位置” 屏幕。指定安装 MapX 的目录。如果当前没有安装 MapX，默认位置为：Program Files\MapInfo\MapX 5.0

   如果当前已经安装了 MapX，则默认为当前的安装目录。

注：我们强烈建议您移除以前版本的 MapX，并在安装之前退出所有 Windows 程序。

要指定其他位置，选择 “浏览” 按钮，然后指定目标位置。单击 **“下一步”**，继续进行安装进程。

6. 指定要安装的产品组件。此时将显示所选组件的磁盘空间要求。在选择组件之后，将显示该组件的说明。如果启用了 “更改” 按钮，则表示该组件具有子组件。单击 “更改” 按钮，显示相应组件及其空间要求的列表。选中要安装的子组件。例如，如果选择 “导出 / 导入格式” 组件，将会显示 5 个子组件（GIF、JPG、TIF、PSD 和 PNG），并且提供了每个子组件需要的磁盘空间。您可以选择安装任意或所有子组件。
7. 此时将显示 “选择程序文件夹” 屏幕，指定程序文件夹。
8. 此时将显示 “开始复制文件” 屏幕。检查显示的信息。如果该信息正确，选择 **“下一步”** 以安装 MapX。进度栏显示了安装的状态。如果要更改信息，选择 **“上一步”** 以返回上一屏幕。
9. 在 MapX Installer 安装完毕之后，将会启动 “MapX 数据设置” 对话框。重复上述详细介绍的各个步骤，完成这一安装。

添加地图控件

安装 MapX 之后，需要添加地图控件。

对于 Visual Basic 用户

在 Visual Basic 工具箱上安装地图控件。在打开的 Visual Basic 项目中执行以下操作：

如果您正在使用 Visual Basic 4：

1. 右键单击 Visual Basic 工具箱，然后从快捷方式菜单中选择**定制控件**。
2. 在 **“定制控件”**对话框中，在列表中查找 MapInfo MapX v5.0。如果没有选择这一项目，则将其选中。单击 **“确定”**。

如果您正在使用 Visual Basic 5 或更高版本：

1. 右键单击 Visual Basic 工具箱，然后从快捷方式菜单中选择 **“组件”**。
2. 在 **“组件”**对话框中，单击**控件**选项卡，然后在列表中查找 MapInfo MapX v5.0。如果没有选择这一项目，则将其选中。单击 “确定”。

工具箱上此时会出现地图控件。要将地图置于 Visual Basic 表单之上，选择地图控件，然后在表单上绘制一个图框。

如果您保存了项目，则下一次重新加载项目时，地图图标将会自动出现在工具箱中。

对于 C++ 用户

您需要在项目中包括 MapX.cpp 和 MapX.h。其中包括用于访问 MapX 控件的类定义和方法实现。MapX.h 和 MapX.cpp 文件可在安装 MapX 目录的 Samples50\CPP 子目录中。

使用 Visual C++ 4.0 版本

从 **“插入”**菜单，选择 **“将文件插入到项目”**。选择 MapX.cpp 作为要插入的文件。

注：请勿选择 **“插入 > 组件”命令**。否则将会创建 .cpp 文件，但是该文件不完整。

使用 Visual C++ 5.0 和 6.0 版本

从 **“项目”**菜单，选择 **“添加到项目 > 文件”**。选择 MapX.cpp 作为要添加的文件。

注：请勿选择 “项目” > “组件和控件”命令。否则将会创建 .cpp 文件，但是该文件不完整。

从较早的 MapX 版本升级 Visual Basic 应用程序

如果 Visual Basic 项目使用早期版本的 MapX，可使用以下步骤来将该应用程序转换到当前版本。

1. 打开要升级到当前版本的应用程序的项目。
2. 从表单删除地图对象。记录地图控件的名称，以及任何可以从其默认设置更改的属性的名称。
3. 从 “工具”菜单，选择 “定制控件”。
4. 在 “可用控件”列表中，取消选择 MapInfo MapX 控件，然后选择 “确定”。

5. 返回到 “工具” > “定制控件”对话框，然后选中 MapInfo MapX 版本 x 控件 （其中的 x 是当前的版本号），再单击 “确定”。
6. 将新的 MapX 地图控件置于表单之上，然后使用和此前版本控件相同的名称命名 （例如 Map1）。

   将新的 DLL 添加到项目中之后，您将需要恢复任意定制的属性。

从初期的 MapX 版本升级 C++ 应用程序

如果您使用早期版本的 MapX 来编写 C++ 应用程序，您将需要使用新的 MapX 打包程序类 （mapx.h 和 mapx.cpp）来将您的应用程序升级到当前版本的 MapX。

您需要修改传递到 CreateDispatch 的字符串。使用较早版本的 MapX，您可能会使用字符串指定对象名，例如：

```
Flds.CreateDispatch("MapX.Fields")
```

使用当前版本的 MapX，您需要指定其他字符串：

```
Flds.CreateDispatch("MapX.Fields.4")
```

要令您的代码和将来版本的 MapX 更好地兼容，您可以使用 GetClsid 来替代字符串。由 GetClsid 返回的结果将可用于当前和将来版本的 MapX。例如：

```
Flds.CreateDispatch(Flds.GetClsid())
```

Visual Basic 速成教学

创建简单的地图

使用 MapX，可以轻松向应用程序添加地图。事实上，您可以向 Visual Basic 表单添加工作地图，无需编写任何代码。

1. 从 Visual Basic 工具箱选择地图控件。（如果地图控件上没有工具箱，请参阅本章第 55 页的 “添加地图控件”。
2. 在表单上绘制图框，表示要令地图出现的区域。 MapX 显示地图的预览。
3. 右键单击地图控件，然后从快捷方式菜单中选择 “属性”。此时将出现 MapInfo MapX “属性”对话框。

4. 确定 “当前工具”选项的位置，靠近于 “常规”选项卡的底部。将 “当前工具”设置为 “1003 - 放大”，然后单击 “确定”。
5. 运行程序，查看地图。请注意当光标位于地图之上时，它将更改为带有加号图标的放大镜。
6. 单击地图。 MapX 放大了您单击的地图位置。您可以重复单击，逐渐放大。您也可以拉出选取框，指定要放大的确切区域。

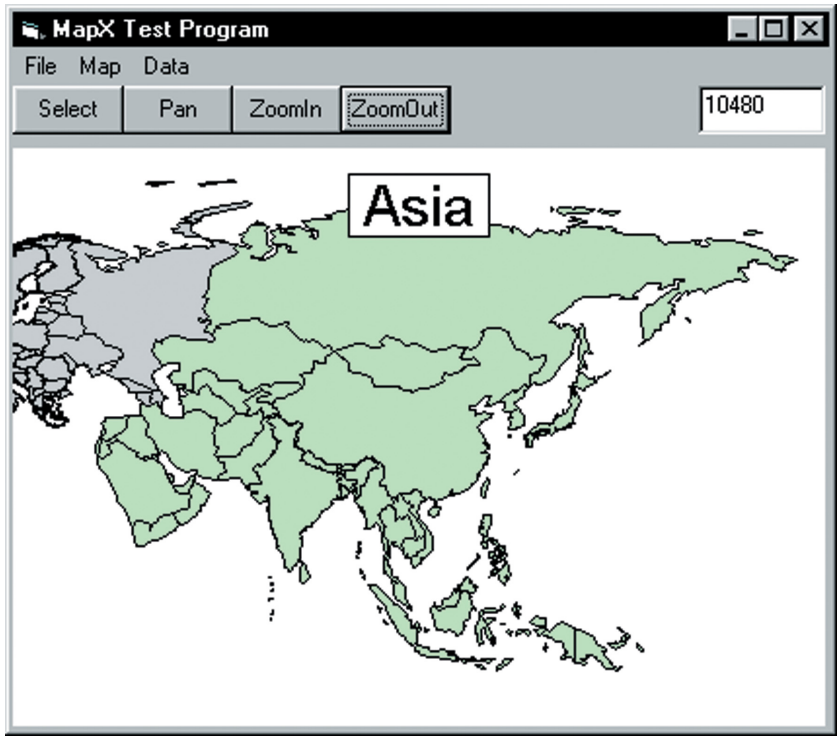
   随着放大的倍数的增加，您将会注意到可以见到更多的地图图元。这是由于单独的图层已经使用 “缩放图层”（在预设缩放范围内自动显示图层）设置。

要缩小图层，可按住 CTRL 键，然后再次单击地图。请注意在按住 CTRL 键时，光标将更改为带有减号的放大镜。

学习示例应用程序是了解 MapX 的方式之一。在文件夹 \Program Files\MapInfo\MapX 5.0\Samples50 中查找示例应用程序

理解 ASIA.VBP

此项目显示带有地图的简单表单。地图上的按钮允许用户选择各种工具 – “选择”工具、“平移”工具、“放大”和 “缩小”工具。左上部的 TextBox 允许用户读取或设置地图的缩放距离。



摘要

“主”表单包含 Map 对象 (Map1)。这一对象置于使用地图控件的表单上。

如果 Visual Basic 工具箱没有为您显示此控件，则需要向工具箱添加控件。

这一应用程序显示了 “亚洲数据”，因为程序员选择了 “亚洲” geoset。通过右键单击 Map 对象，然后选择 “属性”，在设计阶段选择 geoset。“属性”对话框还可用于设置众多其他地图属性。

当用户单击左上部的按钮之一时，应用程序将使用 Map.CurrentTool 属性来更改当前使用哪一工具。例如：

```
Private Sub Command4_Click()
  map1.CurrentTool = miSelectTool
End Sub
```

右上部的文本框允许用户键入预期的缩放距离 （地图上的距离），以英里为单位。该应用程序使用 Map.Zoom 属性应用新的缩放距离：

```
Private Sub Text1_LostFocus()
  map1.Zoom = Text1.Text
End Sub
```

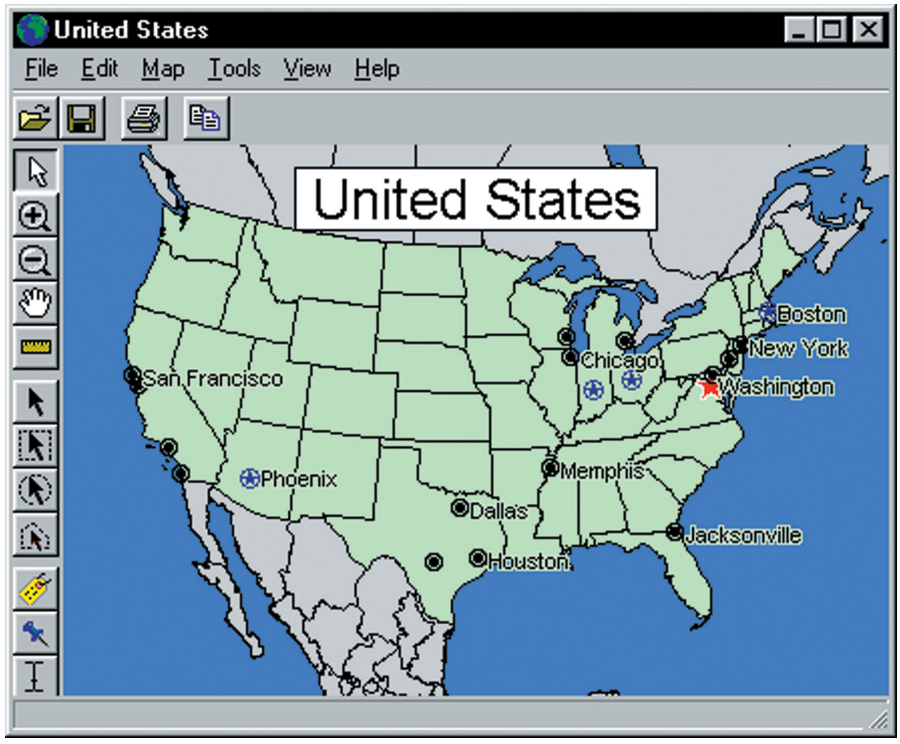
当用户进行缩放时 （使用 “放大”或 “缩小”按钮），地图的缩放距离将会改变。应用程序将使用 MapViewChanged 事件，自动更新右上部的文本。

```
Private Sub map1_MapViewChanged()
  Text1.Text = map1.Zoom
```

```
End Sub
```

理解 VBsample.VBP

此项目显示带有地图的另一个简单表单。但是，它只有在使用 Visual Basic 5 或 6 是才可运行：这一地图默认为美国数据，但是设计处理任意指定的 geoset。通过右键单击地图对象，然后选择属性，可在设计阶段选择其他 geoset。



此项目将使用三种类型的数据绑定。如下所示：

- 点参考
- XY 坐标
- 正常

如果您选择 “点参考” 或 XY，则将会创建图层。有关图层的详细信息，请参阅**第 65 页的第 5 章：“在图层中绘制地图”**。

工具

该工具栏位于屏幕左侧。将图标拖到图标上，确定图标的含义，然后单击预期的工具。您还可以使用 “工具” 菜单访问工具。

箭头工具: 作为默认的指针 / 光标工具。

放大: 获取较近距离的地图视图。

缩小: 获取较宽距离的地图视图。

平移: 在窗口中重新定位地图。

标尺: 测量两个定义点之间的距离。

选择: 选择地图上的对象。

选择矩形: 选择给定矩形中的对象。

选择半径: 选择给定圆形区域中的对象。

选择多边形: 选择给定区域中的对象。

标记: 使用给定数据库中的信息标记对象。

注释工具: 在地图上放置点符号。

添加文本注释: 向地图 添加标题或标签。

插入数据和主题

您可以使用 “地图”菜单中的**插入数据功能**。选中之后，将出现 “添加 Dataset”对话框。选择要添加的数据。

要创建或更改主题，可在 “地图”菜单中选择 “创建”或 “修改”主题，此时将会出现其各自相应的对话框。有关主题绘图的详细信息，请参阅第 111 页的第 8 章：“主题地图绘制和分析”。

查找功能

查找功能命令位于 “地图”菜单中。选中之后，将出现 “查找”对话框。用户将定义在搜索中使用的图层、对象和对象值。此外还可选用优化设置。有关查找功能的详细信息，请参阅第 125 页的第 9 章：“查找地图上的图元”。

在距离之内选择

要在距离之内进行选择，用户必须先使用一个选择工具来选择要搜索的区域。然后则必须从 “地图”菜单选择 **“在距离之内选择”**。此时将出现 “在距离之内选择”对话框。用户将指定要选择的图层、距离、测量单位和图元。有关 “在距离之内选择”的详细信息，请参阅第 105 页的第 7 章：“图元和选择集合”。

查询选择

使用选择工具选择查询区域。“查询选择”功能可以访问与地图关联的数据。该功能位于 “地图”菜单中。

Visual C++ 速成教学

学习示例应用程序是了解 MapX 的方式之一。在以下文件夹中查找示例应用程序：

MapInfo\MapX 5.0\Samples50

在研究示例 C++ 应用程序 mapxsamp.cpp 时，请参阅第 189 页的第 14 章：“使用 Visual C++”以获取帮助，理解如何和 C++ 一起使用 MapX。

注：这些主题假定您正在使用 Microsoft 的 “文档 / 视图”模型 （标准的 MFC AppWizard 应用程序）。示例应用程序可以在 Developer Studio 中使用 mapxsamp.mdp 生成。

Delphi 速成教学

添加地图控件

这些步骤将 MapX 安装到 Delphi 程序包内。这一进程只需要完成一次。

1. 使用新的空项目打开 Delphi。
2. 从 “组件”菜单选择**导入 ActiveX 控件**。
3. 在列表框中，选择 **MapInfo MapX v5.0**，然后单击 “安装”。在 “安装”对话框中，安装到默认程序包，即应为 Borland 用户组件。按 “是”以重新编译该程序包，然后关闭并保存 “程序包”窗口。 MapX 图标应该出现在 “控件”调色板之中，位于 ActiveX 选项卡之下。

创建简单的地图

这些步骤用于创建新的项目，其中包括 MapX。

1. 从 “工具”调色板选择 MapX 图标 ，该调色板位于 ActiveX 选项卡之下。
2. 在表单上绘制图框，表示要令地图出现的区域。 Delphi 随后将显示表单上的地图的预览。
3. 从 “运行”菜单选择 “运行”，以运行新的应用程序。

MapX 文档集

MapX 文档包括印刷的 *The MapX Developer Guide* 一书和 “联机帮助”系统。

MapX 联机帮助系统

MapX 全面的 “联机帮助”系统提供了学习和更加有效使用 MapX 所需的信息。其实质上是一个印刷手册的电子版文档，带有以 Microsoft Visual Basic、 C++ 和 Delphi 编写的附加代码示例。将这些示例剪切和粘贴到 MapX 应用程序，即可加速应用程序的开发进程。

注：尽管 MapInfo 已经作出了大量努力，测试联机帮助中的示例代码，尽可能地确保精确和实用性，但是可以确定出于种种原因，部分代码示例可能无法按照设计意图在您的 MapX 应用程序中工作。为了能够使其正常工作，可能会需要对现有代码做出微小调整或 “修饰”。

您可以采用多种途径来获取信息：

- 使用 “帮助目录”屏幕，从帮助页中选择主题。单击帮助页以显示其主题，然后从列表中选择主题。
- 使用查找功能来搜索特定的单词。选择 “开始”菜单 > “程序” > “MapInfo MapX” > “MapX 联机帮助”然后，单击索引并转至 “查找”选项卡。键入要搜索的单词，然后单击 “重建”。此时将出现一个单词列表，帮助您缩小搜索的范围。单击单词，此时将出现一个包含所选单词的主题列表。双击所需主题，或单击 “显示”以显示相应主题。通过单击**选项**，可以定制搜索，令查找更为精确。
- 使用索引功能来迅速查找主题。键入正在查找的单词的前几个字母。和您键入内容最为接近的主题将会突出显示。单击要显示的索引条目。
- 另请参阅信息：单击任意 “帮助”窗口中的绿色下划线文本，打开相关任务、关键词或短语的有关信息。

“联机帮助”设计在窗口中部分显示，以便您可以在 “帮助”窗口的一侧查看地图、浏览窗口和对话框。当然，您还可以更改 “帮助”窗口的大小，采用最为适宜的工作方式。选择 “帮助” > “始终在前端显示”，将令 “帮助”窗口显示在屏幕之上，以便您可以继续在 MapX 中工作。或者可以使用 Alt-Tab 来在 “帮助”屏幕和 MapX 桌面之间切换。

使用 “联机帮助”的提示

使用代码示例

您可以从 “帮助”窗口轻松地复制代码示例。要复制文本，可以突出显示文本，然后按下 CTRL-C ；将文本粘贴到开发环境，然后再按下 CTRL-V。您还可以使用 “拖放”功能，将突出显示文本从 “帮助”中拖至其他应用程序。

注：尽管 MapInfo 已经作出了大量努力，测试联机帮助中的示例代码，尽可能地确保精确和实用性，但是可以确定出于种种原因，部分代码示例可能无法按照设计意图在您的 MapX 应用程序中工作。为了能够使其正常工作，可能会需要对现有代码做出微小调整或 “修饰”。

下载最新的帮助文件

MapInfo Corporation 定期更新 MapX 联机帮助。要下载这一 “帮助”文件的最新版本，请访问 MapX 网站：**http://www.mapx.com/mapx/html/mapx_docs.html.**

MapX 网站还提供了示例应用程序以及 “讨论区”，您在其中可以张贴问题并与其他 MapX 用户进行交流。

注：在将 MapX 与 Visual Basic 一起使用时，您将会在 MapX 库中发现没有与其关联的帮助主题的属性 / 方法。这些属性 / 方法继承自 Visual Basic。查询 Visual Basic 文档，即可获取有关这些属性的信息。

后续内容

本章提供了 MapX 的概览。本文档的其他章节将以更为详细的篇幅来介绍 MapX 的特定功能。

在开始阅读更多内容之前，花费一些时间来查看 MapX 对象模型图。这一图形包括在 MapX 之中，对于提供 “全貌”非常实用。在学习 MapX 时可将其作为一个参考。

在图层中绘制地图

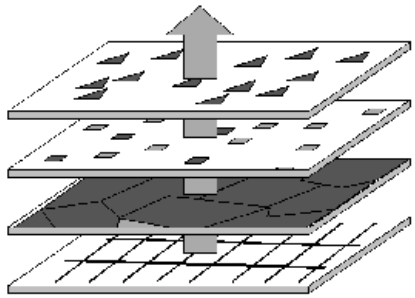
本章介绍表和地图之间的关系，并显示如何将其制作为图层，并达到所需的详细程度。

5 章

- 作为图层的地图
- Layer 集合：地图的构建模块
- Layer 集合的某些属性
- Layer 集合的某些方法
- 使用 LayerInfo 对象创建图层
- Layer 对象
- 图层顺序
- 检查图层
- 检查图元类型
- 缩放图层
- 生成用于图层的标签
- 注释
- 光栅图像
- 动画图层
- 绘制图层

作为图层的地图

在上一章中，我们已经向您介绍了作为图层集合的计算机地图概念。每个包含图形对象的 MapInfo 表均可显示为 “地图”窗口中的图层。例如，您可以显示顾客表、街道表以及国界表。



我们可以将这些图层视为透明的部分，每个图层包含地图的不同组成部分。这些图层依次堆叠在一起，用户即可看到地图的全貌。例如，第一个图层包含国界，第二个图层包含代表首都的符号，而第三个图层包含高速公路。将这些透明部分依次堆叠在一起，即构成一张完整的地图。

现在，在让我们来了解创建地图的具体细节。

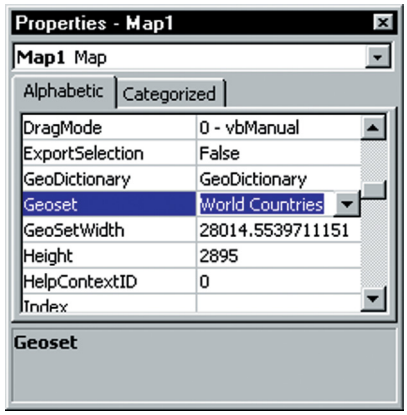
Layer 集合：地图的构建模块

Layer 集合是地图控件的属性之一，包含 Layer 对象。这些 Layer 对象构建自 MapInfo 表，一起构成了地图。每个图层均包含不同的地图图元，例如区域、点、线或文本。 Layer 集合具有用于执行操作的属性和方法，例如从集合添加或移除 Layer 对象等操作。

如何获取 Layer 集合

在设计阶段加载 geoset 是获取 Layer 集合的途径之一。正如我们在此前一章所述， geoset 定义了一个图层及其设置的集合。当您向表单添加地图控件时， MapX 将默认加载美国 Geoset (US.GST)。换而言之，在操作之初，您将会得到美国 Geoset 定义的 Layer 集合。

假定您要编写一个应用程序，该程序在初始状态下为世界地图，即一个构成世界各国地图的多个图层的集合。您也可以在设计阶段指定世界地图 Geoset，实现这一目的。



1. 从 “属性”窗口单击 Geoset。
2. 在 Geoset 行上单击下拉箭头。
3. 从可用 Geoset 列表选择 World Countries （世界地图）。

“世界地图 Geoset”中定义的 Layer 集合将在您运行应用程序时加载。“世界各国”地图将出现在地图控件中。

“属性”页

使用 “属性页”可在设计阶段快速查看构成 Layer 集合的所有 Layer 对象。在设计和测试应用程序时如要更改地图的属性，属性页极其实用。此时，我们将使用属性页来查看 “全球各国”集合中的各个图层。

1. 从 “属性”窗口单击 “定制”。
2. 单击 “定制”行中的 ...。此时将出现 “属性页”。
3. 单击 “图层”选项卡，查看集合中图层的列表。

您可以使用此页来修改集合中的 Layer 对象的众多属性、重新排列图层的顺序、添加或删除集合中的图层。

例如，通过突出显示图层并清除对 “可见”复选框的选择，即可令图层在地图显示时不可见。当显示地图时，该图层即为不可见。自然，您可以在运行时重设图层，通过编程令其可见，例如：

```
Map1.Layers.Item(2).Visible = True
```

我们将会在后续内容中以编程方式接触到更多的图层。

Layer 集合的某些属性

每个地图都具有一个图层的集合。Layer 集合由 Layer 对象构成。Layer 集合具有用于从集合添加或移除 **Layer** 对象的方法和属性。

注：有关 **Layer 集合**方法和属性的完整列表，请参阅第 340 页的 “Layer 对象和 Layers 集合”。

获取集合中的图层数量

“计数”属性对于任意图层都是极其实用的属性。该属性可以告知项目数量，在此情况下是集合中的图层数量。如果要遍历集合中各个项目，例如，获取项目名称，则将会用及此属性：

```
Dim x as integer
For x = 1 to Map1.Layers.Count
          Print  Map1.Layers(x).Name
Next
```

从集合获取图层

“项目”属性从集合获取特定的 Layer 对象。“项目”属性返回一个图层作为对象，并且是 Layer 集合的默认方法。您可以通过索引值引用图层，例如 1、 2 等值，但是您还可以通过其名称引用图层，例如 Highways 或 Cities：

```
Dim lyr as Layer
Set lyr = Map1.Layers.Item("ighways")
```

- 或者 -

```
Dim lyr as Layer
Set lyr = Map1.Layers.Item(3)
```

在应用程序中将会通过 Layer 集合频繁引用对象、属性和方法。利用该项目是默认方法这一事实，可以缩减编写的代码。

例如，下列各行代码将做出同样的分配。

```
Map1.Layers.Item(3)Visible = False
Map1.Layers(3).Visible = False
Map1.Layers.Item("ighways").Visible = False
Map1.Layers("ighways").Visible = False
```

获取集合的地理范围

Bound 属性返回一个 Rectangle 对象，代表集合中的所有图层的地理范围 （UserDraw 图层除外）。如果要缩小地图以显示所有图层上的全部对象，此属性非常实用：

```
' 这将有效设置地图绑定到集合中的所有图层的地理范围，令整个地图均位于视野之内。
    On Error Resume Next
    Set Map1.Bounds = Map1.Layers.Bounds
End Sub
```

Layer 集合的某些方法

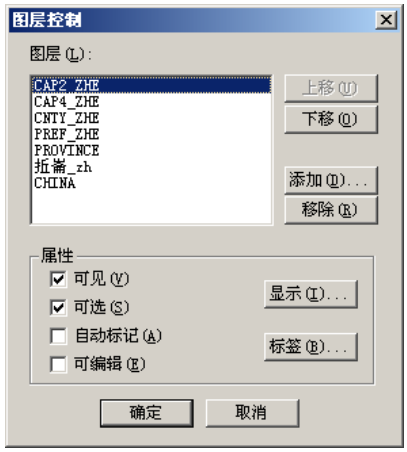
Layer 集合具有若干种方法，用以控制集合中有哪些图层，以及其显示方式。此外还有一种方法，允许用户调用 “图层控制” 对话框，以便控制图层的方法和图元。该方法还可在设计阶段使用 “属性页” 设置，正如本章此前所述。

注：有关 **Layer 集合**方法和属性的完整列表，请参阅第 340 页的 “Layer 对象和 Layers 集合”。

使用 “图层控制” 对话框

LayersDlg 方法提供了一个对话框，用户在其中可以添加图层、移除图层、更改图层顺序和更改图层属性。

“图层控制” 对话框



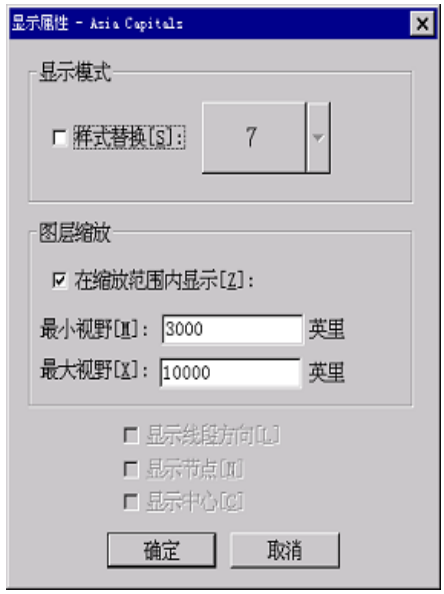
如果用户单击 “确定”，则在对话框中所做的更改将立即应用于地图。

```
Map1.Layers.LayersDlg
```

“图层控制”对话框的显示选项

使用 “显示选项”对话框，可以定制 “地图”窗口中各个图层的显示方式。在 “图层控制”对话框中，用户可以选择图层，单击 “显示”按钮，打开 “显示属性”对话框。用户在此可以更改该图层的默认样式，设置图层显示时的缩放级别，例如：

“图层选项”对话框



显示选项

当用户初次打开一个地图时，边界、直线、点和文本均使用地图 Geoset 文件中的默认设置来显示。用户可以使用 “显示选项”对话框的 “显示模式”来更改对象显示的方式。

例如，用户需要将街道的显示更改为红色虚线。在 “图层控制”对话框中，用户需要选择街道图层，然后单击 “显示”按钮。这将打开 “显示选项”对话框。用户将选中 “样式覆盖”复选框，激活 “样式覆盖”按钮 （较大的灰色按钮）。 MapX 显示适用于图层中对象类型的覆盖按钮。例如，如果图层中包含街道，则将会显示线条样式覆盖按钮。单击该按钮，访问 “线条样式”对话框，用户在其中可以更改街道的宽度、样式和颜色。

对于边界图层，样式覆盖按钮将打开 “区域样式”对话框，用户在其中可以更边界的填充项和边框。“符号样式”对话框显示用户何时需要覆盖包含符号或点的图层的样式。“样式覆盖”只在当前工作会话中有效，正如其他显示设置一样。要令其永久化，您将需要修改 geoset。

使用 LayerInfo 对象创建图层

LayerInfo 对象用于向地图添加新的或现有的图层。该对象具有两个属性和一个方法。 Type 属性用于指定要添加到地图的图层类型，其值设置为 LayerInfoTypeConstants 之一。取决于 LayerInfo 类型，您必须通过 AddParameter 方法来指定特定的参数集。有关 LayerInfo 类型及其必要参数的完整列表，请参阅第 373 页的 “LayerInfo 对象”。在设置之后，属性可以在查看或更改时通过 Parameter 属性更改。

创建图层

您可以创建新的 MapInfo 表并将其添加到地图。这些表既可以是永久的 (miLayerInfoTypeNewTable)，即在硬盘驱动器上创建图层，并在应用程序运行之后仍然保留；也可以是临时的 (miLayerInfoTypeTemp)，即在内存中创建图层，在程序运行之后不再保留。以下示例显示了如何创建新的永久图层和临时图层。

```
' 创建新的永久图层，包含在 USA 图层中选择的
' 所有图元
Dim li as New MapXLib.LayerInfo
Dim ftrs as MapXLib.Features
Dim flds as MapXLib.Fields
Dim ds as MapXLib.Dataset

' 新表将和 USA 图层具有相同的列结构
Set ds = Map1.Datasets.Add (miDatasetLayer, Map1.Layers("USA"))
Set flds = ds.Fields

' 新表将包括 USA 图层中当前所选的全部
' 图元
Set ftrs = Map1.Layers.Item("USA").Selection.Clone

li.Type = miLayerInfoTypeNewTable
li.AddParameter "Name", "USA Selections"
li.AddParameter "FileSpec", App.Path & "\USA_Selections.tab"
li.AddParameter "Fields", flds
li.AddParameter "Features", ftrs
li.AddParameter "OverwriteFile", "1"
' 该图层在添加到图层集合时创建
Map1.Layers.Add li


' 创建新的临时图层，包含 USA 图层中选择的所有
' 图元
Dim li as New MapXLib.LayerInfo
Dim ftrs as MapXLib.Features
Dim flds as MapXLib.Fields
Dim ds as MapXLib.Dataset

' 新表和 USA 图层具有相同的列结构
Set ds = Map1.Datasets.Add (miDatasetLayer, Map1.Layers("USA"))
```

```
Set flds = ds.Fields

' 新表将包括 USA 图层中当前所选的全部
' 图元
Set ftrs = Map1.Layers.Item("USA").Selection.Clone

li.Type = miLayerInfoTypeTemp
li.AddParameter "Name", "USA Selections"
li.AddParameter "Fields", flds
li.AddParameter "Features", ftrs
' 该图层在添加到图层集合时创建
Map1.Layers.Add li
```

添加现有图层

LayerInfo 对象还可用于向地图添加现有图层。使用 LayerInfoTypeTab 时， FileSpec 是需要指定的唯一参数，即要添加到地图的 MapInfo 表的物理位置。此处有若干其他可选参数，允许您设置图层的可见性（Visible 参数）或从新添加的图层创建数据集 （AutoCreateDataset 参数和 DatasetName 参数）。以下是使用这些可选参数的示例，它向地图添加了 USA 图层，确保了图层不可见并且从图层创建了数据集。

```
Dim li as MapXLib.LayerInfo
Set li = CreateObject ("MapXMobile.LayerInfo.5")

li.Type = miLayerInfoTypeTab
li.AddParameter "FileSpec", "\My Documents\Maps\USA.tab"
li.AddParameter "Visible", False
li.AddParameter "AutoCreateDataset", True
li.AddParameter "DatasetName", "dsUSA"

Map1.Layers.Add li
```

移除图层

Remove 方法从地图移除指定的图层。

```
Map1.Layers.Remove 3
```

移除所有图层

RemoveAll 方法从地图移除所有的图层。

```
Private Sub btnLayersRemoveAll_Click()
    Dim nLayers As Integer
    nLayers = Map1.Layers.Count
    ' remove all of the layers
    Map1.Layers.RemoveAll
    nLayers = Map1.Layers.Count
```

图层重新定位

Move 方法在 Layer 集合中重新定位图层。第一个参数是 From 位置 （顶级图层 = 1)，第二个参数是 To 位置。

```
Map1.Layers.Move 1,2
```

Layer 对象

Layer 对象代表采用地图图元形式的矢量制图数据，具有主要的图元类型，例如区域、直线或符号。通常 Layer 对象相应于源自 MapInfo 表的地理对象。 Layer 集合中的每个 Layer 对象的行为相互独立。其样式和其缩放图层可以基于单独的基础更改，图层之间互不影响。

您可以在程序之内随时控制图层对象。您还可以使用 “属性页” 来在设计阶段设置图层属性，如上所述。 Layer 对象具有众多属性和方法。**“图元和选择”**提供了详细说明，介绍用于从图层获取图元的方法。

某些 Layer 属性

注：有关 Layer 对象方法和属性的完整列表，请参阅第 340 页的 “Layer 对象和 Layers 集合”。

AutoLabel

控制图层是否自动标记。**“States”**

```
Map1.Layers(“tates”).AutoLabel = True
```

Name

图层的名称。

```
MsgBox Map1.Layers(4).Name
```

OverrideStyle

是否覆盖此图层的默认显示特征。

```
Map1.Layers(9).OverrideStyle = True
```

Selectable

将 MapX 设置为识别用户何时单击此图层。

```
Map1.Layers(”tates”).Selectable = True
```

Style

OverrideStyle 为 True 时要使用的图层样式。

```
Map1.Layers(9).Style = newstyleobject
```

Visible

图层是否可见。

```
Map1.Layers.Item(2).Visible = False
```

ZoomLayer

将缩放图层设置为开或闭。

```
Map1.Layers("tates").ZoomLayer = True
```

ZoomMax

设置图层可见的最大缩放级别。

```
Map1.Layers("tates").ZoomMin = 580
```

ZoomMin

设置图层可见的最小缩放图层。

```
Map1.Layers("tates").ZoomMin = 45
```

图层顺序

Layer 集合中的地图图层按照增序索引显示 （例如 Layers(1) 是顶层， Layers(2) 是 Layer(1) 之下的图层），底层最先绘制，顶层最后绘制。正确设定图层顺序非常重要。

例如，假定有一个客户点的图层和一个人口地域图层。如果图层在 Layer 集合上的顺序不正确， MapX 将先绘制客户点，然后再显示人口地域图层。您的点将由于人口地域图层而变得模糊。

在地图的设计阶段可以重新确定图层的顺序。从 “属性页”的 “图层”选项卡，选择要重新排序的图层，然后选择 “上移”或 “下移”按钮，将图层移动到其当前位置之上或之下的位置。

要允许用户在运行时重新设定图层顺序，可使用 LayersDlg 方法来表示 “图层控制”对话框，正如其他示例所述。

图层顺序在您使用选择工具时也很重要。选择工具从最顶部的可选择图层选择对象。如果在相同的位置上有若干个对象，则难于选择所需的确切对象。您还可以重新给图层排序，以便要选择的图层是新置于最顶层的图层。

检查图层

在集合中有许多需要检查图层的实例。以下简单的代码片断通过 Layer 集合迭代，确定展示的图层类型，并在对话框中显示消息表示图层类型。

```
Dim lyr as Layer
For Each lyr in Map1.Layers
Select Case lyr.Type
        Case miLayerTypeNormal
            MsgBox  "Layer " & lyr.Name & " is a normal layer"
        Case miLayerTypeRaster
            MsgBox  "Layer " & lyr.Name & " is a raster layer"
        Case miLayerTypeSeamless
            MsgBox  "Layer " & lyr.Name & " is a seamless layer"
        Case miLayerTypeUnknown
            MsgBox  "Layer " & lyr.Name & " is an unknown layer"
        Case miLayerTypeUserDraw
```

```
            MsgBox  "Layer " & lyr.Name & " is a user draw layer"
        Case miLayerTypeDrilldown
            MsgBox  "Layer " & lyr.Name & " is a drilldown layer"
Next
```

在该代码片断中，我们将使用 MapX 定义的常数用于所有图层类型。这些常数统称 LayerTypeConstants。

检查图元类型

此代码片断通过 Layer 集合按照索引来迭代。它将检查集合中的每个 Layer 对象，确定图层中出现的图元类型。

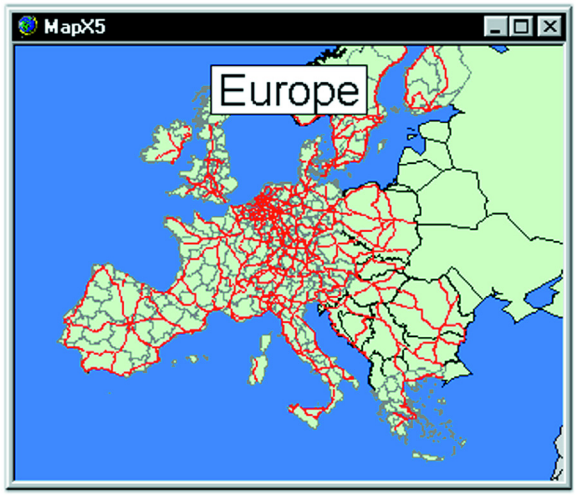
```
Dim i as Integer
For i = 1 To Map1.Layers.Count
   Select Case Map1.Layers(i).PredominantFeatureType
      Case miFeatureTypeRegion
         MsgBox "Layer " & lyr.Name & " contains regions"
      Case miFeatureTypeLine
         MsgBox "Layer " & lyr.Name & " contains lines"
      Case miFeatureTypeSymbol
         MsgBox "Layer " & lyr.Name & " contains symbols"
      Case miFeatureTypeUnknown
         MsgBox "Layer " & lyr.Name & " contains unknown _
         features"
      Case miFeatureTypeText
         MsgBox "Layer " & lyr.Name & "contains text features"
   End Select
Next
```

Layer 集合具有一个基础索引，以便我们可从一个索引值开始按照 Layer 集合计数来进行迭代。我们还评估集合中每个图层的 PredominantFeatureType 属性，使用 FeatureType 常数来评估 PredominantFeatureType。

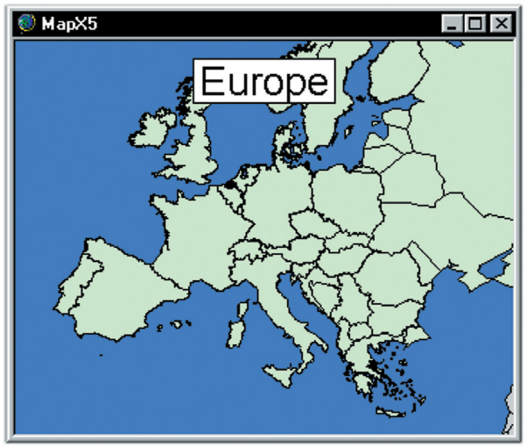
缩放图层

有时需要只显示特定缩放级别的图层。缩放图层允许您在地图的缩放级别处于预设距离中时查看图层。您可以为每个图层设置不同的缩放图层级别。

例如，如果地图中包括街道图层，您可能会在用户缩小的倍数过大时发现街道变得较为模糊。



使用缩放图层，您可以设置地图，以便 MapX 在用户缩小显示大于 5 英里的区域时自动隐藏街道。



以下示例代码向使用 Layer 集合的地图添加了图层。通过修改 Layer 对象属性来添加方法和设置缩放图层。

```
Dim lyrStreets As Layer       `Creates the layer object
`Sets the Streets table as the layer object and orders the layer in `the
map as number 3.
Set lyrStreets = Map1.Layers.Add("treets.tab", 3)
lyrStreets.ZoomLayer = True         `Sets zoom layering to true
lyrStreets.ZoomMin = 0             `Sets minimum zoom to 0 miles
lyrStreets.ZoomMax = 5           `Sets maximum zoom to 5 miles
```

在设置某一图层的缩放图层之后，当您在缩放级别上、下限进行缩放时，图层将显示在地图之上。您还可以使用缩放工具或地图对象的 ZoomTo 方法来地图上进行放大。使用 ZoomTo 方法，您可以指定缩放级别以及 x 和 y 坐标来确定地图中心。

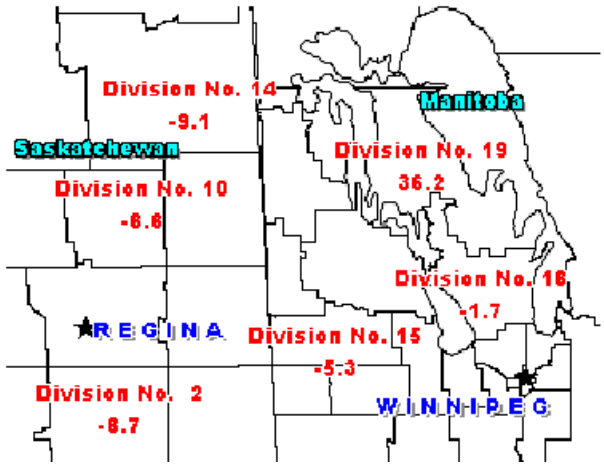
```
Map1.ZoomTo 3, -70.26, 44.05
```

由于 3 英里的缩放值在缩放级别的限定值之内，因此 Streets 图层将以 (-70.26, 44.05) 的点为中心，然后在地图控制的宽度之内显示 3 英里的地图。

相同 “地图”窗口中的不同图层可以在不同的缩放级别上显示。例如，您有一个街道图层、一个城镇边界图层和一个州界图层。您希望街道图层只在缩放级别小于 8 英里时可见。同时希望城镇边界只在缩放级别介于 20 和 200 英里之间时显示。此外还希望州界图层只在缩放级别大于 100 英里时可见。您可以为地图中的每个图层设置不同的缩放级别。

生成用于图层的标签

MapX 为标记图层中地理对象的属性提供了众多途径。其绘图位置基于地理对象的质心位置以及其他附加信息，例如锚点和偏移量。



和属性一样，标签**动态**连接到其地图对象。如果图层关闭或不可见，则标签将不再显示。如果数据或地理信息更改，则标签也将更改。如果创建用于标签的表达式并且更改了表达式，则当前的标签将会动态替换为新标签。

无论您是自动标记地图，还是交互使用标签工具或 LabelAtPoint 方法，标签的内容都将由与地理对象关联的数据确定。

在标签内容之外，可以使用 LabelProperties 对象中的属性来控制位置、显示以及自动标签的外观。您可以设置用于显示标签的条件、显示样式以及图层中所有对象的显示位置。

您还可以通过 Label 集合来控制单独标签的外观。此处有和各个图层相关联的 Label 集合，集合中的每个项目都是一个 Layer 对象。通过控制 Label 对象的属性，您可以从该图层的其余标签来独立控制标签的显示。

控制标签显示

要自动生成图层标签，可将图层的 **AutoLabel** 属性设置为 True。

每个 Layer 对象都有一个 **LabelProperties** 对象，控制标签的方方面面。例如，要隐藏所有图层标签，可将 **LabelProperties.Visible** 属性设置为 False。要指定希望在地图上显示的最大标签数，可设置 **LabelMax** 属性。**重复** 属性控制具有相同名称的图元是否可以在地图上同时具有单独的标签。 **Overlap** 属性控制标签是否允许重叠，将 Overlap 设置为 True 将会标记更多的图元，但是您可能会发现覆盖的标签难于辨识。

缩放图层标签

您可以配置标签只在特定缩放范围内显示，这和在特定缩放范围中显示图层类似。要指定标签的缩放范围，可以设置图层的 LabelZoom、 LabelZoomMax 和 LabelZoomMin 和 LabelProperties 属性。

标签位置

要控制标签位置，可设置 **Position** 属性 （控制标签位置在图元的上、下或质心）和 **Offset** 属性（控制标签偏移图元的距离）。

默认的锚点取决于图层的主要图元类型。例如，区域图元的图层默认为将标签居中置于区域质心之上。

对于直线 / 多边形图元 （例如街道地图），您可以令标签和直线图元并行 – 即设置 **Parallel** 属性为 True。

创建标注

标注是带有指向标记对象直线的标签。在相对较小的区域内，如果有众多标记，则这些标注将会非常实用。例如亚洲地图的标记。图中有许多小国家，并且相互相对较近。如果尝试标记所有国家，标签将会重叠并将难以辨识。

要显示标注线，将要标记的图层的 **LabelProperties.LineType** 属性设置为 miLineTypeSimple (1) 或 miLineTypeArrow (2)。

标签样式

要对所有标签做出样式更改，可设置 **Style** 属性。

您还可以设置适当的样式选项。做出所需的样式更改。在返回到地图之后，所选标签将和指定的样式更改一起显示。

此外还有背景选项。将适当选项设置为没有背景、光晕来创建文本周围的光晕效果。这一设置将文本置于从它所覆盖的地貌上凸起的部位中 （例如部分地区或街道等等）。

交互式标记

尽管您可能自动完成大部分标记，但在某些情况下，还需要使用标签工具一次创建多个标签。标签工具是 MapX 中内置的标准工具之一。要激活该工具，可将 **Map.CurrentTool** 属性设置为 miLabelTool (1010)，用户将可以通过单击图元来标记地图图元。

使用可见性设置是移除图层中的所有标记最为轻松的途径。它将禁用该图层上的所有标签显示，其中包括自动和定制标签。要只清除定制标签 （使用标签工具放置的标签），可使用 **ClearCustomLabels** 方法。

标签可以通过使用选择工具 (miSelectTool 1007) 选择一个或多个标签来交互移动，然后将标签拖放到新的位置。已经手动编辑的标签随后将变为该图层的 Label 集合中的项目。随后可以通过设置 Label 对象的属性，以编程方式来进行编辑。

注释

尽管标签标记功能可以满足大部分文本需求，但是您可能还需要创建文本对象来提供地图注释。与标签不同的是，文本标记没有连接数据 – 您可以将注释置于地图上的任意位置，即使该位置没有图元。

要向地图添加文本或符号注释，可使用 Annotation 集合中的方法和属性以及对象，或使用符号和文本工具添加文本或符号，即 miSymbolTool 1005 和 miTextTool 1006。

每个地图都有一个 Annotations (Map.Annotations 属性) 集合。注释是符号或文本对象，并且绘制在地图顶层。

注释通常用于向地图添加文本消息，或在地图上放置符号。这些注释随地图的大小缩放。注释没有绑定到特定的图层。它们始终显示在最上面。

注意 Annotation 对象没有用于设置位置、符号样式或文本的属性。要控制注释的特性，您必须使用 Annotation.Graphic 属性来获取 Graphic 对象，然后修改 Graphic 对象。

光栅图像

光栅图像是一种计算机图片，其中包含一行一行的小点 （像素）。光栅图像统称有时统称位图。航空图片和卫星图像都是 GIS 中常用的光栅数据类型。

MapX 中支持的光栅图像格式如下：

- TIFF (*.tif)
- MrSID (*.sid)
- ECW (*.ecw)
- Spot (*.bil)
- Jpeg (*.jpg)
- PCX (*.pcx)
- GIF (*.gif)
- Windows 位图 (*.bmp)
- PNG (*.png)
- Photoshop (*.psd)
- Targa (*.tga)

- Windows 元文件 (*.wmf)
- JPEG200 (*.jp2)

将光栅图像显示为图层

使用 MapX 可以显示光栅图像 （位图）作为您所创建的地图的背景。随后您可以在图像之上叠加数据，例如街道地图和客户位置。尽管光栅图像可以是地图的图片，但是该图像不能具有附属到其的数据。该图像作为背景或 “铺垫”供用户查看。

如何显示光栅图像

光栅图像可以自行注册 （GeoTiff 图像），相应图像包含地理信息，可以令 MapX 知道在何处显示它们；或者它们可以具有关联的 .TAB 文件，以告知 MapX 在何处显示图像。 .TAB 文件使用 MapInfo 专业版创建和注册该图像。自行注册光栅将添加到使用 miLayerInfoTypeRaster 类型的 LayerInfo 对象的地图。带有关联 tab 文件的光栅图像将添加到使用 miLayerInfoType 类型的 LayerInfo 对象的地图。

注：当将光栅图像显示为图层时， MapX 将自动设置所有矢量图层的旋转和投影，以便它们可以匹配光栅图像的旋转和投影。

动画图层

动画图层在地图图元需要频繁更新的地方尤为实用，例如实时的应用程序。例如，您可以开发车队管理应用程序，在其中将每个车辆表示为一个点对象。您可以使用全球定位卫星 (Global Positioning Satellite, GPS) 技术接收当前的车辆坐标，然后更新点对象，在地图上显示当前的车辆位置。在此类型的应用程序中，地图对象不断变化，如果正在更新的对象存储在动画图层中而不是常规图层中，地图的重新绘制将会更加迅速。

AnimationLayer 最初设置为空。您可以将 Layer 对象分配给属性，令该图层称为动画图层 （可以是常规图层或用户绘图图层）。将图层分配给 AnimationLayer 属性时，它将在所有图层顶部绘制，包括注释图

层和选择。该图层还处于 Layer 集合的相同位置上。浮动对象，类似图例，将仍然显示在动画图层的顶部，尽管它们由于已经剪切而无需每次重新绘制。如果正常的图层用作动画图层，选择和标记仍然有效。

示例

```
Set Map1.Layers.AnimationLayer = Map1.Layers(3)
```

该属性可用于确定当前哪一图层是动画图层 （如有）：

```
for each lyr in Map1.Layers
if Map1.Layers.AnimationLayer = lyr then ...
end if
next
```

要关闭动画图层，可以将该图层指定为空：

```
Set Map1.Layers.AnimationLayer = nothing
```

绘制图层

使用 miLayerInfoTypeUserDraw 类型的 LayerInfo 对象的 Layers.Add，开发人员可以向地图添加 UserDraw 图层。 UserDraw 图层允许您在独立于任何地图图元的图层中绘制对象。 UserDrawLayers 和 **DrawUserLayer** 事件一起联合使用，后者在需要绘制图层时触发。用户绘制图层的数量不限。

工作原理

首先，向用户绘制图层添加图层集合：

```
' 将 UserDraw 图层设置为 My Layer
Dim li as MapXLib.LayerInfo
Dim lyr as MapXLib.Layer

Set li = CreateObject("MapXMobile.LayerInfo.5")
li.Type = miLayerInfoTypeUserDraw
li.AddParameter "Name", "UserDraw"

Set lyr = Map1.Layers.Add (li, 1)
```

随后将使用该代码在 DrawUser 图层事件中的图层上进行绘制。当

应用程序创建使用 Layer 集合的 Add 方法的 UserDraw 图层时，

事件在窗口需要更新时针对应用程序触发。

示例

完整的示例如下所示。

```
' API DEFS 应该在单独模块中声明
Declare Function MoveToEx Lib "gdi32" Alias "MoveToEx" (ByVal hdc As
Long, ByVal x As Long, ByVal y As Long, lpPoint As POINTAPI) As Long
```

```
Declare Function LineTo Lib "gdi32" Alias "LineTo" (ByVal hdc As Long,
ByVal x As Long, ByVal y As Long) As Long
Declare Function SetMapMode Lib "gdi32" Alias "SetMapMode" (ByVal hdc As
Long, ByVal nMapMode As Long) As Long
Type POINTAPI
        x As Long
        y As Long
End Type
Public Const MM_TWIPS = 6

' this sets the UserDraw Layer to “My Layer”
Dim lyr as Layer
Set lyr = Map1.Layers.AddUserDrawLayer("My Layer", 1)

' 此示例在怀俄明角之间绘制直线
Private Sub Map1_DrawUserLayer(ByVal Layer As Object, ByVal hDC As Long,
ByVal RectFull As Object, ByVal RectInvalid As Object)
Dim pt As POINTAPI

SetMapMode hDC, MM_TWIPS
dim PX as single
dim PY as single

X1 = -111.0542
Y1 = 45.0009
X2 = -104.0528
Y2 = 41.0018

if map1.ClipLine(X1,Y1,X2,Y2) then
   map1.ConvertCoord(PX, PY, X1,Y1, miMapToScreen)
   MoveToEx hDC     , PX, -PY, pt    ' win api call
   map1.ConvertCoord(PX, PY, X2,Y2, miMapToScreen)
   LineTo hDC, PX, -PY           ' win api call
end if

End Sub
```

6 章

- 什么是数据绑定?
- 将数据加入地图的作用
- 如何将数据加入地图
- 使用 Dataset 对象
- 使用 Datasets 集合
- 使用 Datasets.Add 方法
- 使用 Fields 集合
- 使用 Fields.Add 方法
- 将数据显示为点图层 (BindLayer)
- 使新的点图层成为永久图层
- 数据绑定如何使用 GeoDictionary
- 不同类型的数据源
- XML 支持

将数据加入地图

通过数据集可以将用户数据绑定到地图上。例如，如果有 Microsoft Access 版按市 （县）划分的销售数据库以及 Oracle 版销售机构所在地数据库，可以将该数据绑定到地图上，观察趋势或注意两组数据之间的关系。

什么是数据绑定?

数据绑定是将数据源中的数据导入 MapX 的过程。

当今的商业中有许多不同类型的数据库；因此， MapX 使您可以绑定到多种不同类型的数据源。在 MapX 中，数据显示为 Dataset 对象。 Datasets.Add 方法的第一个参数可以指定 DatasetTypeConstants 值，该值指定要进行绑定的数据类型。可以绑定到的数据源类型包括：

数据源	说明
ADO	该类型的数据绑定使用 MS Active 数据对象 ADO Recordset。
DAO	DAO Recordset 对象。该对象可以从 Visual Basic 数据控件、 Access 窗体获得，也可以使用 Visual Basic、 Access 或 C++ 中创建。
Delphi	该类型使用 Borland BDE 数据源。
全局句柄	使用该类型的数据绑定可以导入使用制表符分隔的数据块。
图层	通过该类型的数据绑定可以创建使用 MapInfo 表中的字段的数据集。
Notes 视图 /Notes 查询	这两种类型的数据绑定专门针对 Lotus Notes。
OCI	MapX 可以使用 OCI 从 Oracle 数据源中检索数据。
ODBC	MapX 可以使用 ODBC 从 ODBC 数据源中检索数据。
OLE 数据	用于 PowerBuilder 等容器。
Oracle Express 对象	以便作为数据集访问数据立方。
RDO	使用 MS 远程数据对象和 RDO Resultset 对象。
SafeArray	可以静态绑定安全数组中的数据的 COM 数据集。
不绑定	如果无法支持某种其它格式， MapX 提供了一个 “后门”。使用这种类型的数据绑定，您可以建立一个事件循环， MapX 会询问容器数据值，一次一个单元格。

这些用户创建的自定义数据绑定类型在本章后文详细介绍。

将数据加入地图的作用

将自己的数据绑定到地图上主要有以下两个优点：

- 可以将数据作为地图上的图元查看。假如您有 Microsoft Access 版的美国销售机构所作地表。通过数据绑定，可以使用 Datasets.Add 方法将表作为数据集添加，并使用 BindLayer 对象作为方法的一个参数。这样可以在地图上创建一个新图层，并将每个销售地点显示为美国地图上的一个点。将数据加入地图后，即可使用 MapX 轻松地创建应用程序，用于查找与客户输入的地址距离最近的销售地点。如何查找地图上的图元在第 125 页的第 9 章：“查找地图上的图元”中介绍。

- 可以将属性数据绑定到地图上，然后使用该绑定创建基于数据的主题阴影地图。假设您的 Microsoft Access 版销售机构所在地表还包含每个地点总销售量的列。使用 Datasets.Add 方法可以将包含总销售量的列加入美国地图。将数据加入地图后，即可创建按州划分的主题阴影销售量。主题地图在第 111 页的第 8 章：“主题地图绘制和分析”中介绍。

如何将数据加入地图

使用 Datasets.Add 方法通过程序绑定数据。使用该方法，可以通知 MapX 要使用的数据源、数据源的某些信息以及要绑定到的地图图层。另外， Add 方法不必明确指定数据的自身属性即可绑定数据；此时，MapX 会分析数据并自动确定将数据绑定到地图上的最佳方式。

数据绑定可以通过以下两种方法完成：

- 如果使用 Visual Basic 开发绑定数据控件，在设计阶段，可以使用 Map 对象的 Dataset 属性。

  - 或者 -

- 使用 Datasets.Add 方法通过程序绑定数据。使用该方法，可以通知 MapX 要使用的数据源、数据源的某些信息以及要绑定到的地图图层。另外，Add 方法不必明确指定数据的自身属性即可绑定数据；此时， MapX 会分析数据并自动确定将数据绑定到地图上的最佳方式。

绑定过程会创建 Dataset 对象。该 Dataset 加入 Datasets 集合，包含数据绑定到的地图图层中的图元的计算值。例如，如果数据绑定到美国地图上，每个州将有一个新数据值，用于控制州的绘制方式。如果数据源中包含某个州的多个记录，值将求和、求平均值或计数。数据集具有 Value 属性，可以使用该属性访问地图每一行 （即图元）的计算值。

对于大多数类型的数据绑定， DataSource （Dataset 的第二个参数）实际上是 OLE 接口。 MapX 使用该接口直接从数据源中访问数据。数据实际上不传递给 Datasets.Add。

使用 Dataset 对象

Dataset 对象是将数据源中的数据绑定到 MapX 上的结果。

数据集包含将数据绑定到的地图图层中的图元的计算值。数据源是 DAO Recordset、 ODBC 数据源、全局句柄 （导入制表符分隔的数据块的方法）、 OLE 数据 （用于 PowerBuilder 等容器）和不绑定数据（MapX 询问容器数据值，一次一个单元格）。

使用 Datasets 集合

Datasets 集合是代表地图的所有 Dataset 的对象。 Datasets 集合包含用于添加、删除或处理集合中现有 Dataset 对象的方法和属性。

在指定的集合中添加或删除 Dataset 对象可以使用下列方法完成：

方法	说明	代码示例
添加	创建指定的数据集并将其加入集合。	Set ds = Map1.Datasets.Add_ (miDatasetADO, rs)
删除	从 Datasets 集合中删除指定的 Dataset 对象。	Map1.Datasets.Remove 2

使用 Datasets.Add 方法

使用 Datasets.Add 方法，可以将数据源中的数据绑定到 MapX 上。这样可以将外部数据源中的数据绑定到地图上。使用 Datasets.Add 方法可以指定特定的数据集并将其加入 Datasets 集合。让我们了解一下 Datasets.Add 方法的语法。

注：可选参数在 [方括号] 中。
Datasets.Add Type, SourceData, [Name], [Geofield], [SecondaryGeofield], [BindLayer], [Fields], [Dynamic]

Type 参数

该参数是添加的数据集类型。该参数接受 DatasetTypeConstants 值。

DatasetType 常数

类型	说明
miDatasetADO	ADO
miDatasetDAO	数据访问对象
miDatasetDelphi	Borland Delphi Native
miDatasetDelphi4	Borland Delphi4
miDatasetDelphi5	Borland Delphi5
miDatasetGlobalHandle	以制表符分隔的数据
miDatasetLayer	MapInfo 表
miDatasetNotesQuery	Lotus Notes 查询
miDatasetNotesView	Lotus Notes 视图
miDatasetOCI	Oracle 调用接口数据源
miDatasetODBC	ODBC 数据库
miDatasetOLEData	OLE 数据源
miDatasetRDO	RDO
miDatasetSafeArray	安全数组
miDatasetUnbound	MapX 向容器请求数据。

SourceData 参数

该参数引用数据，不同数据集类型会有所不同。对于大多数类型的数据绑定， DataSource（Datasets.Add 的第二个参数）实际上是 OLE 接口。 MapX 使用该接口直接从数据源中访问数据。数据实际上不传递给 Datasets.Add。

以下是每种数据集类型的有效数据源：

数据集类型	有效的源数据
miDatasetADO	ADO 表
miDatasetDAO	DAO Recordset 对象。
miDatasetDelphi	Delphi Native
miDatasetDelphi4	Delphi4 记录集
miDatasetDelphi5	Delphi5 记录集
miDatasetGlobalHandle	VT_I4 类型的变量，并且 lVal 等于全局内存句柄。
miDatasetLayer	MapInfo 表。

数据集类型	有效的源数据
miDatasetNotesQuery	Lotus Notes 查询
miDatasetNotesView	Lotus Notes 视图
miDatasetOCI	OCIQueryInfo 对象
miDatasetODBC	ODBCQueryInfo 对象
miDatasetOLEData	Datasets.Add 忽略。
miDatasetRDO	RDO 表
miDatasetSafeArray	安全数组
miDatasetUnbound	无 （而是使用 RequestData 事件访问只有编程人员知道格式的数据）。

Name 参数

该参数是唯一标识数据集的字符串。该参数是可选参数，如果未指定，将使用 DatasetN 格式的名称，其中 N 是唯一编号。

Geofield 参数

该参数是数据源中包含地理信息的列的名称或索引。如果未指定该参数， MapX 将搜索各字段，确定数据源中包含 GeoDictionary 中指定的地理信息的列。该列不必是地理数据字段，因为可以是唯一关键字列 （例如邮政编码）。不过，如果您知道数据源中包含地理信息的列，则应当指定。

如果要将数据作为地图上的图元查看，数据源中的 *GeoField* 列必须是唯一的。 *GeoField* 列将用于为新的点图层中的图元命名。如果值不是唯一的，将导致单个点加入新的点图层，从而出现第一个重复的关键字值，重复行中的数据值将合计。

如果指定了 Fields 集合， *Geofield* 参数引用 Fields 集合中的列，而不是源数据中的列。

Secondary Geofield 参数

只有数据集绑定到的图层包含的关键字列不唯一时，才需要该参数。例如，使用 MapInfo 表 “United States Counties”绑定数据，因为市 （县）名对一个州不是唯一的，所以需要 Secondary Geofield 参数。表中的多个州有名为 “Warren”或 “Washington”的市 （县）。因此，需要更多信息来解决数据绑定过程中潜在的不确定性。什么时候要将数据绑定到 New York 的 Warren，什么时候要将数据绑定到 New Jersey 的 Warren？通过将 “County”列指定为 Geofield，将 “State”列指定为 Secondary Geofield，MapX 可以通过重新定义数据绑定到每个州的市 （县）的过程，确定哪些数据绑定到 New York 的 Warren，哪些数据绑定到 New Jersey 的 Warren。这个概念与创建 Find 对象时重新定义边界相同。有关详细信息，请参见 **“查找地图上的图元”**。

如果指定了 Fields 集合， *SecondaryGeofield* 参数引用 Fields 集合中的列，而不是源数据中的列。

BindLayer 参数

该参数指定要将数据连接到的地图图层 （如果要绑定属性数据）或 BindLayer 对象 （如果点参考文件要地理引用导入数据 （例如邮政编码）或包含 Long/Lat 值）。该参数是可选参数，如果未指定， MapX 将在 GeoDictionary 中搜索要连接到的图层。

如果您知道应绑定到的地图图层，考虑到性能原因，应指定该图层。在进行 BindLayer 匹配时，Geofields 必须是唯一的。对于不唯一的数据集，只有第一项将匹配。其余项将忽略。 BindLayer 在本章后文详细介绍。

注：如果在 DataSets.Add 中使用 miBindLayerTypeXY，将创建一个临时表，其中包含一个名为 “GeoName”的列，以便必须在永久图层上创建要成为新数据绑定的 Fields 集合。

Fields 参数

该参数是属于 Field 对象集合的 Field 对象。 Field 对象用于描述数据源中要导入的字段，以及在数据源的多个记录与特定的地图图元匹配时，要使用的合计函数。这将建立一个要绑定到地图上的字段 （列）的集合。该参数是可选参数，如果未指定，所有列均将导入，如果每个图元遇到多个记录，则对数据值求和。 Fields 集合在本章后文详细介绍。

如果指定了 Fields 集合， *Geofield* 和 *SecondaryGeofield* 参数引用 Fields 集合中的列，而不是源数据中的列。

Dynamic 参数

该参数是控制数据绑定是否为动态的布尔值。该参数是可选参数；如果省略，则默认值为 False，表示绑定是静态的 （即 MapX 将在数据库打开时复制所需的数据）。如果指定 True， MapX 将仅在需要数据时 （例如标记时）实时访问数据。如果指定 True 但是数据集不支持动态列，将发生例外。

使用 Fields 集合

数据源可能包含许多列数据。 MapX 对绑定的每个数据列均需要开销，所以，应只绑定地图中需要的数据 （例如要主题映射或标记的数据）。使用 Datasets.Add 的 Fields 参数建立要绑定到地图上的字段 （列）的 Fields 集合。

注：字段变量应声明为 “MapXLib.Fields”。这样可以防止与 DAO “Fields”对象发生冲突。

```
Dim flds As New MapXLib.Fields
```

可以通过 Dataset.Fields 属性访问数据集的 Fields 集合。

方法	说明	代码示例
添加	将字段加入 Fields 集合。	flds.Add “Sales”, “Sum_of_Sales” _ , miAggregationSum
删除	将字段从 Fields 集合中删除。	flds.Remove 3
RemoveAll	将所有 Field 对象从集合中删除。	flds.RemoveAll

使用 Fields.Add 方法

使用 Fields.Add 方法可以将数据源中的数据列作为字段加入 Fields 集合。建立 Fields 集合与 Datasets.Add 方法（或与 Themes.Add 方法；参见第 111 页的第 8 章：“主题地图绘制和分析”）配合使用。 Datasets.Add 的 Fields 参数接受 Fields 集合，使用该 Add 方法建立。数据集创建后，不能对数据集的 Fields 集合使用 Add 方法。

要了解 Fields.Add 方法的详细信息，请参见第 305 页的 “Fields.Add 方法”。

数据合计

Fields.Add 方法的 Aggregate 函数参数确定如果找到多个匹配项， MapX 如何计算字段值。

例如，如果数据类似如下所示：

州	销售额
CA	120
NY	100
CA	50
CA	110

此例在加利福尼亚有三个销售机构，因为只有一个数据值将连接到加利福尼亚，我们需要通知 MapX 对销售额求和、求平均值，或对销售额计数。

Fields.Add 方法的 Aggregate 函数参数默认为 miAggregationIndividual （对于字符串列）和 miAggregationSum （对于数字列）。

以下是用于处理多个匹配项的合计：

Aggregation 函数常数

以下是 Aggregation 函数常数的列表和简要说明。

常数	说明
miAggregationAuto	MapX 根据使用的数据自动选择合计方法。
miAggregationSum	对数据求和。
miAggregationAverage	对数据求平均值。
miAggregationIndividual	导入每个记录。
miAggregationCount	对记录计数。

将数据显示为点图层 (BindLayer)

BindLayer 数据绑定是数据绑定的一种，源数据用于在数据指定的位置创建新的点图层。这种绑定在以下两种情况下发生：

- 数据中包含 x 和 y 坐标时 （例如经度和纬度）。

  点在指定的位置创建。请参见 **“将 X / Y 数据显示为地图上的点图层”**。

- 当数据中包含参考信息 （例如邮政编码）时。

  点在邮政编码所在地 （临近位置）创建。请参见 **“将邮政编码数据显示为地图上的点图层”**。

如果绑定的数据中包含 X/Y 坐标或点信息 （例如邮政编码），而您希望在数据指定的位置看到在新图层或现有图层上创建的点，则需要 BindLayer 对象。

使用 BindLayerObject 添加数据集时，点会自动在地图上显示。不使用 BindLayerObject 添加数据集时则不会。只有 BindLayer 对象是这样。

点将以灰色的星星导入。如果要更改样式，必须将 Layer.OverrideStyle 属性设置为 **True**，并更改样式属性。只有 Layer.OverrideStyle 设置为 True，对样式的更改才会显示。

BindLayer 对象属性

属性	说明	值
LayerType	指定数据绑定到的图层类型。	BindLayerTypeConstant： • MiBindLayerTypeNormal • MiBindLayerTypeXY • MiBindLayerTypePointRef
CoordSys	指定创建图层所在的坐标系。	CoordSys 对象
FileSpec	允许指定文件的名称和位置，使 Datasets.Add 方法可以创建永久图层而不是临时图层。	标识文件路径和文件名的文本字符串。
KeyLength	一个正数，代表生成的图层中字符列的所需长度。	数字值 (1-254)。
LayerName	指定如果 LayerType 为 miBindLayerTypeNormal，要将数据绑定到的图层的名称。如果 LayerType 为 miBindLayerTypeXY 或 miBindLayerTypePointRef，则为新创建的图层的名称。	字符串值。
RefColumn1	包含纬度的字段 （LayerType 为 miBindLayerTypeXY 时）或包含参考值 （例如邮政编码）的字段 （如果 LayerType 为 miBindLayerTypePointRef）。	引用某个列的字符串或整数（基于 1 的索引）。
RefColumn2	指定包含经度的字段 （LayerType 为 miBindLayerTypeXY 时）。	引用某个列的字符串或整数（基于 1 的索引）。
ReferenceLayer	指定要使用的参考文件的名称 （如果 BindLayer.LayerType 为 miBindLayerTypePointRef）。	字符串。

使用 BindLayer 对象的操作顺序如下：

1. 创建 BindLayer 对象。
2. 在添加数据集时，通过将该对象作为 Datasets.Add 方法的 BindLayer 参数传递来使用该对象。

如何创建 BindLayer 对象并显示下面所示的 X / Y 或点参考数据。

将 X / Y 数据显示为地图上的点图层

如果数据中包含 X / Y 坐标，可以通过数据绑定创建并显示点图元图层。使用 BindLayer 对象，并

- 将其 LayerType 属性设置为 miBindLayerTypeXY。

- 将其 RefColumn1 属性设置为包含 X 坐标值的列的名称或索引 （基于 1）。
- 将其 RefColumn2 属性设置为包含 Y 坐标值的列的名称或索引 （基于 1）。

将邮政编码数据显示为地图上的点图层

如果您的数据表中没有 X / Y 坐标，但是包含点参考信息 （例如邮政编码），则仍可以将数据显示为地图上的点。使用点参考绑定过程可以对照包含 X / Y 坐标的独立参考图层为基于邮政编码的数据表进行数据绑定。方法是在建立 BindLayer 对象时指定 BindLayer 对象中的参考图层，如下所示。

MapX 在 \MapInfo\MapX 5.0\Maps 目录中提供名为 Us_zips.tab 的表，可以作为美国邮政编码的参考图层使用。 Us_zips.tab 还具有 “友好名称”（通过 Geodictionary 为图层分配的说明，该名称为 “US 5 Digit ZIP Code Centers”）。

注：在以前版本的 MapX MapInfo 中提供了名为 Zipcodes.cpf 的邮政编码绑定文件。我们不再提供该文件。而是提供 Us_zips.tab 取代 Zipcodes.cpf。

要通过邮政编码数据绑定创建并显示点图元图层，应使用 BindLayer 对象，并：

- 将其 LayerType 属性设置为 miBindLayerTypePointRef。
- 将其 RefColumn1 属性设置为表中包含邮政编码 （或其它点参考）数据的列的名称或索引 （基于 0）。
- 将其 ReferenceLayer 设置为要使用的参考文件的名称。参考文件包含 x 坐标和 y 坐标，数据绑定过程使用这些坐标确定邮政编码数据在地图上的地理位置。 ReferenceLayer 属性可以设置为图层的文件名 （完整文件名说明）或设置为图层的 “友好名称”（通过 Geodictionary 为图层分配的说明）。

注：参考图层不需要在地图中显示，但是必须安装在 Geodictionary 中。

在邮政编码数据绑定中不使用 RefColumn2 属性，因为美国邮政编码是唯一的。

使新的点图层成为永久图层

很容易让使用 BindLayer 对象创建的新点图层成为永久 MapInfo 表。使用 BindLayer.Filespec 属性或指定文件的名称和位置，使 Datasets.Add 方法可以创建永久图层而不是临时图层。如果未指定该属性，图层则为临时图层。该属性适用于 X / Y 数据绑定和点参考 （邮政编码）数据绑定。

以下示例使用 X / Y 坐标通过数据绑定创建一个永久表。

```
Dim BindLayerObject As New MapXLib.BindLayer
Dim ds as Dataset

'Where to save the newly created permanent table
BindLayerObject.FileSpec = “C:\MapInfo\Data\Dealers.TAB”
BindLayerObject.LayerType = miBindLayerTypeXY ‘Binding to Long/Lat
BindLayerObject.LayerName = "US Dealers" 'Layer Name to use
BindLayerObject.RefColumn1 = ("LONG") 'Name of Longitude column
BindLayerObject.RefColumn2 = ("LAT") 'Name of Latitude column
```

```
'Add the dataset and create the permanent layer
Set ds = Map1.Datasets.Add(miDataSetADO, rs, "My Dealers",  "Dealer", ,
BindLayerObject)
```

数据绑定如何使用 GeoDictionary

MapX 使用 GeoDictionary 文件（默认名称为 GEODICT.DCT）跟踪与数据绑定有关的信息。MapX 提供了 GeoDictionaryManager50.EXE 实用程序，可以帮助管理 GeoDictionary。

使用 Datasets.Add 方法时， MapX 可以分析您的数据，确定数据如何绑定到地图上。尤其是，该方法具有以下自动特性：

- 确定数据源中哪一列包含地理信息。
- 确定要绑定到的地图图层。

上述两个特性均是可选的。使用参数可以明确指定数据源中哪一列包含地理信息以及应绑定到哪个地图图层。如果您知道其中任意值，为了提高性能，应明确指定。

要使 MapX 可以将数据绑定到地图图层上，需要满足以下条件：

1. **地图的地理关键字列需要编入索引。**

   大多数地图在安装时提供了编入索引的关键字列，但是由地图创建者确定哪些列是关键字列。例如，“United States”表可能至少包含 3 个关键字列：州名 （“New York”）、州名缩写（“NY”）和 FIPS 代码 (36)。 MapInfo 提供的美国地图使用前两个作为关键字列，因此要求数据源中的数据也包含州名或州名缩写。

2. **地图及其关键字列必须在 GeoDictionary 中指定。**

   地图图层中的某些关键字列可能不包含唯一值。例如美国的市 （县）名：多个州中有名为“Warren”或 “Washington”的市 （县）。因此，如果某个数据行包含市 （县）名“Washington”， MapX 需要更多信息才能解决不确定性。以美国市 （县）为例，使用州来解决不确定性。 SecondaryGeoColumn 参数用于指定数据源中包含解决不确定性所需信息的列。GeoDictionary 包含哪些地图包含不唯一的关键字及什么是其重新定义地图的有关信息。

数据绑定使用的 MapX 事件

将列定义为数据源中的地理列并确定要绑定到的地图图层后，绑定即开始。数据源的每一行与图元“对应”，并导入数据。如果某行中包含的地理列与图元不对应 （可能是拼写错误 [“NA”而不是“MA”]，或只是地图上没有的数据值 [“Puerto Rico”]），则会触发 DataMismatch 事件通知容器。可以忽略 DataMismatch 事件，此时 MapX 只是忽略该行。

在自动数据绑定期间， MapX 将调用 ResolveDataBind，即使数据不确定。

在 “不绑定”数据绑定期间，使用 RequestData 事件建立一个循环，使容器为 MapX 提供数据，一次一个单元格。

刷新数据集

尽管数据源中的数据在 MapX 中合计并存储，但在数据源中的数据更改时， MapX 不会反映更改。可以使用 Dataset.Refresh 方法让 MapX 重新读取源数据并重新合计和存储。不过，这可能非常耗时。Dataset.Refresh 不适用于创建新点图层的数据集。解决方法只需删除该图层并重新添加数据集，或执行以下步骤：

1. 使用 Datasets.Add 创建新点图层， BindLayer 类型设置为 **miBindLayerTypeXY**。在 Datasets.Add 方法中仅指定 X 列和 Y 列，绕过 Fields 对象，只使用 X 列和 Y 列创建新图层。
2. 将数据的其余部分绑定到这个新创建的图层上。使用正常绑定的 Datasets.Add，使用 X 字段作为 Geofield 参数，使用 Y 字段作为 SecondaryGeofield 参数。第二个数据集可以刷新。

注：这种方法在向数据源中添加新行时，不会向图层中添加新图元。

不同类型的数据源

MapX 使您可以绑定到多种不同类型的数据源。 Datasets.Add 方法的第一个参数可以指定 **DatasetTypeConstants** 值，该值指定要进行绑定的数据类型。

miDatasetADO （ActiveX 数据对象）

Datasets.Add 的 SourceData 参数必须是 ADO Recordset 对象。 ADO Recordset 对象可以创建， Recordset 对象的 Source 属性可以设置为一个简单的文本命令 （例如文件系统提供程序的目录名或 DBMS 型提供程序的 SQL 语句）。实现 ADO 可以使关键 Internet 方案占用空间少、网络通信量最小，前端与数据源之间的图层数最少，这些全部为了提供轻量高性能的接口。

miDatasetDAO （数据访问对象记录集）

Datasets.Add 的 SourceData 参数必须是 DAO Recordset 对象。该对象可以从 Visual Basic 数据控件、Access 窗体获得，也可以使用 Visual Basic、 Access 或 C++ 中创建。

注：MapX 最高支持 3.6 版的 DAO 数据。

miDatasetDelphi4 & miDatasetDelphi5 （Delphi DataSource 对象）

这些数据类型仅用于 Borland Delphi 编程语言。要创建 Delphi 数据集，请使用以下步骤：

1. 将数据源 （例如表或查询控件）加入 Delphi 窗体，并通过设置 DatabaseName 和 TableName 属性，将其指向正确的 BDE 数据源。将其 Active 属性设置为 True，以打开数据库。
2. 将 DataSource 控件加入窗体。将其 DataSet 属性设置为您在第 1 步中创建的任意控件。
3. 在菜单中选择 “View” -> “Project Manager”。
4. 在 “Project Manager”窗口中右键单击项目的 .exe 文件，然后选择 “View Source”。
5. 在源代码窗口中，将 “ShareMem”行直接插在 “uses”行后面。
6. 保存项目。

miDatasetGlobalHandle （以制表符分隔的数据）

使用该类型的数据绑定可以导入使用制表符分隔的数据块。 Datasets.Add 的 SourceData 参数必须是 VT_I4 类型的变量，并且 lVal 等于全局内存句柄。全局内存块中每个数据行的格式如下：

```
Field  TAB  Field  TAB  Field  CRLF
Field  TAB  Field  TAB  Field  CRLF
. . .
```

其中 Field 是带引号的字符串值或不带引号的数字值； TAB 为字符 0x09 ； CRLF 为回车换行符 0x0D 0x0A。

miDatasetLayer （MapInfo 表）

通过该类型的数据绑定可以创建使用 MapInfo 表中的字段的数据集。

miDatasetNotesView、 miDatasetNotesQuery (Lotus Notes)

这两种类型的数据绑定专门针对 Lotus Notes。

miDatasetODBC （开放数据库连接数据源）

MapX 可以使用 ODBC 从 ODBC 数据源中检索数据。需要使用 ODBCQueryInfo 对象指定连接字符串、数据源名称和要执行的 SQL 字符串。

ODBCQueryInfo.ConnectString 属性用于设置连接字符串，以便与 ODBC 数据源连接。例如，指定“ODBC;”。还可以加入 “uid=”，“pwd=”或 DLG=。如果缺少必要的信息，数据源的 ODBC 驱动程序会提示用户。

“DLG=”控制连接对话框的显示：DLG=0 表示从不打开对话框， DLG=1 表示总是打开一个对话框，DLG=2 显示连接对话框，但是仅在需要时显示 （即如果未提供所有必要的信息时）。

miDatasetOLEData (PowerBuilder)

用于 PowerBuilder 等容器，在初始化 MapX 控件时，将数据以 miDataGlobalHandle 中说明的格式传递给 MapX。然后调用 miDatasetOLEData 类型的 Datasets.Add，通知 MapX 使用以前导入的数据建立数据集。

miDatasetRDO （远程数据对象）

Datasets.Add 的 SourceData 参数必须是 RDO。 RDO 提供一组对象，使用户可以连接到数据库、执行查询和存储过程、处理结果以及将更改提交给服务器。

miDatasetSafeArray （安全数组）

通过该类型的数据绑定可以创建使用数据数组中的字段的数据集。

miDatasetUnbound （备用方法）

如果无法支持某种其它格式， MapX 会提供备用的数据绑定方法，使您可以建立事件循环，而 MapX 询问容器数据值，一次一个单元格。 MapX 将以需要的数据值行号和列号发送 RequestData 事件。然后，可以使用任何方法获取数据并通过该事件将数据回馈给 MapX。

miDatasetXML

MapX XML 数据集驱动程序是 MapX 数据集驱动程序接口套件的实现。MapX 数据集驱动程序接口基于 Microsoft 组件对象模型 (COM) 体系结构。尤其是， MapX XML 数据集驱动程序提供了一组接口，可以通过 XML 数据访问 MapX。有关在 MapX 应用程序中使用 XML 数据集的详细信息，请参见帮助主题第 99 页的 “XML 支持”

miDatasetOCI （Oracle 调用接口数据源）

MapX 可以使用 OCI 从 Oracle 数据源中检索数据。需要使用 OCIQueryInfo 对象指定连接字符串和要执行的 SQL 字符串。

OCIQueryInfo.ConnectString 属性用于设置连接字符串，以便与 Oracle 数据源连接。在使用 ORAINET 工具包 (OCI) 通过 Oracle Spatial 添加图层时，该参数与 LayerInfo 对象的 ConnectString 参数的值相同。有关详细信息，请参见第 157 页的第 12 章：“访问 DBMS 中的数据”。

XML 支持

MapX XML 数据集驱动程序利用 Microsoft XML Parser。Microsoft XML Parser 有多个版本。不支持版本 1.x。 MapX XML 数据集驱动程序已经过测试，支持版本 2.x。此外，即将发布的版本 3.0 的 Microsoft XML Parser 可能支持，但是尚未经过测试。其它语法分析程序目前不支持。

有关标准一致性方面的问题， MapX XML 数据集驱动程序的标准一致性基于在系统中注册的 Microsoft XML Parser 版本。例如版本 2.5 的 Microsoft XML Parser 标准一致性包括：

- XML：一致性基于扩展标记语言版本 1.0 的规范。
- XSL：一致性基于 XML 样式表语言工作草案 (12.18.1998)。

有关详细信息，请参阅 Microsoft 为 Microsoft XML Parser 提供的支持。

版本 2.0 的 Microsoft XML Parser 作为安装的一部分提供。不过，在编写本文时，版本 2.5 SP1 的 Microsoft XML Parser 可以作为单机再分发版提供，随 Microsoft Windows 2000 SP1 和 Microsoft Internet Explorer 版本 5.01 或更高版本提供。

将 MapInfo XML 数据集加入 Map 对象

MapX 组件库的 Datasets 容器包括添加新数据集的方法。前两个参数对每个数据集驱动程序是唯一的。第一个参数是数据集驱动程序的标识符。每个数据集驱动程序通过唯一的数字代码标识。 MapX 组件库包含这些标识符预先定义的常数。 MapX XML 数据集驱动程序预先定义的常数以及实际的数字代码如下：

```
miDataSetXML = 14
```

第二个参数提供数据集驱动程序在获取数据时所需的必要的元数据。如果是 MapX XML 数据集驱动程序，有两个机制用于传递所需的元数据。两个机制中第一个比较简单的机制要求提供 MapInfo XML 数据集位置的字符串，格式为统一资源标识符 (URI) 或路径 （本地或 UNC）。如果是路径，路径是相对于 MapXtreme 服务器的路径。以下是简单的 Microsoft Visual Basic 示例：

```
DIM XMLURL As String
Dim XMLDataset As MapXLib.Dataset
'Initialize XMLURL
XMLURL = "http://www.xml.com/DataSource.xml"
'Assume MapObject exists within a form
Set XMLDataset = MapObject.Datasets.Add(miDataSetXML, XMLURL)
```

第二个机制较复杂，涉及创建另一个自动化对象 MXMLDatasetDriverConfiguration 对象。该对象具有两个属性，第一个属性是 SourceLink，提供标识 MapInfo XML 数据集位置的 URI 或路径；第二个属性是 FilterLink，提供标识 XSLT 文档位置的 URI 或路径，用于将指定的基于外部方案的 XML 数据集转换为 MapInfo XML 数据集。以下代码段是简单的 Microsoft Visual Basic 示例：

```
Dim XMLDataset As MapXLib.Dataset
DIM XMLResource As MXMLDatasetDriverConfiguration
' Set XMLResource SourceLink
XMLResource.SourceLink = "http://www.xml.com/DataSource.xml"
' Set XMLResource FilterLink - currently unsupported
XMLResource.FilterLink = "http://www.xml.com/DataFilter.xml"
' Assume MapObject exists within a form
Set XMLDataset = MapObject.Datasets.Add(miDataSetXML, XMLResource)
```

创作 MapInfo XML 数据集

创作 MapInfo XML 数据集是指能够开发有效的 XML 文档，可以由 MapX XML 数据集驱动程序正确处理。要了解如何操作，需要了解一般如何创作 XML 文档。因此，有必要查看一下 MapInfo XML 数据集基于的方案。注意， MapInfo XML 数据集是基于 XML 方案进行设计。在编写本文时， XML 方案不是普遍接受的标准；该实现利用了 Microsoft XML Parser 版本 2 的功能，之后支持有限版本的 XML 方案。

有效且格式正确的 MapInfo XML 数据集所基于的方案是 Microsoft XML `rowset` 方案的一个子集。Microsoft `rowset` 方案的通用资源名为 `schemas-microsoft-com:rowset`。以下代码段是基本的 MapInfo `dataset` 方案，属于 Microsoft `rowset` 方案的一个子集：

```
<s:Schema xmlns:s='urn:schema-microsoft-com:xml-data'
xmlns:dt='urn:schemas-microsoft-com:datatypes'>
<!-- baserow definition -->
<s:ElementType name='baserow' content='eltOnly'/>
<s:element type='baserow'/>
<!-- data definition -->
<s:ElementType name='data' content='eltOnly'>
<s:element type='baserow' minOccurs='0' maxOccurs='*'/>
</s:ElementType>
```

```
<s:element type='data'/>
</s:Schema>
```

注：上面代码段的第一行标识两个重要的名称空间。第一个名称空间是指 Microsoft schema 方案，定义方案定义语言。方案定义特定类别的 XML 文档的有效内容或 XML 文档的结构。MapInfo XML dataset 方案利用内嵌方案，一个定义数据内容结构的文档片断。第二个名称空间是指 Microsoft datatype 方案。该方案提供所支持的数据类型的语言定义。

MapInfo XML 数据集由两个不同的文档片断组成。第一个文档片断是内嵌方案。该方案片断符合前面所述的 Microsoft schema 方案，定义 MapInfo XML 数据集的结构。每个列通过使用名称和类型作为独特的行元素属性来标识。第二个文档片断包含实际数据。数据片断符合 MapInfo dataset 方案的数据定义，包含代表各行的多个行元素。以下两个示例提供 MapInfo XML 数据集的示例，两个示例显示相同的数据。在两个示例之后，我们详细分析了 XML 文档，研究各个文档片断。

示例 A

```
<xml xmlns:s="urn:schema-microsoft-com:xml-data" xmlns:ds="urn:schemas-
microsoft-com:rowset" xmlns:dt="urn:schemas-microsoft-com:datatypes"
xmlns:x="#DatsetSchema">
<!-- schema -->
<s:schema>
<s:elementType name='row' content='eltOnly'>
<s:attributeType name='StateAbbr' dt:type='string'/>
<s:attributeType name='StateName' dt:type='string'/>
<s:attributeType name='StatePop' dt:type='14'/>
<extends type='rowbase'/>
</s:elementType>
</s:schema>
<!-- data -->
<ds:data>
<x:row StateAbbr ='AK'StateName='Alaska' StatePop='550043'/>
<x:row StateAbbr ='NY' StateName='New York' StatePop='17990455'/>
</ds:data>
</xml>
```

示例 B

```
<xml xmlns="#xmldataset">
<!-- schema -->
<schema>
<elementType name='row' content='eltOnly'>
<attributeType name='StateAbbr' type='string'/>
<attributeType name='StateName' type='string'/>
<attributeType name='StatePop' type='14'/>
<extends type='rowbase'/>
</elementType>
</schema>
<data>
<row StateAbbr ='AK'StateName='Alaska'StatePop=' 550043'/>
<row StateAbbr ='NY' StateName='New York' StatePop='17990455'/>
```

```
</data>
</xml>
```

首先，我们来看文档的根标记 <xml> 标记。该片断中值得注意的重点是名称空间：

```
<xml xmlns:s="urn:schema-microsoft-com:xml-data" xmlns:ds="urn:schemas-
microsoft-com:rowset" xmlns:dt="urn:schemas-microsoft-com:datatypes"
xmlns:x="#DatsetSchema">
```

前三个名称空间是指前面所述的 Microsoft schema、 rowset 和 datatypes 方案。最后一个名称空间是内嵌方案的内部名称空间 （后面会详细介绍）。有一个重要问题需要讨论 – 名称空间别名的使用。 MapX XML 数据集驱动程序假定 Microsoft schema 方案定义 schema、 elementType 和 attributeType 元素。如果这些元素使用名称空间别名作为前缀，别名必须引用 Microsoft schema 方案。另外， MapX XML 数据集驱动程序假定列定义的 “‘type' 属性由 Microsoft datatypes 方案定义。如果这些属性使用名称空间别名作为前缀，别名必须引用 Microsoft datatypes 方案。最后， data 元素假定在 Microsoft rowset 名称空间中定义。如果该数据元素使用名称空间别名作为前缀，别名必须引用 Microsoft rowset 名称空间。MapX XML 数据集驱动程序明确检查 Microsoft 名称空间，确定是否使用别名。如果别名是上述任何元素的前缀，但是引用不支持的名称空间， XML 数据集将无法正确处理。

Microsoft schema 名称空间的通用资源名为 urn:schema-microsoft-com:xml-data。 Microsoft datatypes 名称空间的通用资源名为 urn:schemas-microsoft-com:datatypes。 Microsoft rowset 名称空间的通用资源名为 urn:schemas-microsoft-com:rowset。

简要看一下 *B 示例* 。注意，只有内部名称空间在根标记中引用。还应注意，上面介绍的元素没有使用任何名称空间别名。这将传递验证检查，通过 MapX XML 数据集驱动程序处理。对于内部名称空间，MapX XML 数据集驱动程序没有任何要求，只要文档有效。

现在，再看一下 *A 示例* ，查看内嵌方案的片断。：

```
<!-- schema -->
<s:schema>
<s:elementType name='row' content='eltOnly'>
<s:attributeType name='StateAbbr' dt:dt='string'/>
<s:attributeType name='StateName' dt:dt='string'/>
<s:attributeType name='StatePop' dt:dt='long'/>
<extends type='rowbase'/>
</s:elementType>
</s:schema>
```

该方案标识称为 row 的元素，该元素实际是扩展 rowbase 元素，一个在上面的 MapInfo dataset 方案和 Microsoft rowset 方案中定义的元素。根据该内嵌方案， row 元素由三个属性组成。这些属性是行的实际列。包括字符串类型的 StateAbbr 属性 / 列、字符串类型的 StateName 属性 / 列以及长整型的 StatePop 属性 / 列。

此处要注意的重点是 – 一定要了解 XML 元素标记区分大小写。当前版本的 MapX XML 数据集驱动程序期望 schema 取代 Schema、 elementType 取代 ElementType、 attributeType 取代 AttributeType。如果不符合该原则，内嵌方案将无法如期处理。

快速看一下 *B 示例* ，同样注意 schema、 elementType 和 attributeType 元素没有使用名称空间别名作为前缀，但是内嵌方案未更改。

最后，查看 *A 示例* 中的数据片断：

```
<!-- data -->
<ds:data>
<x:row StateAbbr ='AK'StateName='Alaska'StatePop='550043'/>
<x:row StateAbbr ='NY' StateName='New York' StatePop='17990455'/>
</ds:data>
```

该片断的根元素是数据元素，为 *A 示例* 在上面的 MapInfo `dataset` 方案和前面所述的 Microsoft `rowset` 方案中定义。因此，标记使用相应的名称空间别名作为前缀，此时为 `ds`。根据方案，包含一个或多个基于 `rowbase` 元素的元素，在该示例中为在上面的内嵌方案中定义的 `row` 元素。每个元素对内嵌方案中标识的每个属性 / 列均具有属性。注意， `row` 元素使用指定的名称空间别名作为前缀。参见 *B 示例*，再次注意，唯一的区别在于缺少名称空间别名。

7 章

- 什么是地图图元?
- 什么是 Features 集合?
- 什么是 Selection 集合?
- 使用 Features 集合
- 了解 Selection 集合
- 图元编辑

图元和选择集合

使用 Selection 和 Feature 对象的方法可以 “标记”或选择符合特定条件的图元。地图上代表纽约市的点就是 Feature 对象的一个示例。假如您需要查找加利福尼亚帕洛阿尔托 5 英里以内的所有潜在客户。在创建了该数据集合后，可以浏览收集的数据、打印数据、计算平均值、计算符合该条件的客户数、将数据保存到文件中或执行其它任务。在本章中，我们将详细介绍图元和选择集合。

什么是地图图元?

地图图元是地图上的点、线条或区域之类的地理对象。

例如，美国地图可能包含州区域、高速公路线条和城市点。在 MapX 中，地图图元表示为 Feature 对象。例如，纽约州可能属于区域类型的 Feature 对象，洲际 95 号公路属于线条类型的 Feature 对象，纽约市属于点类型的 Feature 对象。

什么是 Features 集合?

在 MapX 中，组成地图的不同图层通常在每个图层中拥有相同的图元类型。例如，“US States”图层包含代表每个州的区域图元，“US Highways”图层包含代表美国主要高速公路的线条图元，“US Capitols”图层包含代表每个州的首府的点图元。在 MapX 中，地图图层中的所有图元或地图图层中的部分图元表示为一个 Features 集合。许多 Layer 对象方法返回图层中的 Features 集合。 Features 集合中的图元在地图中**不会**突出显示。要突出显示，应使用 Selection 集合。

什么是 Selection 集合?

与 Features 集合类似， Selection 集合也是 Feature 对象的集合。不过， Selection 集合代表当前选择的 Feature 对象 （用户使用选择工具单击地图时选择的或利用下列各种选择方法选择的）。每个 Layer 对象具有自己的 Selection 集合 (Layer.Selection)。 MapX 自动突出显示 Selection 集合中的所有图元，将这些图元作为所选的图元。

使用 Features 集合

Features 集合与 Selection 集合类似，因为两种集合都是 Feature 对象的集合。不过， Features 集合与 Selection 集合的方法集和属性集不同，两种集合类型的行为方式也不同。所选图元会自动突出显示，而 Features 集合中的图元不会突出显示。

获取图层中的图元

在对 Features 集合进行任何操作之前，必须先创建该集合。 Layer 对象方法创建 Feature 对象集合。

以下 Layers 集合方法提供了获取 Features 集合的各种方法。

方法	说明	代码示例 （Dim fs as Features ‘this creates a collection of Features）
AllFeatures	返回包含图层中所有图元的 Features 集合。	Set fs = Map1.Layers(2) _ .AllFeatures

方法	说明	代码示例 （Dim fs as Features ‘this creates a collection of Features）
NoFeatures	返回图层的空 Features 集合。	Set fs = Map1.Layers(9).NoFeatures
SearchWithinDistance	返回由点对象指定距离内的图元组成的 Features 集合。	Set fs = Map1.Layers(3). _ SearchWithinDistance (objPoint, _ 36.5, miUnitMile, _ miSearchTypeCentroidWithin)
SearchWithinFeature	返回由另一个指定区域图元内的图元组成的 Features 集合。	Set fs = Map1.Layers(3). _ SearchWithinFeature _ (ftr, miUnitMile, _ miSearchTypeCentroidWithin)
SearchWithinRectangle	返回指定矩形边框内的 Features 集合。	Set fs = Map1.Layers(3). _ SearchWithinRectangle(miRect, _ miUnitMile, _ miSearchTypePartiallyWithin)
SearchAtPoint	返回由指定点的图元组成的 Features 集合。	Set fs = Map1.Layers(3) .SearchAtPoint(objPoint)

SearchTypeConstants

搜索常数	说明
MiSearchTypeCentroidWithin	将质心在区域内的图元加入搜索。
MiSearchTypeEntirelyWithin	将区域所包含的图元加入搜索。
MiSearchTypePartiallyWithin	将任何部分在区域内的图元加入搜索。

处理 Features 集合

以下方法可以通过在集合中添加、删除、复制图元对象来处理 Features 集合：

方法	说明
添加	将 Feature 对象或整个 Features 集合加入集合。
Clone	将集合复制为另一个 Features 集合对象。
Common	将该集合与另一个 Features 集合组合，使该组合只包含两个集合中均存在的图元 （INTERSECT 集合运算）。
删除	从该集合中删除 Feature 对象或 Features 集合中的所有图元（SUBTRACT 集合运算）。
Replace	使用 Feature 对象或 Selection 集合对象中的所有图元取代集合中的内容。

Feature 对象

Features 集合由一组 Feature 对象组成。 Feature 对象对应于地图上的图元，例如符号、线条或区域。

Feature 对象对应于图层中的图元 （实际实体），例如纽约、芝加哥、路易斯安那、科特兰市或 I-10 号高速公路。

独立图元

Feature 对象方法可以创建和处理独立图元对象。在创建独立图元对象时，必须将该图元对象连接到地图上，然后才能引用其方法或属性。将图元连接到地图上就是将地图的坐标系与图元关联。

Feature 对象属性

以下属性定义图元对象：

属性	说明	代码示例
CenterX	包含图元的 X 轴质心。	Print ftr.CenterX
CenterY	包含图元的 Y 轴质心。	MsgBox ftr.CenterY
FeatureID	包含图元的 ID。图层中的每个图元均包含在图层内唯一的 ID。这是一个整数值。	IVar = ftr.FeatureID
Length	包含图元的长度。	Print ftr.Length

Perimeter	该长度是图元的周长。	If ftr.Perimeter > 500 Then
		Print “Too Long”
		End If

Name	包含图元的名称。	MsgBox ftr.name
类型	包含图元的类型 （点、线条等）。	ftr.Type = miFeatureTypeSymbol

了解 Selection 集合

MapX 的基本功能是选择地图上的图元，使您可以对图元执行其它任务。用户可以通过单击地图选择一个或多个图元 （点、线条、区域等）。 MapX 会突出显示所有选择的图元。

要查看所选图元的列表，应使用 Selection 集合，该集合是 Feature 对象的集合。 Selection 集合还提供各种方法 （例如 SelectByRadius），使您可以进行各种类型的选择，例如，选择某个城市特定半径内的所有图元。所选的图元将在地图上突出显示。每个图层包含所选图元对象的集合 (Layer.Selection)。

Selection 集合提供在集合中添加和删除图元的方法。此外，如果已有 Selection 集合或 Features 集合，可以在集合中添加、删除、复制图元，或查找两个集合的交集。

有关完整的列表，请参见第 446 页的 “Selection 集合”。

SelectionType 常数

以下常数确定要对选择结果进行的操作。

选择常数	说明
MiSelectionNew	创建新 Selection 集合。
MiSelectionAppend	加入当前 Selection 集合。
MiSelectionRemove	从当前 Selection 集合中删除。

以下代码在指定点创建新 Selection 集合：

```
Private Sub Command1_Click()
    Map1.Layers(5).Selection.SelectByPoint -98, 31.56, miSelectionNew
End Sub
```

SelectionChanged 事件

该事件在 Selection 集合更改时调用。该事件使容器可以响应地图上所作的选择。用户可以使用选择工具或使用 Layer 对象中的一种选择方法更改 Selection 集合。

图元编辑

MapX 可以创建、修改或删除组成地图图层的图元 （点、线条、区域等）。可以编辑任何基于普通 MapInfo 表 (.tab) 文件的图层。

注：MapX 不提供事务处理支持。也就是说，在更新图元时，添加 / 删除 / 更改图元会立即生效。以后不需要保存任何更改，而且不能撤销操作。

如何创建新地图图元

创建新地图图元有两种方法。可以通过分配新 Feature 对象来创建图元，也可以通过使用 Map 对象 FeatureFactory 属性的方法对现有图元执行操作 （例如缓冲）来创建图元。这些类型的图元称为独立图元。独立图元受到一些限制：不能加入任何集合，只允许使用用于定义图元的方法和属性。例如，不能对独立图元使用 Area 属性，独立图元没有 Feature.Layer 属性的值。

图层中的图元不是独立图元，所有属性和方法均可以使用。

FeatureFactory 方法

通过 FeatureFactory 对象的方法可以创建新地图图元，也可以通过对现有图元执行操作 （例如缓冲）来创建图元。

其中大多数方法均返回独立图元对象。这些图元对象自动连接到地图上 （即已与坐标系关联）。也就是说，不需要对这些方法返回的图元使用 Attach 方法。

要获取 FeatureFactory 对象，应引用 Map.FeatureFactory 属性。

MapX 联机帮助提供了如何使用各种 FeatureFactory 方法的详细说明。

如何修改现有图元

如果有指向某个 Feature 对象的指针，该指针代表图层中的实际图元；图层中的图元不是独立图元，所有属性和方法均可以使用。显示的任何属性和方法反映图层中图元的值。

开始通过设置图元样式或者点或位置修改图元后，更改在 “更新”该图元后才会生效。这种安排使您可以对图元进行许多更改，不必在每次更改后都等待数据库更新和屏幕重绘。负作用是 MBR、区域、长度等在更新图元后才会更改。

要更新图元，请执行以下操作之一：

- 如果修改了某个图元并要提交更改，应使用 Feature.Update 方法。
- 如果要使用其它图元取代某个图元，应使用 Layer.UpdateFeature 方法。
- 图元更新时，旧的 MBR 和新的 MBR 均失效，以便屏幕可以正确重绘。

如何删除现有图元

要删除图元，应使用 Layer.DeleteFeature 方法。该图元及其代表的行均将立即从 MapInfo 表中删除。

如何获取要编辑的图元

可以允许用户通过单击来选择图元。然后，要访问 Selection 集合，程序可以使用图层的 Selection 集合。 Selection 集合还提供各种方法 （例如 SelectByPoint），使您可以将图元加入集合。

Layer 对象提供上文所述的各种方法 （例如 SearchAtPoint），使您可以获取 Features 集合。

可以使用 Find.Search 方法执行返回 FindFeature 对象 （Feature 对象的超类）的查询。

检查区域或线条的部分

MapX 中的线条图元或区域图元由许多 Point 对象集合组成。这些 Points 集合的集合通过图元的 Feature.Parts 集合访问。

设置字体属性

Style 对象的属性可以控制地图上图元的外观。例如，可以通过设置 Style.SymbolFontColor 属性更改符号的颜色。

注意， Style 对象不提供 “Size”属性，也不提供 “Bold”或 “Italic”属性。如果需要设置符号的字体大小、粗体或斜体设置，必须获取并修改 Font 对象，如下所述：

1. 引用 **Style.SymbolFont** 属性获取 Font 对象。
2. 通过修改 Font 对象的属性控制符号的外观。

Font 对象不属于 MapX 对象模型；而是标准的对象 （OLE_FONT、 IFontDispatch）。

设置文本图元的字体属性非常类似，只是引用 Style.TextFont 而不是 Style.SymbolFont。

如果指定 Name 属性，应确保字体名称的拼写正确。如果字体名称拼写错误，即使是在需要大写字母的地方指定了小写字母，也会造成问题。

主题地图绘制和分析

主题地图绘制是用于分析和可视化数据的一种强大方法。您为数据提供图形格式，以便可以在地图上查看数据。使用主题遮蔽在地图上显示数据时，数据列表中几乎无法检测到的图案和趋势可以清晰地展现。

有了 MapX，可以创建拥有以下主题类型的主题地图的应用程序：值范围、分级符号、点密度、个别值以及柱状图和饼状图。

- 什么是主题地图绘制?
- 规划主题地图
- 主题地图绘制的类型
- 处理主题地图
- 自定义主题图例
- 标签主题

什么是主题地图绘制?

主题地图绘制是根据特定的主题遮蔽地图的过程。主题通常是数据的某个部分或某些部分，可以从数据集获取。主题利用颜色阴影、填充图案、符号或柱状图和饼状图，以可视化的形式显示数据。根据数据中的特定值将这些颜色、图案或符号分配给地图对象，可以创建不同的主题地图。柱状图和饼状图可以对数据集中的每个记录进行数据比较。

主题地图可以使用许多方法来图解数据。最常用的一种主题地图就是气象图。如果显示红色，表示天气很热 （度数高）；如果显示蓝色，表示天气很冷 （度数低）。通过主题地图绘制还可以发现通过表格数据很难看出来的数据趋势。

在 Themes 集合、Theme 对象和 ThemeProperties 对象中使用属性和方法，可以创建并定义自己的主题遮蔽。 Legend 对象存储主题的关键字，说明主题的颜色、形状和大小代表的含义。

规划主题地图

在创建主题地图之前，一定要先了解组成主题地图的元素以及如何将这些元素放在一起。本节将介绍主题变量，在本节您可以获取数据，特别是使用其它表中的数据，还会介绍主题图层的排列和显示。

主题变量

在主题地图上显示的数据称为主题变量。根据要执行的主题分析的类型，地图可能会显示一个或多个主题变量。值范围、网格遮蔽、分级符号、点密度和个别值地图全部检查一个变量。使用柱状图或饼状图可以同时显示多个主题变量。

还可以创建二元主题地图，其中一个地图对象 （例如符号）代表两个不同的数据片。例如，符号颜色可以代表一个主题变量，而符号大小可以代表另一个主题变量。

何处获取数据

开始创建主题地图之前，需要确定要显示的信息并找到信息所在的位置。用于创建主题的数据来自数据集的 Field 对象或 Fields 集合。这些字段作为 Themes 集合 Themes.Add 方法的 Fields 参数传递。

Themes 集合

每个数据集包含一个 Themes 集合。 Themes 集合可以在主题集合中创建、计数、添加或删除 Theme 对象。有关 Themes 集合的方法的信息，请参见第 481 页的 “Theme 对象和 Themes 集合”。

创建主题

如果要使用数据集中的数据按主题遮蔽地图，应使用 Themes.Add 方法为数据集创建 Theme 对象。数据集加入地图后，可以使用一行代码完成主题的创建：

```
Map1.Datasets(1).Themes.Add miThemeRanges "TotPop", _
"My Ranges Theme"
```

该代码使用字段 “TotPop”为 Datasets 集合中的第一个数据集创建范围主题。

有关 Themes.Add 方法的信息，请参见第 485 页的 “Themes.Add 方法”。

MapX 如何选择默认主题

如果未指定 Themes.Add 方法的 Type 参数 （或指定为 miThemeAuto）， MapX 将尝试根据导入的字段数以及已显示的其它主题类型来选择适合的默认主题，如下所述。

字段数	图层类型	算法
大于 1	点、线条、区域	默认的主题类型为饼状图。如果已存在多变量主题，应尝试使用提供的字段集合中的第一个字段创建单变量主题 （参见以下算法）。
1	点	如果绑定字段通过各个值合计，应尝试创建个别值主题。如果已存在范围主题或个别值主题，则不能创建默认主题。如果绑定字段不是通过各个值合计，应尝试创建分级符号主题。如果已存在对象主题，应尝试创建范围主题。如果已存在范围主题或个别值主题，则不能创建默认主题。
1	线条	如果绑定字段通过各个值合计，应尝试创建个别值主题。如果已存在范围主题或个别值主题，则不能创建默认主题。如果绑定字段不是通过各个值合计，应尝试创建范围主题。如果已存在范围主题或个别值主题，应尝试创建分级符号主题。如果已存在对象主题，则不能创建默认主题。
1	区域	如果绑定字段通过各个值合计，应尝试创建个别值主题。如果已存在范围主题或个别值主题，则不能创建默认主题。如果绑定字段不是通过各个值合计，应尝试创建范围主题。如果已存在范围主题或个别值主题，应尝试创建点密度主题。如果已存在点密度主题，应尝试创建分级符号主题。如果已存在对象主题，则不能创建默认主题。

注：饼状图、柱状图和分级符号均属于对象主题。

确定默认主题类型是针对图层进行的。也就是说，在尝试为图层 B 上的新主题确定默认主题类型时，不会考虑图层 A 上的主题。

ThemeType 常数

可以创建的主题类型如下：

类型	说明
miThemeRanged	范围主题
miThemeBarChart	柱状图主题

类型	说明
miThemePieChart	饼状图主题
miThemeGradSymbol	分级符号主题
miThemeDotDensity	点密度主题
miThemeIndividualValue	个别值主题
miThemeAuto	MapX “最佳猜测”主题
miLabelRangedTheme	范围标签主题
miLabelIndividualValuesTheme	个别标签值主题

每个主题有自己的用途和唯一的属性。例如，使用 miThemeRanged，可以根据人口密度按主题遮蔽世界地图。可以使用红色的分级阴影遮蔽国家，最深的红色代表人口密度最大的国家，最浅的红色代表人口密度最小的国家。您可以很容易看出世界人口的分布。

使用主题地图绘制并不仅限于显示数字值。标称值也可以按主题遮蔽。例如，您有地下电缆数据集。过去六个月内尚未使用过的地下电缆将标记为优先状态。使用 miThemeIndividualValue 可以根据修复状态遮蔽这些电缆。值相同的所有记录将使用相同颜色遮蔽。有关每个主题地图类型的详细信息，请参见本章后文的各节。

主题加入指定集合后，将创建 Theme 对象，可以处理对象的各个属性。

Theme 对象

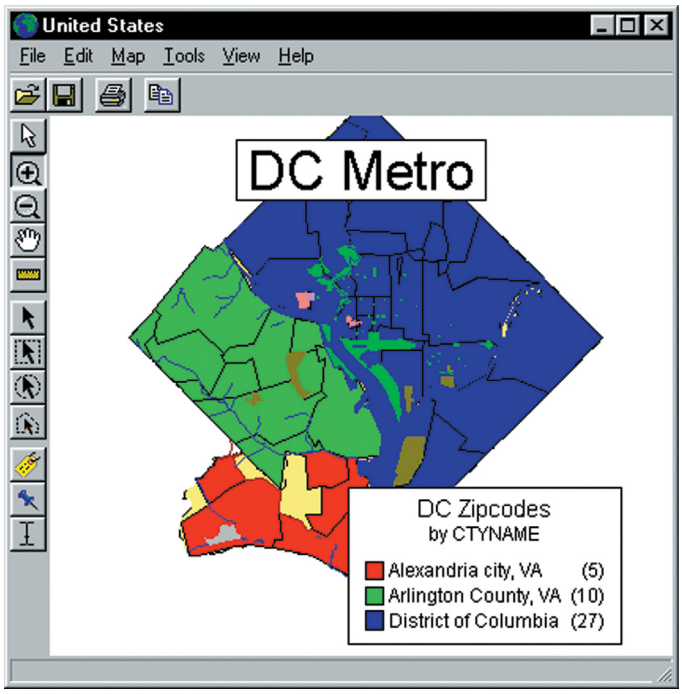
Theme 对象设置 Themes 集合中主题的属性。通过修改 Theme 对象的方法和属性可以确定如何查看主题以及主题的类型等。

有关所有 Theme 对象属性的完整列表，请参见第 481 页的 “Theme 对象和 Themes 集合”。

主题地图绘制的类型

个别值地图

个别值地图显示由数据集特定字段中包含的个别值遮蔽的点、线条或边界。在个别值地图中可以同时使用数字值和标称值。 MapX 为每个唯一的值指定自己的颜色或符号。个别值地图使用符号类型时，符号从默认的地图样式中获取。



例如，软饮料经销商维护华盛顿特区向其购买软饮料的超市的表 （按邮政编码划分）。每个超市以不同的价格销售该经销商的软饮料品牌。如果该经销商使用个别值按价格遮蔽邮政编码边界，所有以 49 分销售该软饮料的商店将使用一种颜色遮蔽，而所有以 51 分销售该软饮料的商店将使用另一种颜色遮蔽，依此类推。为每个唯一的值指定自己的颜色。经销商可以看到超市之间的价格分布，可以根据价格确定哪里应提高销售量。

如果要使用标称数据遮蔽点、线条或边界，只能按个别值遮蔽。标称数据可以是非数字数据 （例如提供的食物的名称、类型或销售的汽车品牌），也可以是数字数据，通过数字代表非数字数据。日期可以认为是数字数据，可以在范围地图和个别值地图中使用。

例如，您获得了消费者调查结果。调查中的一个问题是 “您喜欢在星期日下午做些什么？” 可能的回答包括：

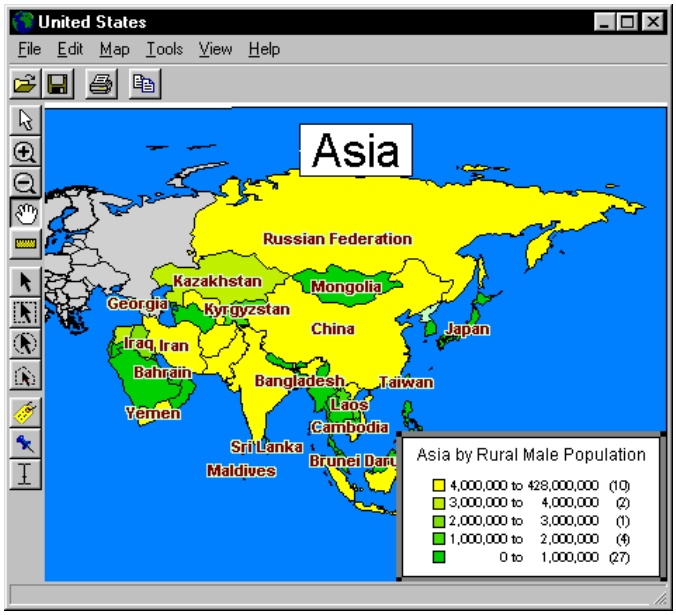
1. 睡觉
2. 看电视
3. 开车
4. 读书
5. 运动或观看运动
6. 参观博物馆或艺术馆
7. 看电影

您希望使用对星期日喜欢的活动的回答来遮蔽每个消费者点。数据集的 SUNDAY 字段包含与消费者喜欢的活动对应的数字。不过，该列中的数字不代表数量值。即使是 7 > 5，也并不代表看电影大于运动或观看运动。在数字作为名称而不是作为值使用时，必须按个别值遮蔽对象。数字只能用于指代娱乐活动，以便为其分配颜色。

个别值主题地图的设置通过 IndividualValueCategories 集合显示，这是 IndividualValueCategory 对象的集合，主题中每个唯一值一个对象。要获取 IndividualValueCategories 集合，应引用 ThemeProperties.IndividualValueCategories 属性。

范围地图

在创建范围主题地图时， MapX 将所有数据集行分成多个范围，并为每一行的对象分配相应范围的颜色、符号或线条。例如，如果有针对看电视地区的气象站的数据集，并且希望根据报告的降雪量遮蔽地点。



使用范围地图功能， MapX 将降雪量分为多个范围。例如，所有在过去一个月内的降雪量在 0 到 5 英寸之间的气象站属于一个范围。降雪量在 5 到 10 英寸之间的气象站属于一个独立的范围。降雪量在 10 到 15 英寸之间的气象站属于第三个范围，而报告的降雪量超过 15 英寸的气象站属于第四个范围。

数据集中的所有记录均分配给某个范围，然后根据该范围分配颜色。例如，报告的降雪量超过 15 英寸的气象站使用红色遮蔽。其它范围使用比红色浅的颜色遮蔽，最后一个范围使用灰色遮蔽 （默认颜色）。在显示地图时，颜色可以很清楚地表明累计降雪量最大和最小的地点。

如果区域的大小与数据值的大小没有直接关系，范围也很有用。

范围值的类型

MapX 可以使用五种分布方法自动创建范围：Equal Count、 Equal Ranges、 Natural Break、 Standard Deviation 和 Quantile。范围通过 ThemeProperties 对象的 DistMeth 属性设置。 DistMeth 属性可以设置为以下任意 DistribMethod 常数。

DistribMethod 常数

Equal Count (miEqualCountPerRange) 在每个范围中的记录数相同。如果希望 MapX 使用 Equal Count 将 100 个记录分为 4 个范围， MapX 会计算范围，根据您设置的舍入因子，每个范围大约包含 25 个记录。

使用 Equal Count （或任何其它范围方法）时，一定要注意任何可能会影响主题地图的极端数据值 （在统计学中，这些值称为异值）。例如，如果按照 Equal Count 遮蔽以下数据库：

John	5000	Andrea	7000
Penny	6000	Kyle	5500
Miguel	4500	Angela	7500
Linda	5000	Elroy	6000
Ben	100	Mark	7000

Ben 和 Miguel 分在同一个范围 （因为他们的值最低）。这样做可能不会产生您需要的结果，因为 Ben 的值明显低于任何其它值。

Equal Ranges (miEqualRangeSize) 按相等大小的范围划分记录。例如，表中的字段包含的数据值在 1 到 100 范围内。需要使用四个相等大小的范围创建主题地图。MapX 产生范围 1-25、26-50、51-75 以及 76-100。

记住，根据数据的分布， MapX 可能会创建没有数据记录的范围。例如，如果按 Equal Ranges 遮蔽以下数据库：

John	100	Andrea	90
Penny	6	Kyle	1
Miguel	4	Angela	92
Linda	95	Elroy	89
Ben	10	Mark	10

MapX 产生四个范围 （1-25、 26-50、 51-75 以及 76-100）。不过请注意，其中只有两个范围 （1-25 和 76-100）实际包含记录。

Natural Break 和 Quantile 是两种显示分布不均匀的数据的方法。

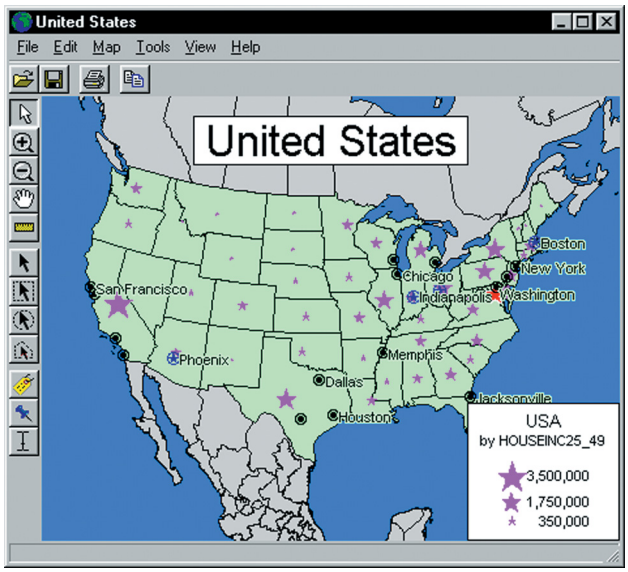
Natural Break (miNaturalBreak) 根据算法创建范围，使用每个范围的平均值将数据更均匀地分布到各个范围。该方法通过分布值，使每个范围的平均值与该范围的每个范围值尽可能接近。这样可以确保范围可以正确地通过其平均值表示，并且每个范围内的数据值非常接近。

使用 **Standard Deviation** (miStandardDeviation) 创建范围时，中间范围在值的平均值处中断，高于和低于中间范围的范围高于或低于平均值一个标准差。

您也可以使用 **Custom** (miCustomRanges) 定义自己的范围。在 RangeCategories 集合中指定自己的范围（MapX 不会计算范围）。

分级符号地图

分级符号地图使用符号代表不同的值。无论使用哪种数据类型，均可以使用分级符号。例如，使用分级符号显示各州的销售订单。使用分级符号主题， MapX 会根据销售订单字段中的值改变每个符号的大小。



您也可以通过将大小成比例的符号分配给消费者的感兴趣程度，表现每个消费者对指定产品所表现出的感兴趣程度。

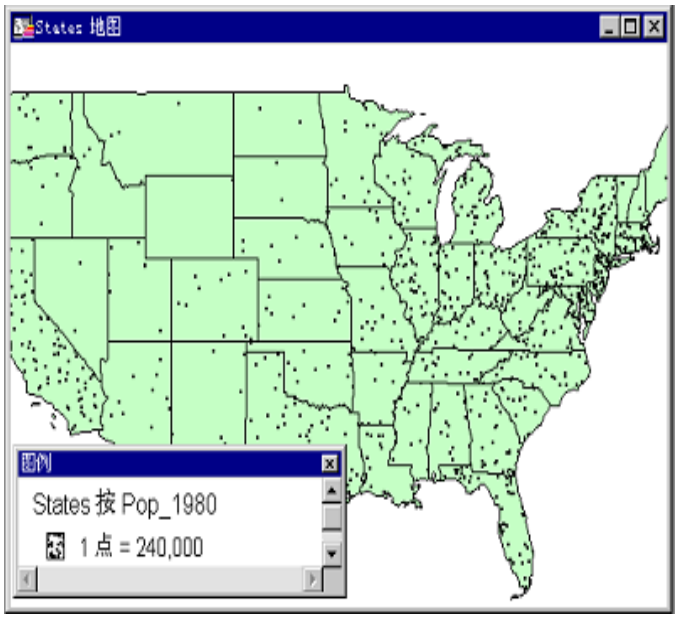
使用数字数据时，分级符号地图最适合。如果使用餐馆表，根据每个餐馆提供的食物类型创建分级符号没有任何意义。不过，如果要显示 20 个不同的快餐店售出的汉堡包数，分级符号则适用。

分级符号地图上有两个主题属性可以自定义：DataValue 和 SymbolStyle。 SymbolStyle 属性控制使用的符号以及按 DataValue 属性指定的值绘制的符号大小。高值和零值之间的所有值包含内插的磅值大小。SymbolStyle 属性是一个 Style 对象，可以控制符号样式的所有方面，例如颜色、字体、旋转等。您也可以提供自己的位图作为符号使用。

Style 对象具有符号选取器 (Style.PickSymbol)，显示 “Symbol Style” 对话框。通过该对话框，用户可以选择符号样式属性。 Style 对象在用户单击对话框中的 “OK” 后更新为新属性。

点密度地图

点密度地图使用点数来代表与边界或区域关联的数据值。某个区域中的总点数代表该区域的数据值。如果某个市 （县）有 10000 个老年人，每个点代表 100 个老年人，该市 （县）界中将有 100 个点。



要显示一个点代表大量内容的原始数据时，点密度尤其适用：人口、快餐店数、销售某个品牌的苏打水的经销商数等。

例如，如果有按市 （县）界划分的人口表，可以使用点密度选项显示每个市 （县）界内人口的集中程度。您可以控制点密度地图的两个主题属性。可以指定一个点的值。例如，您有按市 （县）划分的人口统计表。在 New York 的 Rensselaer County 有 20000 个中学生。如果使用点密度方法，按中学生数遮蔽 Rensselaer County，每个点可能代表 200 个学生。此时， Rensselaer County 中将有 100 个点。

如果增大每个点所代表的值，则减少了地图上显示的点数。可以修改点密度地图，使一个点代表 400 个学生。此时， Rensselaer County 中将只有 50 个点。

```
Map1.Datasets(1).Themes("My DotDensity _
Theme").ThemeProperties.ValuePerDot = 400
```

第二个选项根据您的需要更改点的大小。如果处理大量人口或大量内容，应减小点的大小，使点的分布容易分辨。反之，如果处理小数据集，应增大点的大小，以便更加清晰地图解您的分析。

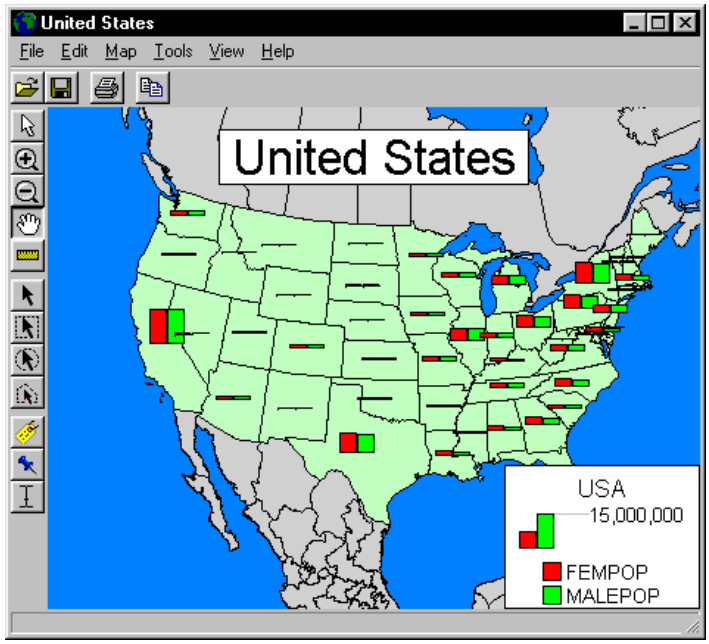
```
Map1.Datasets(1).Themes("My DotDensity _
Theme").ThemeProperties.DotSize = miDotSizeLarge
```

注：区域内点的分布是随机的。例如，如果按人口遮蔽州 （省）， New York 的点遍及整个州；而不是集中在 New York City，尽管该州的大多数人口都居住在此地。

柱状图地图

与一个变量的主题地图 （例如值范围或分级符号）不同，主题柱状图地图可以每行同时检查多个变量。为每个地图对象 （图元）在对象质心处建立柱状图，以便通过比较柱体的高度，分析特定图表中的主题变量。还可以在地图的所有图表中检查同一个变量。

例如，您有包含女性和男性人口的美国州界表。使用柱状图可以创建一个主题地图，为每个州显示一个双柱图表：一个柱体代表女性人口，另一个柱体代表男性人口。可以比较每个州的人口差异，也可以检查多个州并比较一个州与另一个州的人口或人口差异。要获得最佳效果，分析中每个柱状图使用的柱体不要超过四到六个。



您可以控制柱状图的五个主题属性：DataValue、 Size、 Independent、 MultiVarCategories 和 Width。

DataValue 属性与 Size 属性组合使用，控制主题图形相应于特定值的大小。该属性的默认值设置为地图绘制图元的最大数据值。

Size 属性与 DataValue 属性组合使用，控制主题图形相应于特定值的大小。该属性使用 Paper 单位 (PaperUnit) 指定主题图形的高度。该属性是一个双精度值，默认值为 .25 英寸。

Width 属性使用 Paper 单位 (Map.PaperUnit) 指定柱状图主题的每个柱体的宽度。该属性是一个双精度值，默认值为 .25 英寸。

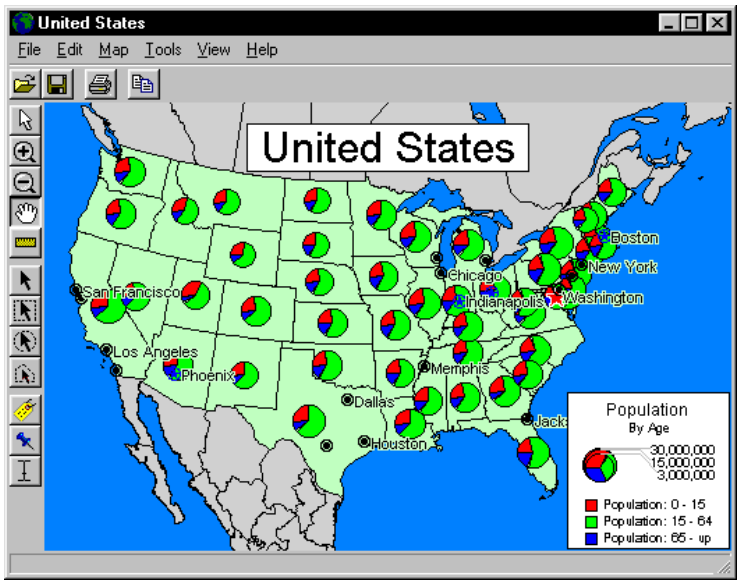
Independent 属性控制柱体的数据值是否应独立对待 （不是人口和平均收入等可比较的值）。该属性是一个布尔值，默认值为 False。对于多值柱状图主题，单个图元的每个柱体的数据与主题中其它字段的柱体无关，或数据字段的最大值明显不同，开发人员应将该属性设置为 True。人口柱状图主题就是该属性的一个示例，其中一个柱体可能代表州的人口，另一个柱体可能代表出口的等级。人口数据可能是百万级，而等级只是在 1 到 50 之间。如果 Independent 属性设置为 True，人口最多的州的人口柱体将等

于出口柱体中最高等级状态的高度。如果 Independent 属性仍设置为 False，将很难从出口等级柱状图获得任何含义，因为排在第一位的州如果只有一个人，出口柱状图将使用州人口柱状图的大小。

柱状图主题地图通过 MultivarCategories 集合展现给 OLE，MultivarCategories 集合是 MultivarCategory 对象的集合。柱状图中的每个柱体在该集合中均有一个对象。

饼状图地图

通过使用饼状图的主题地图绘制，还可以每行同时检查多个变量。与比较柱状图中柱体的高度类似，在饼状图中，比较的是单个饼中的楔形，或检查所有饼状图中的特定楔形。饼状图还可以进行部分比较。



对于分析人口统计数据，饼状图和柱状图均非常有用。例如，您有美国人口统计信息的数据集。您的数据集显示几个主要人口统计组的人口。使用饼状图可以显示每个人口统计组的人口，并查看其在每个饼状图中所占的比例。这样，可以按州查看人口统计组的分布，或整个美国人口统计组的分布。还可以查看一个人口统计组，确定该组中的人口在不同州中的不同之处。要获得最佳效果，分析中每个饼状图使用的楔形不要超过四到六个。

您可以控制饼状图的四个主题属性：DataValue、 Size、 Graduated 和 MultiVarCategories。

DataValue 属性与 Size 属性组合使用，控制主题图形相应于特定值的大小。该属性的默认值设置为地图绘制图元的最大数据值。

Size 属性与 DataValue 属性组合使用，控制主题图形相应于特定值的大小。该属性使用 Paper 单位 (PaperUnit) 指定主题图形的高度。该属性是一个双精度值，默认值为 .25 英寸。

Graduated 属性控制饼状图的大小是否根据饼状图的总值分级。该属性是一个布尔值，默认值为 True。

饼状图地图通过 MultivarCategories 集合展现给 OLE，MultivarCategories 集合是 MultivarCategory 对象的集合。饼状图中的每个楔形在该集合中均有一个对象。

二元主题地图绘制

二元主题地图绘制使用点或线条对象代表两个主题变量。例如，一个星星可以代表一个变量，例如青少年的数目，而填充为蓝色的星星代表他们每年的购买额。

要在 MapX 中创建二元地图，应创建两个主题地图并互相叠放，使对象显示两个变量。

地图和变量的类型

适合二元地图绘制的主题地图类型只有范围地图和个别值地图。可以根据数据为二元地图选择以下两个组合之一：

- 两个范围地图
- 一个范围地图和一个个别值地图

如果使用非数字变量，其中一个地图必须是个别值地图。不能创建包含两个非数字变量的二元地图。

显示属性

要在一个符号内显示两个变量，一定要为每个变量选择不同的符号属性。例如，不能为两个变量同时选择颜色，因为一个颜色会覆盖另一个颜色。选择以下组合之一：

- 颜色和符号类型
- 颜色和大小
- 大小和符号类型

符号类型只应用于标称数据或非数字数据，因为符号类型和数量之间没有内在的关联。

处理主题地图

所有主题属性均可以在运行时调整。以下两种方法可以实现调整：

- Theme.ThemeDlg method.
- 主题的 ThemeProperties 对象。

ThemeDlg Method

Theme 对象包含 **ThemeDlg** 方法，该方法显示一个对话框，使用户可以修改主题。以下代码行显示该方法多么简单。

```
Map1.Datasets(1).Themes(1).ThemeDlg
```

将出现一个对话框，通过该对话框可以更改适合于调用参数的主题类型的参数。

使用该方法可能存在两个缺点。首先，您将受到设计的限制。如果这些对话框不适合您的颜色主题，或您不喜欢它们的布局方式，您却不能更改。使用这些方法的另一个问题是有时它们过于强大。可能您只希望用户更改范围主题中最高范围的颜色或希望保留颜色，而不希望用户选择范围数。使用 ThemeDlg 方法，用户可以更改主题的任何 （及所有）部分。

ThemeProperties 对象

您还可以通过处理 ThemeProperties 对象，只更改主题的属性。该操作很容易实现，使您可以更好地控制用户可以更改和不能更改的内容。ThemeProperties 对象存储在 Themes 集合中。ThemeProperties 对象属性用于定义主题地图的外观（颜色、符号等）。ThemeProperties 对象实际包含主题外观的有关信息。ThemeProperties 对象的某些属性由其它对象提供。

有关所有 ThemeProperties 对象属性的完整列表，请参见第 488 页的 “ThemeProperties 对象”。

几个 ThemeProperties 属性实际上是其它对象。这些对象包括 RangeCategory 对象、 IndividualValue 对象、MultiVar 对象和 Style 对象。要了解 RangeCategory 对象的属性，请参见下表。其它与主题相关的对象行为方式类似。

属性	说明	代码示例
Max	在范围主题中设置某个范围的最大值。	Map1.Datasets(1).Themes(1).Properties.RangeCategories(3).Max = 625
Min	在范围主题中设置某个范围的最小值。	Map1.DataSets(1).Themes(1).Properties.RangeCategories(3).Min = 595
NumItems	显示范围中的项数。	Print Map1.DataSets(1).Themes(1).Properties.RangeCategories(1).NumItems
Style	代表该范围样式的样式对象。	Map1.Datasets(1).Themes.Item(1).Properties.RangeCategories.Item(1).Style.PickRegion

范围主题地图包含一个 RangeCategory 对象的集合，每个范围一个，按升序排序。ThemeProperties.RangeCategories 属性存储该集合。

如果 ThemeProperties.DistMethod 为 MiCustomRanges， MapX 将假定您自己设置该值，并将在对数据值分组时使用您定义的范围。如果重新计算主题时范围出现重叠，则会出错。

自定义主题图例

在创建主题视图时， MapX 会自动创建图例，说明颜色、符号或大小所代表的含义。

Legend 对象

每个主题均有 Legend 对象 (Theme.Legend)。 Legend 对象包含控制图例显示的属性。

使用 Legend 对象的 LegendDlg 方法很容易修改。请参见以下示例以及将启动的对话框。

```
Map1.Datasets(1).Themes(1).Legend.LegendDlg
```

标签主题

通过标签主题可以向图层数据集中添加主题，修改图层的标签显示特性。标签主题有两种。包括：

- 范围值
- 个别值

个别值主题

对于个别值主题，可以使用每个个别值主题类别的 Style 属性分别修改标签样式。可以使用 ApplyAttribute 主题属性修改标签的前景颜色或其所有字体属性。例如，要使第一个个别值的标签使用大写和粗体，应执行以下代码：

```
Theme.ThemeProperties.IndividualValueCategories(1).Style.TextFont.Bold
= True
Theme.ThemeProperties.IndividualValueCategories(1).Style.TextFontAllCap
s = True
Theme.ThemeProperties.ApplyAttribute = miApplyAttributeAll
```

范围主题

对于范围主题，可以使用每个范围主题类别 （或 bin）的 Style 属性修改标签样式。可以使用 ApplyAttribute 主题属性修改标签的前景颜色、大小或其所有字体属性。此外，可以使用 SpreadBy 主题属性，自动将字体大小或前景颜色在所有范围类别之间传播。在应用所有属性并执行传播操作 （颜色或大小）后，所有属性 （自动传播的属性除外）均来自模型样式。在 MapX 中，模型样式是第一个范围类别样式。例如，要自动传播标签的字体大小并且仅修改字体大小，应执行以下代码：

```
Theme.ThemeProperties.ApplyAttribute = miApplyAttributeSize
Theme.ThemeProperties.SpreadBy = miSpreadBySize
```

要添加新的标签主题，可以使用包含以下某个主题 ThemeType 常数的 Themes.Add 方法：

- miLabelRangedTheme
- miLabelIndividualValueTheme.

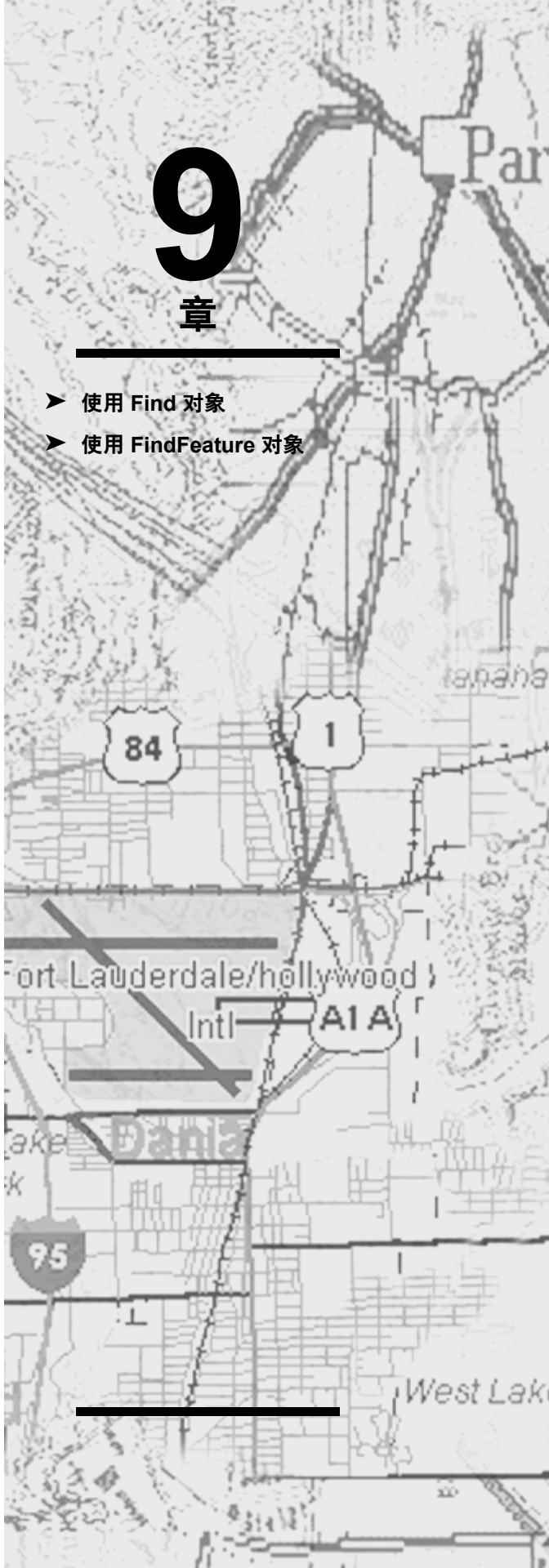
有关可用主题类型的完整列表，请参见第 573 页的 “ThemeTypeConstants”。然后可以使用 Theme.ThemeProperties 属性配置主题。

9 章

- 使用 Find 对象
- 使用 FindFeature 对象

查找地图上的图元

通过 Layer 对象的 Find 方法，您可以在地图对象中搜索图层，并且确定特定图元在该图层中的位置。



使用 Find 对象

Find 对象使您可以确定图元在地图中的位置。您可以查找线条、符号或区域图元。要使用 Find 方法，您正搜索的图层必须具有索引字段。

例如，如果您要确定 Albany 城在 NY 州内的位置，可以使用 Find Object Search 方法搜索城市图层和州图层来找到指定的图元。

Find Object 属性可用于指定查找参数。有关与 Find 对象关联的所有方法和属性的完整列表，请参见第 311 页的 “Find 对象”。

优化边界

Find.Search 中的优化边界用于区分同名的多个图元。例如，如果您旅行到 Albany，则目的地可能位于纽约、加利福尼亚或乔治亚。在您说 “Albany, New York”时，将对该州进行优化边界。

使用 FindFeature 对象

Find.Search 方法返回它通过 FindFeature 对象找到的图元。FindFeature 对象将找到的 Feature 对象的属性作为它自己的属性存储。此外，FindFeature 具有存储 Find 操作的结果码的 FindRC 属性。FindFeature 对象是 Feature 的超类，增加了要返回的最匹配的字符串。 FindRC 属性是指示找到或未找到图元的原因的数值。

结果码

返回与 Find 对象有关的信息，例如找到或未找到它的原因。这是一个数值结果码。下表描述这些这些数值结果码。

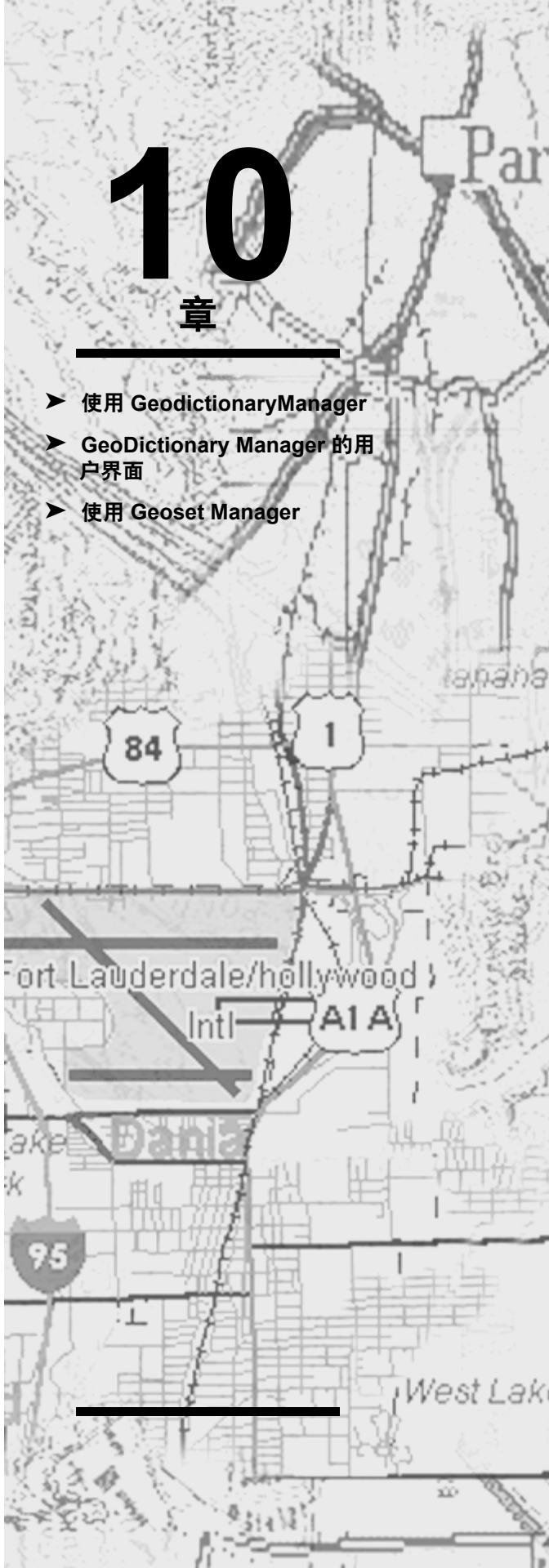
数值	意义
** 个位 **	
xx1	精确匹配。
xx2	使用了缩写文件的替代形式。
xx3 (-)	未找到精确匹配。
xx4 (-)	未找到任何对象名；未找到匹配项。
** 十位 **	
x1x	未确定街道的一侧。
x2x (+ / -)	已在最小 / 最大范围内的地址号。
x3x (+ / -)	不在最小 / 最大范围内的地址号。
x4x (+ / -)	未指定地址号。
x5x (-)	街道不相交。

数值	意义
x6x (-)	匹配的行不具有地图对象。
**** 百位 ****	
1xx (+ / -)	只在非指定区域的一个区域中找到了名称。
2xx (-)	在非指定区域的多个区域中找到了名称。
4xx (-)	未指定任何区域，并且找到了多个匹配项。
5xx (+)	在指定区域中多次找到名称。

一旦确定了结果码后，可按下例所示将这些结果用于所选情况。来自文本框的用户输入指示要确定位置的城市和州。该输入被引入 Find.Search 方法。一旦从主要城市或小城市表中确定了该城市的位置，就可以将注释添加到该地图，并且该地图重新位于找到了该图元的位置的中心。

使用 Geodictionary 和 Geoset Manager

Geodictionary Manager 和 Geoset Manager 是随 MapX 一起提供的实用程序，用于辅助您的地图绘制应用程序。本章介绍如何使用这两种实用程序以及它们各自的界面。



使用 GeodictionaryManager

Geodictionary 是包含与 MapInfo 表有关的注册信息的文件， MapX 可以在自动数据绑定期间匹配这些表。在 Geodictionary 中只应注册可以或将要与之进行匹配的 MapInfo 表。

注：不需要注册应用程序在 Geodictionary 中使用的每一 .tab 文件，实际上，在注册不需要的文件时会造成一些系统开销。

Geodictionary Manager 应用程序用于操作 Geodictionary。可以通过图形用户界面或命令行参数运行 Geodictionary Manager 可执行文件 (GeoDictionaryManager50.exe)。这些命令行选项用于从安装程序调用 GeoDictionaryManager40.exe，从而注册 MapX 应用程序可能需要与 Geodictionary 进行匹配的表。

背景信息

以前的 MapX 版本使用为 Microsoft Map 开发的 MapInfo Data Installer 来管理 Geodictionary。该 Data Installer 过去还管理 geoset （可以一起显示的 MapInfo 表的组合），这已不适合 Geodictionary Manager 并会导致混乱。现在应使用 Geoset Manager 应用程序来创建和管理 geoset。 Geodictionary Manager 将永远不会修改 geoset，并且也不会删除用户磁盘上的文件。在 MapX 5.0 （或更高版本）中，永久的 Geodictionary 文件不再是 MapX 运行所必需的。

注：已对命令行选项进行了更改，要求包含空格的文件名或路径名需要用双引号括起来。因此，以前可以正常使用的某些命令行现在可能无法使用了。

Geodictionary 文件

Geodictionary 包含在通常称作 geodict.dct 的二进制格式的文件中。该 geodictionary 文件可以使用注册表项 HKEY_LOCAL_MACHINE\SOFTWARE\MapInfo\MapX 5.0\GeoDictionary 定位或禁用。对于该注册表项有三种可能的有效值：

- **可以包含 Geodictionary 文件的完整文件规格** （例如 “C:\Program Files\MapInfo MapX\Maps\geodict.dct”）。结果就是数据目录设置为指向该注册表项所指定的 Geodictionary 文件的路径。内存中 geodictionary 是从 Geodictionary 文件初始化的，并且 Geodictionary 中包含的图层也可以自动与其进行匹配。添加到地图中的图层被添加到内存中的 geodictionary，这样就可以自动与其相匹配。
- **可以只包含路径说明** （例如 “C:\Program Files\MapInfo MapX\Maps\”）。数据目录被设置为该注册表项指定的路径。内存中 geodictionary 被初始化为空。添加到地图中的图层仍被添加到内存中的 geodictionary，这样就可以自动与其相匹配。
- **完全不可以存在**。内存中 geodictionary 被初始化为空。添加到地图中的图层仍被添加到内存中的 geodictionary，这样就可以自动匹配。

注：MapX 的卸载将留下 geodict.dct。这是预计的情况，因为 Geodictionary 文件可能在安装前已存在 （来自早期的 MapX 版本），并且可由其它应用程序共享。

多个 MapX 应用程序可以共享同一 Geodictionary 文件，也可以单独使用它们。为使 MapX 应用程序可以使用其自己的 Geodictionary 文件 （或者完全不使用任何 Geodictionary 文件），可以在设计阶段更改 **Map.GeoDictionary** 属性，以指向不同的注册表项。然后，在运行时初始化地图控件时， MapX 将查询以下注册表项：

HKEY_LOCAL_MACHINE\SOFTWARE\MapInfo\MapX 5.0\<Map.GeoDictionary 的值 >

此注册表项的值将在前面详细解释。 **Map.GeoDictionary** 的默认值是 “GeoDictionary”，因此 MapX 将使用该默认的注册表项。

Geodictionary Manager 始终读取并更新默认的 Geodictionary 文件 （该文件由注册表项 HKEY_LOCAL_MACHINE\SOFTWARE\MapInfo\MapX 5.0\GeoDictionary 引用）。如果您在用户界面中更改了 Geodictionary 文件的位置，则更新该注册表项以指向这一新位置。

Geodictionary 为每一注册的表 （mapinfo .tab 文件）包含一项。与每一注册表项一起存储的数据是：

- 引用该表或在图层控制对话框中显示的**用户友好的名称**。它在 .tab 文件中默认为 “Description”标记，如果不存在则为文件名。可以通过用户界面对其进行编辑。
- **索引字段的列表**。通过该用户界面，您可以选取哪些索引字段您希望 MapX 在自动数据绑定期间考虑将列与之相比较 （如果可能）。将忽略未选取的字段。在第一次注册一个表时，默认情况下最初选取所有索引的字段。
- **优化表名称**。某些表 （例如 US Counties）包含不唯一的索引列。在这一情况下，需要优化表来确定对数据的确切匹配。如果该表具有不唯一的索引列，将启用优化列组合框，并且用户将可以挑选优化的表。
- **Geoset 文件名列表**。在自动数据绑定期间选择一个表时，将该表的 geoset 列表传递给 ResolveDataBind 事件，以便 MapX 程序从中选择一个 geoset。默认情况下， MapX 将挑选第一个未处理的事件。请注意：已匹配的表始终装载，即使它不在装载的 geoset 中。此外，在 Geodictionary 中没有为某一表列出任何 geoset 并不是一个错误。在该情况下，只装载该表。最后，只能将与 Geodictionary 文件位于同一目录中的 geoset 文件添加到该列表中。

GeoDictionary Manager 的用户界面

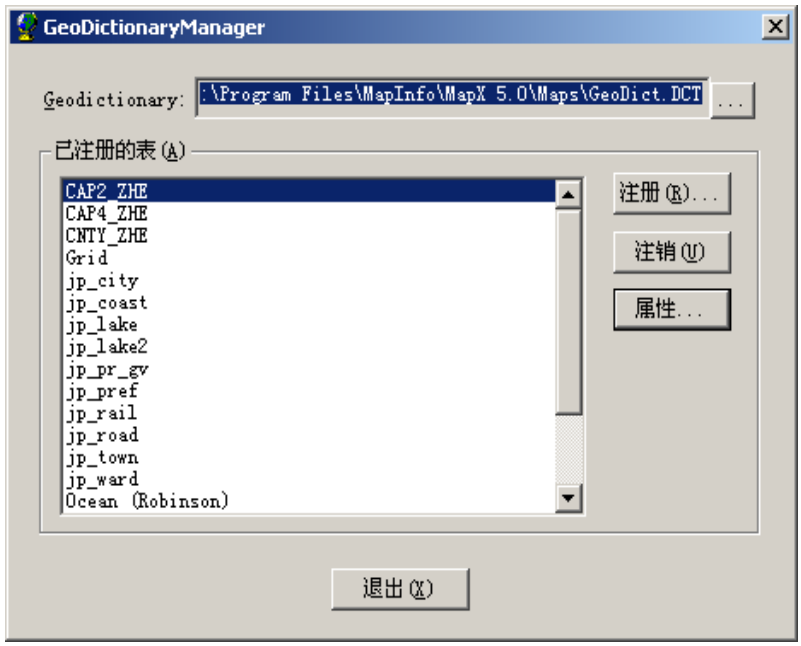
本节描述 Geodictionary Manager 的用户界面。

运行 GeoDictionary Manager

要在您希望手动注册图层时运行 GeoDictionary Manager：

1. 开始 -> 程序 -> MapInfo MapX vX.0 ->
2. 从 Windows 的 “打开”对话框中选择 “Geoset”。
3. 在 “工具”菜单中，选择 “运行 GeoDictionary Manager”。

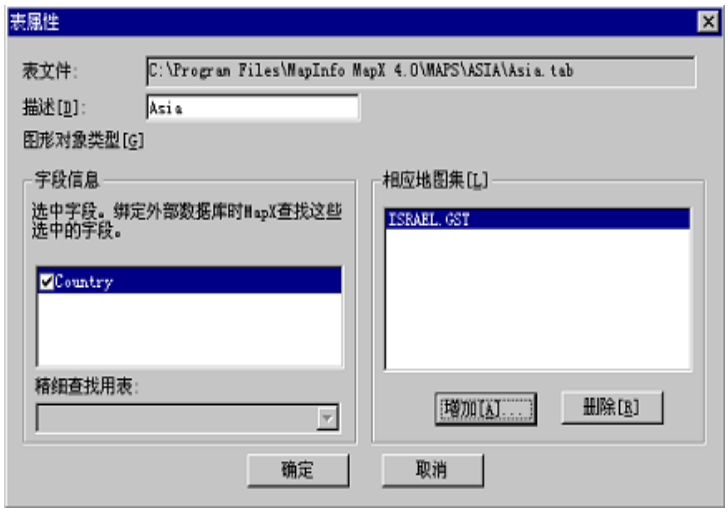
   将显示 “Geodictionary”对话框。



部分	说明
Geodictionary	Geodictionary 编辑框是只读的编辑框，包含指向所管理的 Geodictionary 的完整路径。这一信息可以通过在注册表中查找 MapX 项来获得。通过 Geodictionary 编辑框右侧的按钮，用户可以浏览要管理的其它 Geodictionary。浏览到其它 Geodictionary 将更改 MapX 的注册表项，以便所有随后的 MapX 会话都将使用新的字典。该按钮还可用于更改存储于注册表项 “HKEY_LOCAL_MACHINEOFTWAREapInfoap X.0earchPaths”中的默认 MapX 搜索路径。
已注册的表	“已注册的表”列表框包含在 Geodictionary 中注册的所有表的用户友好名称的列表。
注册	这将显示常见的文件打开对话框，而将 “文件类型”组合框设置为 “MapInfo 表 (*.tab)”。将该表复制到数据目录 （包含 Geodictionary 的目录）中，或者将包含该表的目录添加到数据搜索路径中。在吸收了该表 （通过复制或通过将目录添加到搜索路径）后，将显示 “表属性”对话框。用户可以通过在单击的同时按住 Ctrl/Shift 键来选择多个要注册的文件。如果选择了多个表，属性按钮将被禁用。请参见 **“表属性”对话框**。

部分	说明
注销	从 Geodictionary 中删除选定的表。此 “注销”按钮将不会磁盘中删除文件。用户可以通过在单击的同时按住 Ctrl/Shift 键来选择多个要注销的文件。
属性	为选定的表显示 “表属性”对话框。请参见下面的 “表属性”对话框。

使用 “表属性”对话框为给定的表设置 Geodictionary 字段。



MapInfo 表 — 包含 MapInfo 表 （如果该表与 Geodictionary 位于同一目录中）的文件名或指向该文件的完整路径名 （如果它没有文件名）的只读的编辑框。

说明 — 提供用来更改表的用户友好名称的机制。该控件在 .TAB 文件中默认为 Description 标记；如果未找到该 Description 标记，则默认为文件名，但用户可以更改该文件名。请注意，在 Geodictionary Manager 中对说明的更改将只存储在 Geodictionary 中，并且将不会在该表中反映出来。这使得 Geodictionary Manager 可以轻松地使用只读数据，例如 CD-ROM 上的数据。

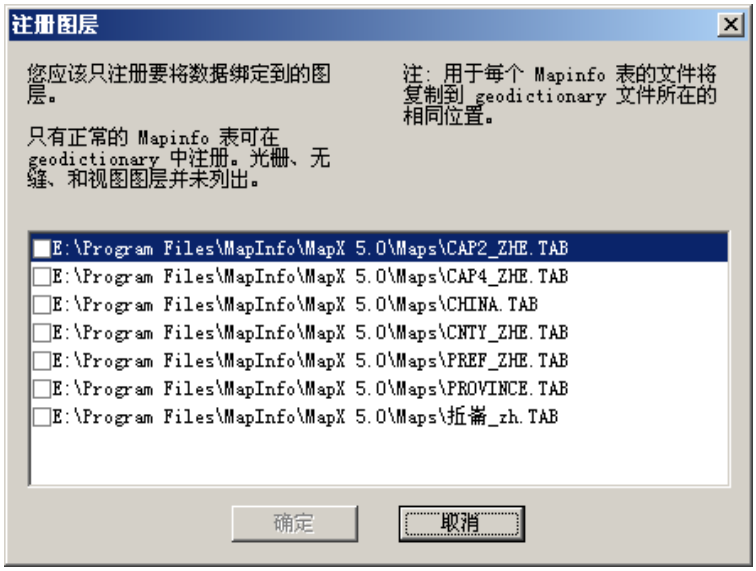
几何 — “几何”静态文本控件显示在表中存储的实体的类型。

字段信息 — 包含表中索引列的列表。如果为给定列选取了该框，则在匹配过程中将搜索该字段。

优化表	某些表 （例如 US Counties）包含不唯一的索引列。在这一情况下，需要优化表来确定对数据的确切匹配。如果该表具有不唯一的索引列，将启用优化列组合框，并且用户将可以挑选优化的表。
候选 Geoset	可能准备进行装载的 geoset 文件名的列表 （如果在数据绑定期间已选择了 MapInfo 表）。可以列出零个或多个 geoset 文件名。在自动数据绑定期间选择该表时，将该表的 geoset 列表传递给 ResolveDataBind 事件，以便 MapX 程序从中选择一个 geoset。默认情况下， MapX 将挑选第一个未处理的事件。 请注意： 始终装载该匹配的表，即使它不在装载的 geoset 中。 在 Geodictionary 中没有为某一表列出任何 geoset 并不是一个错误。在该情况下，只装载该表。 只能将与 Geodictionary 文件位于同一目录中的 geoset 文件添加到该列表中。 可以添加或删除多个候选的 geoset
添加	这将显示一个列表框，该列表框包含与 Geodictionary 文件位于同一目录中的 geoset 文件名的列表。选择一个文件名会将该文件名添加到候选 geoset 的列表中。
删除	从候选 geoset 的列表中删除选定的 geoset 文件名。

在 Geodictionary 中注册图层

如果希望 Geoset Manager 自动注册您的图层，则选择 “工具” > “注册图层”。将显示 “注册图层” 对话框。



选取您要注册的图层，然后单击 “确定”。

命令行选项

Geodictonary Manager 命令行提供向 Geodictionary 添加表或从 Geodictionary 删除表的机制。

geosetpath

指向包含表列表的 geoset 文件的路径。通常路径是必需的；仅指定文件名是错误的。

- 使用 **geosetpath** 选项向 Geodictionary 注册该 geoset 中所有有效的 .tab 文件。（geoset 中的 Raster、 Seamless 和 View 表将被忽略。）
- 使用 **geosetpath /remove** 选项从 Geodictionary 注销 geoset 中所有有效的标记文件。

tablepath

指向将添加到 Geodictionary 中的 MapInfo 表的路径 （该选项不具有任何参数）。

- 使用 **tablepath** 选项来向 Geodictionary 注册一个文件。
- 使用 **tablepath /remove** 选项从 Geodictionary 注销一个文件。

geosetfile

只是位于与 Geodictionary 文件相同的目录中的 geoset 文件的文件名。

- 使用 **tablepath /geoset=geosetfile** 选项向 Geodictionary 注册一个文件并将 geosetfile 添加到要匹配的候选 geoset 列表中。
- 使用 **tablepath /geoset=geosetfile /remove** 选项从其要匹配的候选 geoset 列表中删除 geosetfile。

commandfilepath　在文件的每一行包含有效的 migm30 命令行的文本文件的完整路径名。

- 使用 **/file=commandfilepath** 选项处理命令行的文本文件，在该文件的每一行上有一个命令行。不要在文件的每一行上包括 “migm30.exe”。
- 使用 **/file=commandfilepath /remove** 选项处理命令行的文本文件，在该文件的每一行上有一个命令行。 **/remove** 选项将在执行前添加到每一命令行上。不要在文件的每一行上包括 “migm30.exe”。

具有空格的路径或文件名必须用双引号括起来。如果可能， Geodictionary Manager 将无提示运行命令行。尝试多次注册一个文件并非错误。注销当前未注册的表并非错误。

使用 Geoset Manager

使用 Geoset Manager 可以很容易地保留供您使用的地图图层及其设置的集合。Geoset 可以帮助您避免在每次要处理图层时要分别打开和显示这些图层的耗时的工作。

计算机地图被组织成多个图层。将图层视作在顶部彼此堆叠的透明层。通过打开地图来开始建立一个 geoset。每一地图图层都包含不同的地图对象，例如区域、点、线条和文本。

例如，一个图层可以包含州边界，第二个图层可以具有表示首都的符号，第三个图层可以包含文本标签。通过将这些图层堆叠到其它图层的顶部，就可以开始构建完整的地图。

一旦您创建了自己的 geoset 后，就可以定制图层显示的方式，并且可以添加、删除图层或对它们进行重新排序。

从 Geoset Manager，使用示例 geoset，或者创建您自己的 geoset。

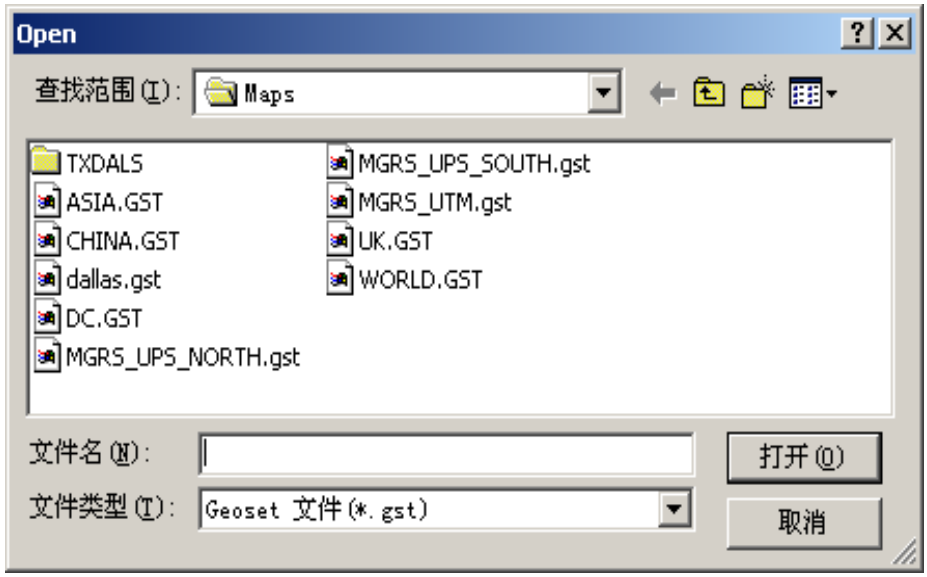
若要启动 Geoset Manager，从 “开始”菜单上的 “MapX 程序组”中选择 “Geoset Manager”。

注：在 MapX 的试用版中， Geoset Manager 将只能在开发人员的计算机上使用。

打开现有的 Geoset

MapX 包括一些示例 geoset。您可以在它们的基础上建立自己的 geoset，也可以照原样使用它们。要显示现有 geoset：

1. 选择 “文件” > “打开 Geoset”。将显示 “打开”对话框。

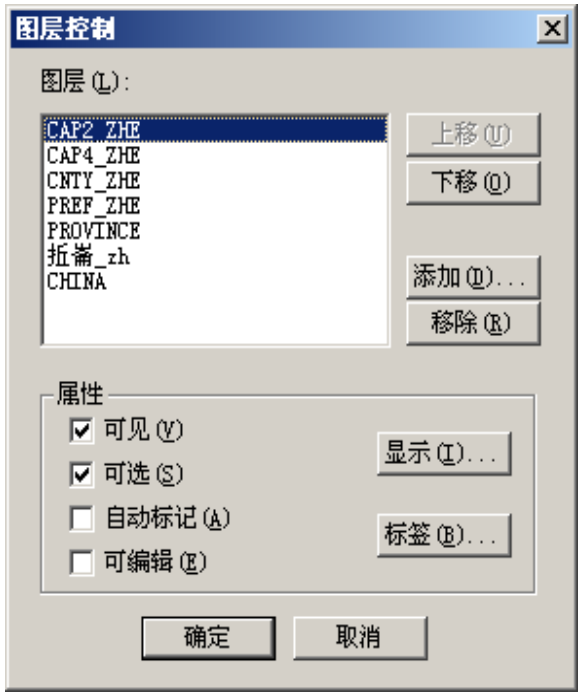


2. 选择您所需的 geoset 并单击 “打开”。将显示该 geoset。

从显示的对话框中，您可以使用 “地图”菜单或 “视图”菜单中的项更改各种图层设置。您还可以插入其它 geoset 以随当前 geoset 一起显示。然后，将这些新设置保存到当前 geoset 或通过 “另存为”命令创建新的 geoset。我们将在随后的几节中详细介绍这些项。

创建 Geoset

1. 选择 “文件” > “新建 Geoset”。将显示 “图层控制”对话框。
2. 单击 **“添加”**将显示 “打开”对话框。
3. 选择您要用作 geoset 的一部分的图层。按住 Ctrl 键的同时进行选择可以标记多个图层。这些选定的图层将显示在 “图层控制”对话框中。



从该对话框中，您可以设置显示和标签属性，重新排序图层显示的方式，删除或添加其它图层并设置图层是否可见，包含自动标签，或者将图层设为可选的。请参见下一节中的图层控制的说明。您可以在显示初始 geoset 后在以后编辑这些属性。

4. 单击 “确定” 可显示该 geoset。将显示您新建的 geoset。

使用后面几节中提供的信息可以控制和定制 geoset 中的图层。在您创建好 geoset 后，需要保存它。

要保存 geoset：

1. 选择 “文件” > “保存”。将显示 “另存为” 对话框。
2. 输入名称，然后单击 “保存”，将用您指定的属性保存这些图层。

使用 “地图” 菜单控制图层

使用 “地图” 菜单中提供的选项可以控制您的地图显示的方式。

放大

使用 “放大” 工具来获得地图或图层的更近的区域视图。

1. 选择 “地图” > “放大” 或单击菜单栏上的 “放大” 按钮，或者在 Geoset Manager 窗口中单击鼠标右键并选择 “放大”。

   将显示 “放大” 鼠标图标。

2. 在您要放大的区域的中心单击 “放大” 光标，通过两倍的线性因子放大该区域。该点在放大的视图中将是地图的中心。重复此步骤，直到您具有适当的放大级别为止。

要放大矩形区域：

1. 选择 “地图” > “放大”或单击菜单栏上的 “放大”按钮，或者在 Geoset Manager 窗口中单击鼠标右键并选择 “放大”。

   将显示 “放大”鼠标图标。

2. 通过对角拖动 “放大”鼠标图标在地图或布局中拖出一个矩形。该矩形内的区域将被放大。

缩小

使用 “缩小”工具来获得地图或图层的更宽的区域视图。

1. 选择 “地图” > “缩小”或单击 “缩小”按钮，或者在 Geoset Manager 窗口中单击鼠标右键并选择 “缩小”。

   将显示 “缩小”鼠标图标。

2. 在您要缩小的区域的中心单击 “缩小”鼠标图标，通过两倍的线性因子扩大该区域。该点在缩小的视图中将是地图的中心。重复此步骤，直到您具有适当的缩放级别为止。

要缩小矩形区域：

1. 选择 “地图” > “缩小”或单击 “缩小”按钮，或者在 Geoset Manager 窗口中单击鼠标右键并选择 “缩小”。

   将显示 “缩小”鼠标图标。

2. 通过对角拖动 “缩小”鼠标图标在地图或布局中拖出一个矩形。矩形内的区域被缩小，这样可以显示地图的更多内容。

平移

使用 “平移”来重新确定地图在其窗口中的位置。

要移动或调整地图显示：

1. 选择 “地图” > “平移”或单击 “平移”按钮，或者在 Geoset Manager 窗口中单击鼠标右键并选择 “平移”。

   将显示 “平移”鼠标图标。

2. 单击该地图的一个区域。

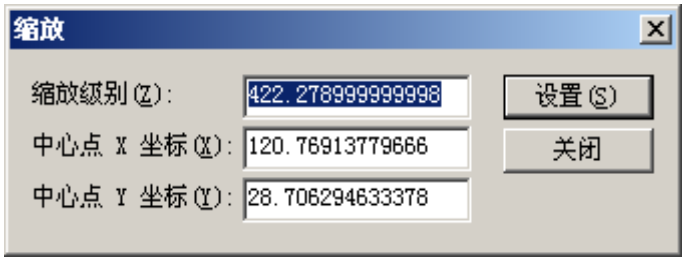
3. 在按住鼠标左键的同时，沿适当的方向拖动该地图。在松开鼠标按钮后， Geoset Manager 将在其新位置中重新绘制该地图。

缩放为

在地图上缩放为特定的 X 和 Y 坐标并设置缩放级别。

选择 “地图” > “缩放为”。

将显示 “缩放为”对话框。



输入您要缩放为的坐标。

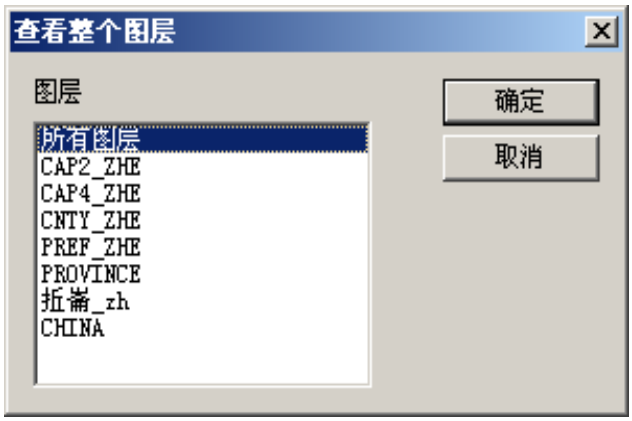
查看整个图层

使用 “查看整个图层” 可以缩放并显示地图中的整个图层或所有图层。

如果地图包含覆盖不同地域范围的图层，则使用此选项。例如，您可以具有包含纽约郡、高速公路、ZIP 编码和尤蒂卡城的街道的地图。如果您选择 “所有图层”，将会缩小视图以显示整个地图。但是，如果您只是对查看尤蒂卡城的街道感兴趣，则选择尤蒂卡城的街道图层。则视图将会放大以显示这些街道。

要显示整个地图或地图图层：

1. 选择 “地图” > “查看整个图层”，或者使用鼠标右键在 “地图” 窗口上右击并选择 “查看整个图层”。将显示 “查看整个图层” 对话框。



2. 选择要显示的具体图层或所有图层。单击 “确定” 将查看图层。

图层控制

使用 “图层控制” 对话框可以：

- 更改地图图层在活动窗口中的显示
- 确定显示、删除、添加、成为可选择、缩放分层、加标签和设置哪些图层
- 更改地图图层的顺序。

要访问 “图层控制”：

- 选择 “地图” > “图层控制”，或者单击 “图层控制”按钮，或者使用鼠标右键在地图上单击并选择 “图层控制”或创建新 geoset。

“图层控制”对话框控制地图的显示方式。一些重要信息包括：

- “图层控制”对话框在当前地图窗口中显示图层的列表，并指示每一图层是否显示、可选择或自动加标签。
- 图层包括数据表、光栅图或在地图窗口中堆叠的主题地图。
- 图层在 “图层控制”对话框中的顺序就是图层在地图窗口中的顺序。例如，在边界图层放置于点图层之下时，这些点保持可见。
- 若要使用一个图层，在该图层上单击来选择它。通过在适当的框中选取或清除可以控制其设置，以显示图层、使图层可选择或自动给图层加标签。

部分	说明
上移	上移一个或多个图层。
下移	下移一个或多个图层。
添加	向地图添加一个或多个图层。从表中显示的列表框中选择。
删除	从地图中删除一个或多个图层。
可见	指示图层是否可见。为您要在地图中保持可见的一个或多个图层选取此复选框。有时候，您可能要在 geoset 中包括多个图层以用于计算目的，并且不希望它们在 geoset 中实际显示。
可选择	指示图层是否可选择。为您要成为可选择的一个或多个图层选取此复选框。如果您要选择多个对象或给多个对象加标签，则图层必须是可选择的。要想使一个图层成为可选择的，必须显示该图层。可以同时使多个图层成为可选择的。但您只能一次从一个图层上选择。
自动加标签	选取此选项可以自动给地图加标签。这些标签取自在 “标签选项”对话框的 “标签”部分中指定的表列。
显示	这将显示 “显示选项”对话框。使用该对话框可为地图图层指定显示属性。
标签	这将显示 “标签属性”对话框。使用该对话框可为地图图层指定标签属性。

若要为地图图层指定显示属性，从 “图层控制”对话框中单击 “显示”按钮。

显示模式

部分	说明
覆盖样式	选取该选项后将覆盖图层的默认样式。
样式按钮	在选取了 “样式覆盖” 选项后启用。这将显示适合于所选图层中包含的对象类型的按钮：“区域”、“线条”、“符号” 或 “文本”。单击可以显示适当的对话框。

缩放分层

部分	说明
在缩放内显示	选取该选项可以激活缩放分层。通过缩放分层，您可以设置选定的图层将可见的最小和最大距离。例如，如果您想在近于 3 英里时看到地图上的特定街道，则将最小缩放比例设置为 0 并将最大缩放比例设置为 3。
最小缩放比例	指定选定图层可见的最小距离。
最大缩放比例	指定选定图层可见的最大距离。

自动加标签

要使用该图层中的信息给地图中的图层自动加标签：

1. 从 “图层控制” 对话框中，选择您要加标签的图层；然后选取 “自动加标签” 复选框。
2. 单击 “确定”。

如果装载了数据集并且指定了该数据集的字段，则地图通过来自 “标签选项” 对话框的 “标签” 部分中指定的表列的标签重新显示。

一次每个表只显示一列。

在 “标签属性” 对话框中，可以更改标签的可见性、内容、字体、文本颜色、线条样式和位置。

“标签属性” 对话框

“标签属性” 对话框确定标签的可见性、内容、字体、文本颜色、线条样式和锚点。这些设置用于使用 “标签” 工具自动加标签和以交互的方式加标签。在使用 “标签” 工具时，标签放置于您通过单击鼠标按钮指定的位置。

要访问 “标签属性” 对话框：

1. 选择 “地图” > “图层控制”。
2. 单击 “标签” 按钮。将显示 “标签属性” 对话框。

部分	说明
标签用于：	选择您要在标签中反映的数据集名和字段名。
可见性：	
启用	取消该选项将允许显示标签。
禁用	选取该选项将禁止显示标签。

部分	说明
在范围内显示	选取该选项后可以激活缩放加标签。通过缩放加标签，您可以设置标签可见的最小和最大距离。例如，如果您只想在近于 3 英里时看到地图上的特定标签，则将最小缩放比例设置为 0 并将最大缩放比例设置为 3。标签大小不会随着缩放或比例的更改而更改。
最小缩放比例	指定标签可见的最小距离。
最大缩放比例	指定标签可见的最大距离。
允许重复文本	选取该选项可以多次在地图上放置相同的标签。
允许覆盖文本	允许显示具有相同文本的多个对象。
最大标签	输入将显示的最大标签数；标签是从指定的表中按它们在该表中输入的顺序选择的。例如，如果指定的表是 States 表并且您输入 10，则在该表中 （在视图中）列出的前 10 个州将加标签。
样式：	
“文本样式”按钮	单击 “文本样式”按钮将显示 “文本样式”对话框。请参见 “文本样式”对话框。
标签线条	选择要将标签添加到锚点的线条类型，或者没有任何线条类型。
无	不显示具有标签的线条。
简单	通过使用将标签连接到对象的矩心的简单线条来创建标注。标签线条将在您从最初创建的地方移动标签后显示。
箭头	通过使用将标签连接到对象的矩心的箭头线条来创建标注。标签线条将在您从最初创建的地方移动标签后显示。
位置：	
锚点	单击一个图标可以选择相对于标签锚点的标签位置。菱形字符表示标签锚点；矩形表示标签。选定框的边框是粗体。
使用线条旋转标签	选取该选项可以旋转文本以保持与线段平行。对于点区域将忽略此设置。
标签偏移量	指定标签应距锚点放置的点数 （文本大小的度量单位）。

投影

如何展平地球的曲面以便您可以在扁平的纸张或 （接近）扁平的计算机屏幕上绘制地图？您可以使用投影。投影是定义如何使对象扁平的系统。您可以以多种不同的投影来显示地图。

在您将对象从球形世界转换为相对扁平的计算机屏幕时，注定会有一些扭曲。

当来自球面的对象在平面上显示时，投影是一种减少扭曲发生的方法。有多种不同类型的投影，每一类型均设计为减少给定区域的扭曲量。

要选择一种投影：

选择 “地图” > “投影”，或者在地图上单击鼠标右键并选择 “投影”。

使用 “视图”菜单

使用 “视图”菜单中的 “选项”可以指定状态栏或工具栏是否显示并更改地图距离单位或坐标。

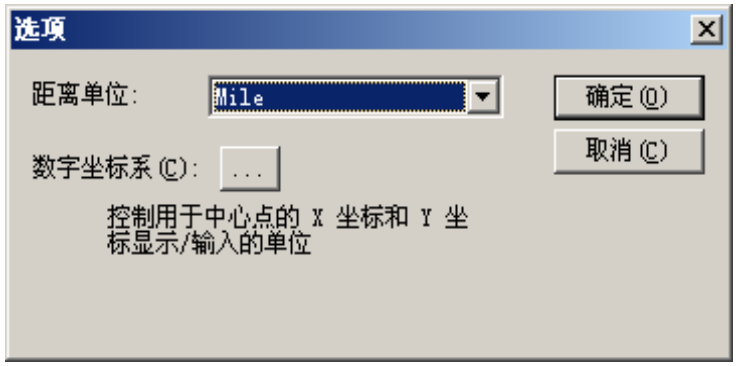
工具栏显示

若要显示或隐藏 “地图”工具栏、“Geoset 名称”工具栏或状态栏，应选取或清除适当的菜单选项。

选项

使用 “选项”对话框设置地图所使用的距离单位和数字坐标。

要设置选项，选择 “视图” > “选项”。将显示 “选项”对话框。



部分	说明
距离单位	选择用于地图的距离单位。
数字坐标	选择 Geoset Manager 将用来查找 X 和 Y 坐标的数字坐标。坐标系统是一组参数，指示如何为对象解释位置坐标。几何对象中的每一点均由一对数字表示。这些数字是用于这些点的坐标。

工具和 GeoDictionary Manager

使用 “工具”菜单中的选项来访问 GeoDictionary Manager。通过 GeoDictionary Manager，您可以在 geodictionary 中注册各图层。在 geodictionary 中注册的图层可在其它 MapX 会话中用于数据绑定。

11 章

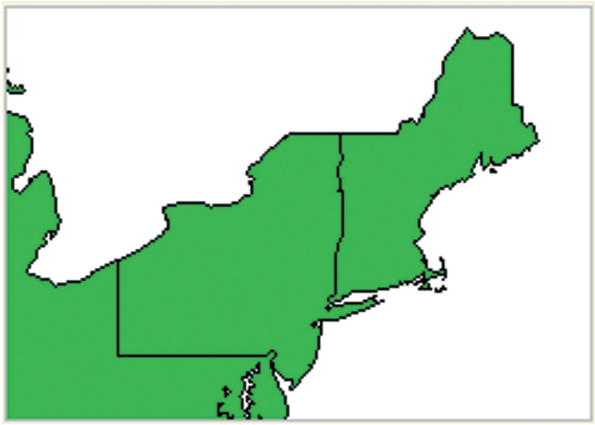
- 什么是逐层细化图层?
- 应当了解的术语和概念
- 如何开发逐层细化应用程序
- 准备逐层细化图层
- 创建逐层细化工具
- 重置逐层细化图层
- 逐层细化图层的限制和要求
- 详细信息 ...

使用逐层细化图层

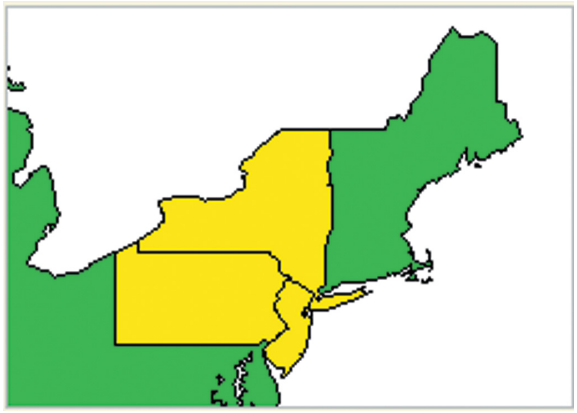
在本章中，您将了解一种特殊类型的地图图层，称为逐层细化图层，用户可以使用该图层执行 “逐层细化”分析和查阅。如果地图中包含逐层细化图层，用户可以点击地图的某个部分，查看相应地区的详细信息。逐层细化图层提供直观易用的界面，使用户可以通过点击查阅数据。

什么是逐层细化图层?

逐层细化图层是一种地图图层，具有特殊的图元。逐层细化图层使用户可以通过点击地图中需要了解详细信息的部分，以分层的方式查阅地图。

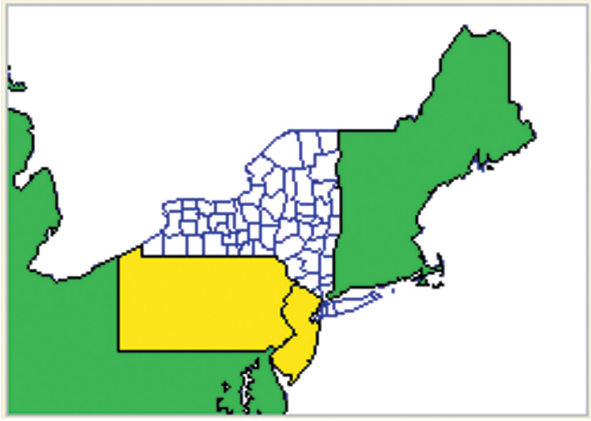


地图初次显示在屏幕上时，逐层细化图层就象一个普通的地图图层。例如，该逐层细化图层显示销售地区。



如果用户选择逐层细化工具并单击地图中的某个图元，该图元将由许多更小的地区取代。例如，如果单击某个销售地区，该地区的图元将由组成该地区的各州 （省）界取代。单击是一种发出请求的基本方式，表明 “为我显示地图该部分的详细信息”。

根据应用程序和逐层细化图层的建立方式，用户也许可以反复逐层细化。每次连续单击，地图就会显示用户单击的区域更详细的信息。单击州 （省）界可能会显示组成该州 （省）的市 （县）界。



要减少显示的详细信息量，用户可以选择其它工具 （“汇总”工具），然后单击逐层细化的区域。较小的逐层细化的区域将由较大的区域取代，如果需要，可以撤销逐层细化操作。

逐层细化图层通过一组表建立。您需要为逐层细化层次结构中的每一 “层”提供一个表，并提供一个特殊的表，定义各层的层次关联方式。不过，用户不会感觉到逐层细化图层的复杂性。逐层细化图层在 MapX 地图中作为单个图层显示。如果用户显示 “Layer Control”对话框，逐层细化图层将在图层列表中显示为单个项。

应当了解的术语和概念

要了解本章的其它内容，需要先了解逐层细化地图的某些术语和基本概念。

- 要支持逐层细化地图，地图中必须包含逐层细化图层。这是一种特殊的图层，由两个或两个以上独立的表组成。逐层细化图层将不同表中的对象作为地图中的一个图层显示。
- 要使逐层细化图层有效，必须至少包含两层，以便组成层次结构。例如，假设逐层细化图层显示州 （省）界，用户可以单击某个州 （省），显示该州 （省）的所有市 （县）界。在前面的示例中，州 （省）界代表一层，市 （县）界代表另一层。这两层组成层次结构。每个州 （省）包含许多市 （县）。在此例中，我们称州 （省）界为父图元，每个市 （县）界为子图元。
- 通常，用户可以选择逐层细化工具，然后单击地图中需要了解详细信息的部分，逐层细化地图的某个部分。要实现逐层细化工具，应使用 CreateCustomTool 方法创建自定义工具。
- 使用逐层细化工具触发 ToolUsed 事件。在该事件中，通过展开用户单击的图元来处理用户的逐层细化操作。展开图元分为两个步骤：首先，获取将取代用户单击的图元的子图元列表，并将这些子图元加入图层 （调用 Layer.DrilldownAddFeatures 方法）。然后，删除用户原来单击的图元。也就是说，图元由其子图元取代。

- 任何逐层细化应用程序均需要层次结构管理器，该软件组件了解逐层细化图层中各层之间的层次关系。用户通过单击展开图元时，层次结构管理器作为程序的一部分，确定需要添加的子图元。
- ToolUsed 事件过程就是层次结构管理器的一个简单示例。用户使用自定义的逐层细化工具单击地图时，单击事件由 ToolUsed 事件过程处理；该过程中执行的代码确定要添加哪些图元来展开该地图图元。
- 可以建立逐层细化图层，使层次结构中的指定层可以使用两个或两个以上的备用表。例如，假设对于每个州 （省）界，可以使用两组备用的子图元：州 （省）在展开时可以显示一组市（县）界，也可以显示一组电话交换机号码。此类逐层细化图层仍被认为是包含两层，一个父层和一个子层。不过，在子层可以在两组备用图元之间进行选择 （市 （县）与电话交换机）。使用这种安排，您的应用程序必须确定要显示的子层，系统可能会提示用户选择。

如何开发逐层细化应用程序

逐层细化应用程序需要进行许多设置和准备。主要步骤汇总如下 （本章后文会提供每个步骤的详细信息）。

1. 获取您在建立多层逐层细化图层时要使用的各个表。表可以使用 MapInfo Professional 创建，也可以向 MapInfo Corporation 或第三方供应商购买。
2. 创建一个包含特殊列和特殊元数据的新的空逐层细化表 （.tab 文件）。元数据为每个组件表指定一个层名，同时标识组件表中的重要列 — ID 列和标题列。
3. 将逐层细化表加入地图 （例如，将逐层细化表加入您使用的 Geoset，或通过 Layers.Add 之类的方法将逐层细化表加入地图）。
4. 在应用程序中添加用户界面元素 （例如工具栏按钮），使用户可以选择逐层细化工具并单击地图进行逐层细化。
5. 在应用程序中添加代码，响应用户对逐层细化工具的使用 （也就是说编写层次结构管理器）。该代码需要检测用户已选择的图元；确定哪些子图元将取代该图元；调用不同方法（DrilldownRemoveFeatures、 DrilldownAddFeatures）展开或收缩地图图元。

准备逐层细化图层

要建立逐层细化图层，需要提供两个或两个以上的一组 MapInfo 表。特别是需要：

- 一个 MapInfo 表用于逐层细化图层中的每个详细信息层。这些表称为**组件表。**
- 另一个空表，包含描述组件表的特殊元数据。该表称为**逐层细化表。**

例如，假设您要显示州 （省）界，但是允许用户通过单击州 （省）显示该州 （省）的市 （县）界 – 一个两层的逐层细化图层。您将需要三个表：一个州 （省）界 （组件）表，一个市 （县）界 （组件）表，一个逐层细化表。

注：逐层细化表是 “空的”，因为其中没有自己的永久数据 （元数据除外）。在地图中显示逐层细化图层时， MapX 会创建一个临时表，然后将组件表中的图元复制到该临时表中。逐层细化图层中显示的图元实际上是组件表中图元的副本。在 MapX 应用程序终止时，将丢弃该临时表，逐层细化表只剩下 .tab 文件。

组件表的要求

逐层细化图层中的每个图元必须具有标识关键字 （可能是 “New York”之类的字符串）。单个层中的所有关键字必须是唯一的；例如，州 （省）界层只能包含一个名为 “Washington”的州。不过，图元的标识关键字不需要在逐层细化图层的所有其它层中均是唯一的。例如，如果逐层细化图层包括州（省）界和市 （县）界，则可以同时包含 “Washington”州和 “Washington”市。

逐层细化表的要求

典型的 MapInfo 表由一组文件组成；例如，World 表由文件 World.tab、World.map、World.id、World.ind 和 World.dat 组成。逐层细化表与其它表不同，因为逐层细化表由单个文件组成：filename.tab。

.tab 文件是一个文本文件，可以在任何文本编辑器中查看或编辑。

逐层细化表的 .tab 文件必须定义具体的列和元数据关键字，如下所述。在阅读以下要求时，有关示例请参阅 “逐层细化表示例”。

逐层细化表必须定义三个标准列：**关键字、层**和**标签。**所有三个列均是字符 （字符串）列，宽度为 32 个字符。

逐层细化表必须包含一组元数据关键字。元数据关键字的语法如下：

- 关键字 begin_metadata 标记 .tab 文件中元数据部分的开头。
- 每行元数据包含两个元素：关键字和值。例如 “\IsDrilldown”关键字的值为 “True”。所有关键字和值均使用双引号引起来。
- 逐层细化表必须包含 \IsDrilldown 关键字，该关键字的值必须为 True。
- 每个关键字以 “\”（反斜线）开头。
- 元数据关键字可以在层次结构中嵌套。层次结构中的每一层以反斜线 (\) 字符标记。关键字的值限制在 239 个字符以内。
- 逐层细化表包括 \DDMap\ComponentMaps\ 关键字层次结构。在该层次结构中为每个组件表指定四个元数据关键字：

组件表元数据关键字

关键字	说明
File	必需的关键字，用于标识组件表的路径和文件名。
LevelID	必需的关键字，用于定义该组件表的标识符。 **例如**：如果该表包含州 （省）界，可以使用 “States”作为关键字值。在调用 **DrilldownReset** 之类的方法时，将 “States”作为层参数传递。
FeatureIDCol	标识组件表中包含唯一逐层细化关键字的列号。可选关键字；如果省略，则使用列号 1。
FeatureCaptionCol	标识组件表中将用于标记的列号。可选关键字；如果省略，则使用列号 1。

例如，“逐层细化表示例”包含关键字层次 “\DDMap\ComponentMaps\One\LevelID”和 “\DDMap\ComponentMaps\Twox\LevelID”。注意， **DDMap**、 **ComponentMaps** 和 **LevelID** 是关键字层次中必需的标准部分，而 **One** 和 **Twox** 是可自定义的部分。可以使用任意喜欢的关键字名称代替 **One**、 **Twox**、 **Twoxx** 等；这些关键字名称只是为了区分每个组件表的关键字集，没有任何其它用途。

在 \DDMap\HierarchyManager\ 关键字层次中，有三个附加关键字：

关键字	说明
IsDLL	在版本 4 中不使用；保留供以后使用。布尔值指示符； “TRUE”表示该逐层细化图层使用 DLL 作为层次结构管理器。
ID	在版本 4 中不使用；保留供以后使用。 DLL 的名称或 CoCreate 的 GUID。
InitialLevel	初次加载 / 显示逐层细化图层时要显示的初始组件表。可选关键字；可以包含空的逐层细化图层。

创建逐层细化工具

应用程序的用户界面应使用户可以逐层细化地图的各部分。通常，用户通过选择逐层细化工具 （例如工具栏按钮）并单击地图，进行逐层细化。

您可以使用 CreateCustomTool 方法实现逐层细化工具。 CursorConstants 集包含两个光标，专门用于逐层细化应用程序：miDrilldownExpandCursor 和 miDrilldownContractCursor。例如：

```
' Drilldown Expand Tool
Map1.CreateCustomTool 1, miToolTypePoint,
miDrilldownExpandCursor, miDrilldownContractCursor,
miDrilldownContractCursor
```

该示例创建一个可以同时用于**逐层细化和汇总**操作的工具。默认情况下，该工具充当逐层细化工具，显示 “展开”光标 （带 + 符号的光标）。如果用户在单击时按住 SHIFT 或 CTRL 键，光标会更改为 “收

缩”光标（带 - 符号的光标）。这种安排使用户不必将光标一直移动到工具栏上即可切换工具；内置的 ZoomIn 和 ZoomOut 工具作用相同。

创建了自定义工具后，可以通过设置 CurrentTool 属性激活工具。

```
Map1.CurrentTool = 1
```

每次使用自定义逐层细化工具均会触发 ToolUsed 事件。在 ToolUsed 事件过程中，您将需要执行产生逐层细化行为的代码。该过程主要分为四个步骤：

1. 使用 SelectByPoint 或 SearchAtPoint 之类的方法确定用户单击的地图图元。
2. 确定应取代用户单击的图元的子图元集。例如，可以使用一个或多个嵌套的 Case 语句确定哪些子图元取代所选的父图元。
3. 调用 DrilldownAddFeatures 方法将子图元加入地图。
4. 调用 DrilldownRemoveFeatures 方法从地图中删除父图元（用户单击的图元）。

注：这些添加 / 删除操作不会对组件表进行任何修改；您不是在“编辑”表。在使用 DrilldownAddFeatures 方法添加图元时，唯一的结果是将图元复制到当前可见的图元集中。

创建汇总工具

实现了逐层细化工具后，可能需要为用户提供一个汇总工具 ó 与逐层细化工具作用相反的工具。

创建汇总工具的方法与创建逐层细化工具大致相同。您将需要相同的方法 （DrilldownAddFeatures 和 DrilldownRemoveFeatures）。区别在于不是添加子图元并删除父图元，而是恰好相反，添加父图元并删除子图元。

重置逐层细化图层

可以通过调用 DrilldownReset 方法 “重置”逐层细化图层。重置逐层细化图层将清空整个图层，然后使用其中一个组件表中的图元重新初始化该图层。

例如，可以在用户界面中加入 “重置”按钮。用户单击该按钮时，将调用 DrilldownReset 方法来恢复逐层细化图层的原始状态。

加入重置按钮是个很好的方法，因为这样可以使用户快速方便地将逐层细化图层恢复到相似状态。

逐层细化图层的限制和要求

某些限制适用于逐层细化图层，参见下表中的小结。

- 逐层细化图层中的每个图元必须包含在该组件表中唯一的 ID （尽管 ID 不需要在组成逐层细化图层的所有组件表中唯一）。
- 不能在逐层细化图层中使用光栅图基表。
- 在使用 Themes.Add 创建主题时，计算包含许多行的图层 （例如逐层细化图层或服务器图层）的范围时可能会需要一段时间。 Add 方法的 **ComputeTheme** 参数可以为任意主题类型创建非计算主题。非计算主题使您不必自动计算范围即可创建主题。然后可以自己创建范围。对于逐层细化图层和服务器图层，这种方法速度较快。
- 逐层细化图层不 “记忆”各逐层细化层的状态 （例如哪些图元已展开）。如果希望应用程序恢复上次使用地图时的准确状态，需要通过编写代码，在退出时存储地图的逐层细化设置，在启动时恢复逐层细化设置。
- 尽管可以编辑逐层细化图层中的图元，但是所作的编辑不会保存，组件表不受编辑的影响。在编辑逐层细化图层中的图元时，不是在修改组件表，而是在修改组件表中的图元的临时副本。

详细信息 ...

有关逐层细化图层的详细信息，请参见以下内容。

相关方法和属性

Layer.DrilldownAddFeatures 方法将图元加入逐层细化图层。

Layer.DrilldownRemoveFeatures 方法将图元从逐层细化图层中删除。

Layer.DrilldownReset 方法清空整个逐层细化图层，然后仅显示该图层中的一个详细信息层。

Layer.GetDrilldownFeaturesByID 方法检索指定 ID 的图元。

要确定某个图层是否属于逐层细化图层，应测试 Layer.Type 属性是否返回 miLayerTypeDrilldown 值（值为 7）。

相关常数

CursorConstants 集包括两个常数 （miDrilldownExpandCursor 和 miDrilldownContractCursor），专门为逐层细化 / 汇总工具设计。在 Map.CreateCustomTool 方法中使用这些常数。

12 章

- 访问远程空间数据
- 通过 LayerInfo 对象使用 Layers.Add 方法
- 通过 .tab 文件访问远程表
- 通过 X/Y 列对 DBMS 数据进行地图绘制
- Oracle 支持
- DBMS LayerInfo 参数
- DBMS 连接字符串格式
- MapX DBMS 图层查询
- 访问属性数据
- 性能问题
- 使用缓存
- MapInfo 地图目录
- 使 DBMS 表成为可以进行地图绘制的表
- 故障排除

访问 DBMS 中的数据

MapX 提供空间服务器访问。这是一个强大的功能，可以让开发者连接到存储在空间服务器中的活动数据，如在 Mircosoft SQL Server、Oracle、Informix、DB2 数据库上运行的 MapInfo SpatialWare 或 Oracle8i Spatial 数据库。通过空间服务器，公司可以在企业数据库中存储它们的地图数据，以便集中管理和安全。 SpatialWare 等空间服务器提供了高级查询处理功能，提高了一个组织的空间数据在服务器上的性能。

访问远程空间数据

您可以通过不同的 DBMS 服务器使用 MapX 访问作为图层数据的数据。服务器包括：

- MS Access/SQL Server/ 其它 ODBC 数据源。
- 空间服务器，如 SpatialWare for Microsoft SQL Server、 SpatialWare Informix DataBlade 和 SpatialWare DB2 Extender。
- MapInfo Geocoding DataBlade for Informix。
- Oracle8i Spatial。

您可以使用 **Layers.Add 方法**从 DBMS 的数据中添加一个图层。有两种技术：

- 需要在运行时动态计算查询时，通过 **LayerInfo 对象**使用 Layers.Add 方法。这对于新输入数据的情况比较合适。例如，您有一个应用程序，可以确定生成用户输入收益额的商店的地理位置。通过该应用程序输入的收益额无法在设计阶段得知，因此只能在运行时添加到查询中。这可以代替以前的 AddServerLayer 方法。
- 在设计阶段已知查询时通过 .tab 文件使用 Layers.Add。例如，如果一个应用程序要求显示所有正在使用的消防栓，则可以设置查询，选择所有正在使用的所有消防栓。这可以在设计阶段设置。 .tab 文件可以直接放在 geoset 中，确保在初始化使用该 geoset 的任何应用程序时都加载该文件。

用于添加空间数据的每种技术的详细信息包含在下面几节中。一旦决定了数据处理方法，请按照相应章节中的说明进行操作。

注：MapX 5.0 不支持打开 SpatialWare 3.0 （或更低级别的表）。

通过 LayerInfo 对象使用 Layers.Add 方法

如果您的 Layers.Add 成功，但您的 DBMS 图层数据不在地图上显示，则请检查图层顺序。图层可能已经绘制，但被另一图层隐藏。您可以使用 LayerInfo 属性指定图层顺序。如果 DBMS 图层是 “地图”窗口中的第一个或唯一图层，则说明默认的缩放级别设置得过小，无法看到数据。您可以通过运行 MapBasic 工具在 MapInfo_MapCatalog 中调整服务器图层默认缩放级别 (misetmbr.mbx)。

```
Dim lInfo As Object
Dim lStr As String
Set lInfo = CreateObject("mapx.layerinfo.4")
lInfo.Type = 4   ' layer type is RDB
lInfo.AddParameter "NAME", "city_125"' layer name
lInfo.AddParameter "TOOLKIT", "ODBC"   ' use "ORAINET"  for oracle
direct access
lInfo.AddParameter "CONNECTSTRING", "DRIVER={SpatialWare 32 Bit
Driver4.00};HOST=Champagne;UUID=oracle;UPWD=secret;UID=GEORGETOWN;PWD=s
ecret;OSID=DB1;DLG=0"
lInfo.AddParameter "QUERY", "SELECT * FROM CITY_125"
lInfo.AddParameter "CACHE", "ON"
lInfo.AddParameter "MBRSEARCH", "ON"
```

```
g_map.Layers.Add lInfo
Set lInfo = Nothing
```

注：MapX 在选择并更新图元时不对其进行记录锁定。对图元的最后一次更新是已存储的图元。在多用户应用中，一个应用采用了较长的事务处理方法 / 锁定，否则应采用商务惯例来防止发生不希望发生的更新冲突。

通过 .tab 文件访问远程表

MapX 可以 “实时”访问 DBMS 数据，也可以打开 MapInfo Professional 链接的表。但是，该链接表是只读的，无法由 MapX 刷新。实际上，数据源于远程数据库中，无法反映本地链接版本中的数据。

您可以创建一个 .tab 文件，对远程地图进行访问。要使用 MapInfo Professional 生成一个 .tab 文件，请选择 “文件 -> 打开 DBMS 表”。

该 .tab 文件是一个文本文件；您可以使用任何文本编辑器创建 .tab 文件。一旦创建了 .tab 文件，则可以象您访问任何其它 MapInfo .tab 文件一样通过 Layers.Add 方法或 Geoset Manager 以编程方式访问该文件。

通过 X/Y 列对 DBMS 数据进行地图绘制

您可以从有 X/Y 坐标的 DBMS 表中添加图层。您需要创建一个地图目录并将表注册为 SpatialType 4.0，然后将两个列容器指定为坐标。应对表中的这些列进行检索。通过 ODBC 连接到 DBMS 并在查询中将新列指定为 “对象”。

您可以在 MI Upload MBX （位于 MapInfo 专业版的 Tools 目录下）中使用 MapInfo 专业版，将 MapInfo 点表 （如 CITY_125）上载到 MSAccess 或 SQL Server。

Oracle 支持

除 MapX ODBC Connectivity 组件外，您可以安装 Oracle8i Spatial。Oracle8i Spatial 是 Oracle Corporation 空间数据库的一个相对新的实现。尽管它与以前的 Oracle SDO 实现有一些相似之处，但它有显著差异。 Oracle8i Spatial 通过关系型方案维护该 Oracle SDO 实现。但是， MapInfo 不支持 Oracle SDO 关系型方案。MapX 不支持通过 OCI 与 Oracle8i 以及通过 ODBC 与其它数据库的同步连接。MapX 不使用当前 Merant 的 ODBC 驱动程序通过 ODBC 支持 Oracle8i 空间几何结构表。

支持 Oracle v8.15 和 v8.16

要在 MapInfo 中连接到 Oracle8I，必须安装 Oracle8i 客户程序。有关详细信息，请参阅 Oracle 文档。

注：如果已安装 Oracle 8.16，则任何 Oracle Spatial 表的所有插入 / 更新都将采用 Oracle 8.16 格式。因此，所有表都应在进行任何 MapX 编辑前使用 Oracle 8.16 Spatial 数据移植实用程序移植到 8.16 版的格式。

几何结构转换

下表说明了从 MapInfo Spatial 对象到 Oracle8i Spatial 的转换 (SDO_Geometry)。

从 MapInfo	到 Oracle 8i
NULL geometry	NULL
Point	1 POINT
Line	2 LINESTRING Geometry 包含一个直线字符串
Region	3 POLYGON Geometry 包含一个多边形。
Polyline	6 MULTILINESTRING Geometry 有多个直线字符串。
Region with multiple polygons	7 MULTIPOLYGON Geometry 有多个多边形。
Ellipse	NULL
Arc	NULL
Rectangle	NULL
Text Object	NULL
Rounded Rectangle	NULL
PieChart、 Barchart	NULL

下表说明了从 Oracle8i (GTYPES) 到 MapInfo Spatial 对象的转换。

从 Oracle 8i GTYPES	到 MapInfo
0 UNKNOWN_GEOMETRY （Spatial 忽略了此几何结构。）	Point
1 POINT Geometry 包含一个点。	Point
2 LINESTRING Geometry 包含一个直线字符串。	Line 或 Polyline
3 POLYGON Geometry 包含一个多边形。	Region
4 Collection Geometry 是一个异构的元素集。	Point
5 MULTIPOINT Geometry 有多个点。	Point
6 MULTILINESTRING Geometry 有多个直线字符串。	Polyline
7 MULTIPOLYGON Geometry 有多个多边形 （多个外部边界）。	Region

多维几何结构支持 （超过 2 个维）

MapX 可以读取这些数据并将其作为包含在超过 2 个维中定义的几何结构的 SDO_GEOMETRY 列的图层来绘制地图。也就是说，只有头 2 个维用作 x 和 y。

SDO_GEOMETRY 弧圆转换

圆和圆弧可以被解析为折线，分辨率为每个 360 度的圆 25 段。

不可转换的 Oracle 对象的可视化

MapX 无法转换的 Oracle Spatial 对象将在 SDO_Spatial 点或纵坐标数组的第一个 SDO_Spatial 纵坐标所在位置产生一个具有默认样式 （黑星）的 Point 对象。这样即可在它所在适当地域实现对不可转换对象的直观表示。不可转换对象包括用户定义的对象 GTypes 0,4,5，或者包含无法识别的 Gtypes、 Etypes 或解释的无效 sdo_geometries。使用 sdo_validate_geometry()，第二个类也会失败。

质心支持

MapX 将使用 SDO_POINT 作为多边形的质心值。该质心图元用于定位标签，同时也会影响对象的工具选择。如果点在区域内，则 Oracle SDO_GEOMETRY.SDO_POINT_TYPE 字段 （如果不为 NULL）可以被认为是质心图元。如果点在区域外，则会始终计算它的质心。

注：当前， MapX 中没有任何方法或工具来设置区域图元的质心，但可以读取和使用已存储的质心。

附加信息

以下 **sdo_geometry** 值不可在不丢失数据的情况下与 MapInfo 对象格式互相转换。因此，如果更新，则会丢失某些几何结构细节。

- GTYPE 0 – 用户定义 / 未知的几何结构
- GTYPE4 – 异构的元素集
- GTYPE 5 – 多点

这些类型如果尚不为 NULL，则会被认为 （根据情况）是单点 SDO_POINTTYPE 值。它们 “抓住”了有序数组中的第一个点，它会被认为是一个 NULL 几何结构。

注：包含圆或弧组件的任何其它 SDO_GEOMETRY 值已转换为直线字符串，每 360 度包含 25 条直线。

Oracle Spatial 参考支持 (SRID)

Oracle 8.16 SDO_GEOMETRY 列可以用一个 Spatial 参考系统来定义。这可以通过在 USER_SDO_GEOM_METADATA 中提供 Oracle SR_ID 来实现，也可以通过在已存储的 SDO_GEOMETRY 值中对该 SR_ID 赋值来实现。如果表包含带有赋值的 SR_ID 的 Oracle 8.16 Spatial 列，则 MapX 将能够查询和正确解释此数据。 MapInfo_MapCatalog 必须包含该 MapInfo Professional Coordsys 字符串 （如该数据的 SR_ID 中所示），因为它是当前用于解释和更新该数据的 MapInfo_MapCatalog 中的 Coordsys。

如果 Spatial 列不包含 SR_ID 值 （该值为 NULL），则 MapX 也能够通过在 MapCatalog 中定义的 MapInfo Professional Coordsys 解释该数据。

“连接”对话框临时解决办法

对于 Oracle8I，如果不指定完整的连接字符串，则不会显示任何 “连接”对话框。 DLG= 选项无效。如果没有提供 userID 和口令，则连接会失败。由于 MapInfo Professional 创建的 DBMS 表 .tab 文件不含口令 （出于安全原因），这会导致 tab 文件的打开操作失败。

临时解决办法是使用您想访问的数据库的 userID/ 口令编写一个简单的 “连接”对话框，它从用户处获得口令，使用带连接字符串的图层信息进行 DBMS 图层的 Layers.Add 操作。 Oracle 连接字符串应如下所示：

```
"SRVR=superior;UID=user1;PWD=secret;"
```

添加一个临时图层。该操作也会添加与连接池的所需连接。由于口令已经经过验证，该连接将由所有在 userid 相同的同一数据库上打开的所有 tab 文件使用。因此， .tab 文件将成功打开。

对于这些要在 geoset 中使用的 tab 文件，必须在应用程序启动时进行此登录。您可以按此方式连接 / 登录到多台服务器，从这些服务器上打开的表会成功地使用。

DBMS LayerInfo 参数

.tab 文件和 LayerInfo 对象接受相同的参数。但是，参数名的命名约定稍有不同。有关 MiLayerInfoTypeServer 类型的 LayerInfo 对象支持的完整参数列表，请参阅第 373 页的 “LayerInfo 对象”。

DBMS 连接字符串格式

ODBC 连接字符串格式

ODBC 连接字符串的格式与 MapInfo Professional ODBC/ 链接表中相同。该字符串由几个由分号 (;) 分隔的子句定义。每个子句都采用 Key=Value 格式。重要关键字如下所示。

关键字	说明
DLG=	控制 “连接”对话框显示的数字： 0 – 抑制 “连接”对话框 （MapX 主题所要求）。 1 – 显示 “连接”对话框。 2 – 显示 “连接”对话框，但仅在需要时显示 （即如果并非提供了所有所需信息） [默认值]。 **注释：**不适用于 Oracle Spatial

关键字	说明
DSN=	指定 ODBC 数据源名称。 注意：如果使用 DSN= 语法关键字，则所指定的名称必须与用户系统上所用的数据源名称匹配。注意不同用户会对同一数据源使用不同的名称。如果无法提前知道要使用的数据源名称，则使用 DRIVER= 语法关键字，而不要使用 DSN= 语法关键字。 **注释：**不适用于 Oracle Spatial
DRIVER=	指定已安装的 SpatialWare 或 IUS/UDO 驱动程序的确切驱动程序名称。用来代替 DSN= 语法关键字。示例： DRIVER={SpatialWare 32 Bit Driver} **注释：**Informix 2.80.0861 不支持 DRIVER=。 另外，它不适用于 Oracle Spatial。
UID=	如果需要，指定数据源的所需 UserId。
PWD=	如果需要，指定数据源的用户口令。

要匹配两个字符串，连接字符串中不需要口令。如果两个表在同一数据库中，则连接字符串相同。

注：连接属性 / 参数不再按顺序排列，可以使用一个对话框从现有连接池中获取连接，避免重复连接。也就是说，在 MapX 的以前版本中，如果每次使用一个对话框连接到同一数据库或者未以记录的顺序对连接字符串中的连接关键字排序，则不会共享连接，而会获得多个连接。

Oracle Spatial 连接字符串格式

以下是 Oracle Spatial 关键字。该字符串由几个由分号 (;) 分隔的子句定义。每个子句都采用 Key=Value 格式。重要关键字如下所示

关键字	说明
SRVR=	反映服务器的服务名在 Oracle8i Net8 EasyConfig 实用程序中设置的值。这是 Oracle8i 连接所需的，但不适用于 ODBC 连接。
UID=	如果需要，指定数据源的所需 UserId。
PWD=	如果需要，指定数据源的用户口令。

连接字符串示例

下面是 Oracle Spatial IUS/UDO、 SpatialWare ODBC 驱动程序的连接字符串示例。

Oracle Spatial 连接字符串：

```
UID=george;PWD=password;SRVR=ontario
```

IUS/UDO 连接字符串：

```
DRIVER={INFORMIX 2.80 32
BIT};UID=informix;PWD=secret;DATABASE=sw;HOST=adak;SERVER=adak_tli;SERV
ICE=sqlexec;PROTOCOL=onsoctcp;
```

SpatialWare 连接字符串：

```
DRIVER={SpatialWare 32 Bit
Driver3.50};HOST=fire;UUID=oracle;UPWD=secret;UID=GEORGETOWN;PWD=GEORGE
TOWN;OSID=QASW1
```

SpatialWare for SQL Server 字符串：

```
DRIVER={SQL SERVER};SERVER=LAMONT;VID=Troll;
PWD=secret;Database=GEORGETOWN
```

MapX DBMS 图层查询

DBMS 图层的指定查询定义了 DBMS 中的数据的结果集，它是在要添加的图层中的数据。您可以建立一个相当复杂的查询，进行功能强大的服务器大小分析，它在 mapx 中定义了一个可以绘制地图的图层。 MapX 将在内部使用此查询来访问数据。

MapX 基于您的查询生成了若干个内部查询来访问基于选择 / 关键字的查询和绘图器中的数据。从中选择对象列的源表必须在服务器上的 MapInfo Mapcatalog 中注册。 MapX 要求执行此操作，才能获得有关几何结构列 （如用于绘制图层的坐标系、存储格式和默认样式）的某些元数据。

为了在添加新图层使查询成功执行，该查询必须包含几何结构列和关键字列。有时，为了对小型数据集建立更复杂的查询 （无论是空间检索，还是空间谓词都会导致查询失败），可以禁用谓词，指定 MBRSEARCH “OFF”，以便可以对结果进行地图绘制。

几何结构列

如果不指定 MapX 可以识别的几何结构列，则 AddLayer 将失败。 MapX 将通过在 Mapcatalog 中查阅表的几何结构列并检查该列的结果集数据类型来确定该列。您可以通过伪列名 “OBJECT”对几何结构列进行一般性引用，也可以引用使用特定列名的几何结构列。需要此表单以引用 X/Y 可绘制地图层的几何结构列。您可以通过任何服务器支持的几何结构函数 / 表达式来指定几何结构列。

示例

```
Select OBJECT from rdbsdata
Select sw_geometry from rdbsdata
select sw_member, ST_Buffer(geometry, 66.0, 0.1) from rdbsdata a
geometry function
Select st_geometry(st_point(72.5, 42.5.)) from rdbsdata a geometry
constructor

Spatialware function examples:
select sw_member, ST_Buffer(geometry, 66.0, 0.1) from rdbsdata
select ST_Overlap(flood100.sw_geometry, lake.sw_geometry) from flood100,
lake where ST_Overlaps(flood100.sw_geometry, lake.sw_geometry)

Spatialware constructor examples:
select 1 "prinx", st_geometry(st_point(72, 42)) from dual
```

```
Oracle sdo_buffer example:
Select prinx, mdsys.sdo_geom.sdo_buffer(geoloc, (select diminfo from
sdo_geom_metadata where table_name = 'ALINE'), 20) from aline where
prinx = 1

Oracle constructor example:
Select 1 "prinx",
mdsys.sdo_geometry(3,null,null,mdsys.sdo_elem_info_array(1,3,3),
mdsys.sdo_ordinate_array(-79.919909,40.553465,-71.060457,45.363657))
from dual
```

关键字列

关键字列必须在查询在返回，才能将它作为图层进行地图绘制。这是使 MapX 可以识别结果集中各行以对图层执行阴影、选择和标签操作的原因。

注：大多数情况下，不需在查询中指定关键字列。

MapX 将查询和确定可用于唯一引用结果集中行的最佳关键字列，然后将这些列添加到查询中 （如果它们不在查询中）。在大多数情况下，这将是主关键字 / 唯一索引。

注：对于 Oracle Spatial 图层，使用 ROWID。

对于某些查询，MapX 无法识别关键字。在对视图或同义词的查询中是这样。视图或同义词必须显示在 mapinfo mapcatalog 中。在大多数情况下，它们还必须根据底层的 Spatial 索引系统的要求进行注册。由于 MapX 无法确定这些对象的关键字，所以提供一个机制，允许应用程序开发者 / 查询写入识别结果集中的关键字列。关键字必须是单列，而且必须是结果集中的特殊值。要识别要用作关键字列的列，可以指定 prinx 或 mi_prinx 的列别名 （例如，选择 custid mi_prinx、 ccustname、 mycust 中的对象）。

示例

```
Select customer_id prinx, object, from customer_view
```

列别名 “prinx” 用于识别 customer_id 列并将其用作图层的关键字列。您还可以在 create view 语句中为所需关键字列定义别名，以自动识别该视图任何查询的关键字列。

示例

```
Create view customer_view as select customer_id prinx, geloc from
customer
```

一般情况下，如果在结果集中找到 prinx 或 mi_prinx 的列名列别名，则该列将用作图层的关键字列。这样即可使应用程序 / 查询编写者指定所需的关键字列。

访问属性数据

要使用所有可用的数据列，请指定一个查询，如 “Select * From tablename”。不要求指定 * （星号）；而可以明确指定您想使用的列。为了获得最佳性能，请限制您的查询，以便它只检索所需的列。

要使用 DBMS 图层中的空间对象访问查询中所选的属性数据，使用数据集类型为 midatasetlayer 的 datasets.add 方法 （从现有图层中获取属性）。

添加 DBMS 图层时，出于性能原因，您应该只在查询中指定您想在应用程序中使用的列。这些列是空间列 （即关键字列，如果未指定则会自动添加）以及您想用以标记或创建主题的列。对任何可以绘制地图的表，您可以使用伪列 “OBJECT”来引用包含空间数据的列。这是在使用含 X/Y 列的表上 MapMarker MDIGEOADDRESS 列的表所要求的。

注：您可以使用任何服务器端表达式 / 函数来指定列。在实际应用中不要使用 select * from tab。

以下代码示例将打开一个 MapInfo *.tab 文件，然后访问远程数据并将其与图层联系起来。您现在可以基于列标记或放置主题。

```
Dim Lay As Layer
Dim dsname As String
Set Lay = Map1.Layers.Add(filename)
dsname = Lay.Name
Map1.Datasets.Add miDataSetLayer, Lay, dsname
```

性能问题

与数据库服务器建立连接需要几秒钟。这是一次性开销，在首次打开表时发生。

一个 “打开”操作也需要几秒钟。每次打开新表时都会发生这种一次性开销。

地图显示速度取决于从服务器中检索的数据量。在某些情况下，从服务器上显示地图明显慢于从本地文件中显示地图。速度还取决于 MapX 是否已经缓存了正在显示的图元。

使用缓存

通过了解如何在 MapX 中使用缓存，可以提高您的应用程序的性能。下面几节说明了缓存是什么、它如何在 MapX 中使用以及 LayerInfo 对象的 CACHE 参数。

缓存是什么?

应用程序可以从远程数据库中访问 MapX 图元，而不是从本地文件中访问。需要对地图进行操作，而不是每次从数据库中读取这些记录。 MapX 可以在缓存中临时存储这些记录。这样就限制了应用程序和远程数据库之间的调用次数。服务器图层中的记录可以进行缓存，以提高应用程序的性能 （如绘图、主题、标记等）。服务器图层数据在 MapX “地图”窗口中读取并绘制时在内部进行缓存。所有后续重新绘制操作将从该缓存中读取，而不是进入服务器数据库中读取该数据。缓存可以大大提高重新绘制操作的性能。

开发者可以使用几个设置来定制如何使用缓存。通过在 LayerInfo 对象中指定 CACHE 参数添加服务器图层时可以启用 （或禁用）缓存。在默认情况下 “启用”缓存。

如何使用缓存

对于缓存的各个记录，每个属性数据值都存储在内存中，每个图元对象都存储在磁盘上的临时 Rtree 文件中。对于含许多记录和 / 或记录较大 （如每个属性数据记录的字节数）的表，缓存会使用很大量的

内存。如果应用程序试图缓存太多的数据，则可以要求使用虚拟内存，但会降低性能。对于如何使用缓存，应用程序可以选择使用。 MapX 提供了许多用来控制缓存的机制。

LayerInfo 对象和 CACHE 参数

将服务器图层添加到地图中时，可以通过在 LayerInfo 对象中指定已传递到 Layers.Add 的 CACHE 参数来启用缓存。这些参数有四个可能的值：ON、 OFF、 ALL、 USER。 ON 是默认值。

参数	说明
OFF	值为 “Off”，则说明图层决不会使用缓存。所有数据操作将直接进入数据库服务器进行。
ON	LayerInfo CACHE 的值为 “ON”，则说明 MapX 将维护缓存记录，以最大限度地提高标准 MapX 操作的性能。对缓存进行维护，至少包含图层的所有已显示记录，在当前 “地图”窗口的 MBR 中显示。一旦缓存初始 “地图”窗口 MBR，则完全在缓存初始范围内的平移和缩放操作将访问缓存的记录，无需查询数据库。如果平移 / 缩放操作在缓存区之外，则 MapX 将新的 “地图”窗口 MBR （视图）添加到缓存中，并从数据库服务器中获取缺少的记录，然后将其添加到缓存中。 最初不丢弃旧的地图视图；而是维护以前地图视图的内部历史记录。要避免缓存增长过大，可以在 MapX 上放置几个控件，以便确定何时丢弃旧的缓存视图 （“地图”窗口 MBR 区域）。这些控件是 LayerInfo 对象的参数，可以在初次添加图层时设置。通过它们，开发者可以设置以下限制：缓存使用的最大内存量或磁盘空间、可以在历史记录中维护的最大以前 “地图”窗口视图数、可以在缓存中维护的最大记录数和 / 或旧 “地图”窗口视图可以在缓存历史记录中保留的最长时间。这些显示可以单独使用，也可以合并使用，以便进行缓存管理，以最大限度地满足应用程序的需要。 有关特定参数名称、默认值和可用设置，请参阅第 373 页的 “LayerInfo 对象”。 **注释**：可以设置这些限制的默认值，以与 MapX 以前版本保持一致，即将在缓存历史记录中维护的最大以前地图视图数 （不包括当前地图视图）默认为零。

参数	说明
USER	LayerInfo CACHE 的值为 USER，则说明 MapX 将创建缓存，但只有应用程序开发者指定的记录才会放在缓存中。指定将哪些记录放在缓存中的可用机制是 BoundConstraint、 FeaturesConstraint 和 AllFeaturesConstraint 对象。“约束”一词指这些对象会约束缓存，使其纳入指定记录。 BoundsConstraint 对象 （参阅第 249 页的 “BoundsConstraint 对象”）可用于将所有记录放在图元 MBR 与约束 MBR 交叉的缓存中。 FeaturesConstraint 对象 （参阅第 285 页的 “FeaturesConstraint 对象”）可用于将特定记录添加到缓存中。例如，如果打算进行的分析包括多个步骤和 / 或图元的 Feature 或 RowValues 的读取 （可能是对一组从 Layer.Search、 Layer.SearchWithinDistance 等返回的图元），则最好在分析期间将这些记录放在本地缓存中，完成后再将其删除。FeaturesConstraint 提供了这种功能。如果一个应用程序将执行一个触及每个记录的操作，分析较多，则最好临时缓存整个图层数据集。可以使用 AllFeaturesConstraint 完成此操作 （参阅第 236 页的 “AllFeaturesConstraint 对象”）。将缓存设置为 ON 时，也可以使用这些约束对象。在这种情况下，它们可以将记录添加到缓存中，但对以前“地图”窗口视图的缓存历史记录没有任何影响。将缓存设置为 OFF 或 ALL 时可以使用这些约束对象，但没有任何效果。 **注释：**约束对象对非服务器图层没有任何影响。
ALL	ALL 指 MapX 将一次性缓存整个表，无需在绘制时逐次添加和删除记录。十分频繁访问的小表是这种选项的理想选择。

*** 提示：**可以通过使用 Layer.Refresh 方法从数据库中刷新整个缓存 [参阅第 357 页的 “Layer.Refresh 方法”]。该操作将刷新任何当前分配的约束对象和当前视图中的数据，将清除所有其它缓存历史记录。

注：如果试图缓存的数据或图层太多，则可以强行使用虚拟内存，但可能无法提高性能。

MapInfo 地图目录

如果 MapX 应用程序需要访问远程空间数据，则 DBMS 中必须有一个特殊的表，称作 MapInfo 地图目录。对于每个数据库都必须创建一个目录，然后才能在 MapX 中查看该数据库中作为图层的任何表。地图目录必须包含您想从数据库中访问的每个可以绘制地图的表中有关空间列的信息。

您的 MapX 应用程序可以使用服务器上已经存在的地图目录。（该目录由不同的 MapInfo 客户应用程序共享，包括 MapInfo Professional。）如果服务器上没有地图目录，则需要创建一个地图目录。

注：MapX 支持在远程数据库中存储单个图元的样式信息。因此，图层是从地图目录中获取自己的样式的。

使用 MapInfo Professional 管理地图目录

如果您使用 MapInfo Professional，则可以使用 MapInfo Professional 创建并管理地图目录。 MapInfo Professional 包括 MapBasic 实用程序，如 MIODBCAT.MBX，有助于创建和管理地图目录。

通过 MapInfo Professional 创建的地图目录可以由 MapX 应用程序使用。例如，在公司环境中，一个用户 （可能是数据库管理员）可以使用 MapInfo Professional 创建地图目录，然后众多最终用户都可以运行 MapX 应用程序从该地图目录中读取信息。

将空间数据加载到 DBMS

如果您的空间数据采用 MapInfo 表的形式，则可以将该数据导入 DBMS 数据库中。

- 要将数据加载到 SpatialWare for Oracle 中，请使用随 MapInfo Professional 提供的 MISWUP32.MBX 实用程序 （位于 SpatialWare 客户程序安装目录中）。
- 要将数据加载到 SQL Server SpatialWare、 DB2 SpatialWare、 IUS/UDO SpatialWare 和 Oracle Spatial 中，请使用 MapInfo EasyLoader。它随 MapInfo Professional 一同分发，也可以从 www.mapinfo.com 下载。（对于 DB2 和 IUS/UDO，必须在服务器上安装 SpatialWare Extender/ Cartridge）
- 要将非可绘制地图的表或仅由点构成的表加载到任何 DBMS 中，请使用工具目录中随 MapInfo Professional 提供的 MIUPLOAD.MBX 实用程序。这样即可将 Point 对象坐标存储在不同的 x/y 数字列中。您可以通过在 MapCatalog 中将这些表注册为 “X/Y Schema”表，从而对其进行地图绘制。

手动创建 MapInfo 地图目录

如果您不使用 MapInfo Professional，则需要手动创建地图目录 （或者让数据库管理员创建地图目录），如下所述。对于每个服务器 / 数据库，您只需创建地图目录一次。

1. 在可绘制地图的表所在的特定数据库中创建 MAPINFO 用户。
2. 在数据库中查 MAPINFO_MAPCATALOG 表。 Create Table 语句需要与以下 SQL Create Table 语句等效：

```
Create Table MAPINFO_MAPCATALOG (
SPATIALTYPE Float,
TABLENAME Char(32),
OWNERNAME Char(32),
SPATIALCOLUMN Char(32),
DB_X_LL Float,
DB_Y_LL Float,
DB_X_UR Float,
DB_Y_UR Float,
COORDINATESYSTEM Char(254),
SYMBOL Char(254),
XCOLUMNNAME Char(32),
YCOLUMNNAME Char(32),
RENDITIONTYPE INTEGER,
RENDITIONCOLUMN CHAR(32),
RENDITIONTABLE CHAR(32)
)
```

注：表的结构必须与该语句完全一样。可以进行的唯一替换是支持 varchar 或文本数据类型的数据

库；这些数据类型可以替换为 Char 数据类型。

3. 对 TABLENAME 和 OWNERNAME 创建唯一索引。这样，每个所有者只有一个表可以进行地图绘制。

```
create unique index mapcat_i1
on mapinfo.mapinfo_catalog (OwnerName,TableName)
```

4. 对 MAPINFO_MAPCATALOG 授予 Select、 Update 和 Insert 特权。这样，用户即可使表成为可以进行地图绘制的表。 Delete 特权应为数据库管理员保留专用。

```
grant select, insert, update on mapinfo.mapinfo_mapcatalog to
public
```

每个记录的样式

通过对每个记录的样式的支持，可以在空间数据库实现中增加一个图元，它长期以来一直在 MapInfo TAB 文件中。具体地说，它可以使单个图层中的每个几何结构都有其自己的样式。例如， Oracle8i Spatial 的单个 “政府机构”图层可以有学校、市政厅、图书馆和警察局，每个点类型都使用自己的符号表示 （即所有学校都使用一个学校符号表示）。与此类似， SpatialWare SQL Server 中的单个 “公路”图层可以有不同类型的公路，大街以单像素黑线表示，二级公路以双像素红线表示，州际公路以双红线表示。

列	数据类型	说明
RENDITIONTYPE	INTEGER	表中每个记录的表示类型的数字指示器。有效值有： • **0:** 所示表中没有生效的记录样式。将使用表的默认样式照常读取 / 更新对象。 • **1:** 所示表使用每个记录的样式。表中包含样式的列在 RENDITIONCOLUMN 中记录，它包含一个 MapBasic 字符串 （例如地图目录 SYMBOL 列中当前使用的同一格式）。
RENDITIONCOLUMN	VARCHAR(32)	该列包含所示表中有对象样式值的列名 （表示类型为 1）。
RENDITIONTABLE	VARCHAR(32)	但是，目前未使用该列，必须使用它，然后才能让 MapX 正确地检测到此新图元是否存在。计划将来推出一个增强功能，使样式在独立的表中进行管理，使列中的值可以由 RenditionColumn 标识。另外，包含一个可用来查询此属性识别的表中的样式的关键字。此功能尚未实现。

注：如果这些列不存在，则对象样式将继续照常使用 （如表的默认样式将适用于所有对象），一个

例外情况将在下面进行讨论。如果这些列存在于地图目录中，则会使用它们，如下所示：

已知局限 / 要求

使用明确请求样式列的图层时，可能在更新现有图元 （或其它行数据值）和插入新记录时会遇到困难。原因是：MapX 根据需要在 INSERT/UPDATE 语句中自动添加样式列，对于没有隐藏样式列的图层而言会生成两次指定列名的 SQL 语句。

使 DBMS 表成为可以进行地图绘制的表

对于您想从 MapX 访问的每个空间表，需要在 MAPINFO_MAPCATALOG 表中添加一行。如果您不使用 MapInfo Professional 管理地图目录，则必须在 MAPINFO_MAPCATALOG 表中手动添加行。下表说明了各列的语法和含义：

列名	赋值	示例
SPATIALTYPE	**注释**：该列说明了如何存储和检索数据的空间对象格式以及列中支持和不支持的空间对象类型。小数点左边的数字是**空间对象格式**。右边的数字是可以在列中存储的**空间对象类型**。 **MapInfo 空间对象格式** **1.x**：用 micode （一个串行化的 quadtree 关键字）检索的 X/Y 列中的点图层 **2.x**：Oracle MD/SDO 1 版 HHCODE_ - **不支持** **3.x**：Oracle MD/SDO 1 版 HHCODE_PARTIONED - **不支持** **4.x**：X/Y 列中的点图层 **5.x**： SpatialWare for Oracle **6.x**：Ingres SOL - **不支持** **7.x**： Sybase SQS - **不支持** **8.x**：Oracle SDO 2 版 - **不支持** **9.x**： MapInfo Geocoding DataBlade SpatialWare Point Module **10.x**： MapInfo Geocoding DataBlade XY Module **11.x**： SpatialWare IDS/UDO datablade **12.x**： SpatialWare Extender for DB2 **13.x**：Oracle Spatial **14.x**：**SpatialWare for Microsoft SQL Server** **空间对象类型** **x.0**：仅点 **x.1**：仅直线 **x.2**：仅区域 **x.3**：支持所有类型	5.3
TABLENAME	表的名称。	DRAINAGE
OWNERNAME	表的所有者姓名。	GEORGETOWN

列名	赋值	示例
SPATIALCOLUMN	包含空间图元 （如有）的列的名称： SW_GEOMETRY （使用 SpatialWare Type/IDS/UDO 可以绘制地图） NO_COLUMN （使用 X 可以绘制地图） MI_SQL_MICODE （使用 MI Code 可以绘制地图） 或者数据类型为 ST_SPATIAL 的 IDS/UDO、DB2 或 Oracle 列的名称。 Oracle 8i SDO_GEOMETRY 列的名称。	SW_GEOMETRY
DB_X_LL	图层边界矩形左下角的 X 坐标，采用 COORDINATESYSTEM 所示的单位 （见下）。	ñ360
DB_Y_LL	左下边界的 Y 值。	ñ90
DB_X_UR	右上边界的 X 值。	360
DB_Y_UR	右上边界的 Y 值。	90
COORDINATESYSTEM	是 MapInfo CoordSys 子句的字符串 （但最开头没有关键字 “CoordSys”），它说明地图投影、坐标单位等。对于简单的 Lon/Lat 地图，说明 “Earth Projection 1, 0”。	Earth Projection 1, 0
SYMBOL	MapInfo Symbol 子句 （如果图层只包含点）；或者 Symbol 子句后跟 Pen 子句 （表示线性图元的样式），再后跟另一 Pen 子句 （表示区域边界的样式），再后跟 Brush 子句。	Symbol(35,0,12) Pen(1,2,0) Pen(1,2,0) Brush(2,255,255)
XCOLUMNNAME	对于 X/Y 可绘制地图的表，指定包含 X 坐标的列名。如果没有该列 （即如果该表使用单一空间列，而不是 X 列对），则指定 NO_COLUMN 或置空。	NO_COLUMN
YCOLUMNNAME	对于 X/Y 可绘制地图的表，指定包含 Y 坐标的列名或者指定 NO_COLUMN。	NO_COLUMN
RENDITIONTYPE	它说明如何应用对象样式信息。 • “**0**” - 说明所有对象将在 Map.Catalog 的符号字段中指定的样式应用到这些对象。 • “**1**” - 说明表中有一个单列，包含表中每个对象的样式信息的字符串表示 （例如，每个对象都可以有自己的样式）。	0 或 1

列名	赋值	示例
RENDITIONCOLUMN	如果 RENDITIONTYPE 是 “1”，则该字段会在空间表中存储包含样式信息的列的名称 （由 TABLENAME 标识）。该列将自动添加到对表的任何查询中并随对象的更新而得到维护 （更新）。用户不应该使用 intersect 或 update 语句在查询中指定此列，因为可能出现问题。在 select 子句中包含此列的查询 （不包括通配符 “*”）可以通过 Dataset 对象访问这些值。样式列中含 NULL 值的行将 MapCatalog 的 SYMBOL 字段中的样式应用到对象中。	MI_SYMBOLOGY
RENDITIONTABLE	当前未使用，但留待将来使用，必须存在此列，才能让 MapX 正确标识和应用记录一级的样式。	NULL

Symbol、 Pen、 Brush 子句语法

如果手动创建 MAPINFO_MAPCATALOG 表，对远程空间数据库提供支持，则需要指定符号样式以及可能的直线和填充样式。

指定点样式

使用 Symbol 子句指定点样式。有三类 Symbol 子句：用于指定 MapInfo 3.0 样式符号的子句；用于指定 TrueType 字体符号的子句；用于指定位图符号的子句。

Symbol 语法	示例
Symbol(形状 , 颜色 , 大小) 或 Symbol(形状 , 颜色 , 大小 , 字体 , 字型 , 旋转) 或 Symbol(位图名 , 颜色 , 大小 , 定制样式)	Symbol(35,0,12) Symbol(64,255,12,"MapInfo Weather" ,17,0) Symbol("sign.bmp", 255, 18, 0)

指定直线样式

使用 Pen 子句指定直线样式。在地图目录中，您需要指定两个 pen 子句：一个是指定线性图元的显示，另一个是指定区域边界的显示。

Pen 语法	示例
Pen(厚度 , 图案 , 颜色)	Pen(1, 2, 0)

指定填充样式

使用 Brush 子句指定已关闭图元 （区域）的样式。

Brush 语法	示例
Brush(图案 , 颜色 , 背景色)	Brush(2, 255, 65535)

故障排除

如果在使用 MapX SpatialWare 应用程序时遇到问题，请使用下表来帮助分析和解决问题。

问题说明	可能原因	解决办法
MapX “图层不可匹配”。	试图对 SpatialWare 图层进行数据绑定。	SpatialWare 图层当前不支持数据绑定。
使用指定的索引未找到任何对象。	建立了对不存在的表的查询。 空间查询结果中不包含空间对象。 建立了对非空间表的查询。	检查表名是否正确和适当。另外，该表需要是可以绘制地图的表。 使用 EasyLoader Upload 实用程序来使表成为可以进行地图绘制的表。 检查查询，查找可能的语法错误。另外，请确保查询结果包括在 MapInfo_MapCatalog 的空间列中指定的字段。
MapX datasets.rowcount 的值为零。	数据集是从 DBMS 服务器上创建的。	对于从 DBMS 服务器上创建的数据集， MapX datasets.rowcount 的值将始终为零。使用图层的 AllFeatures.Count 属性
地图的缩放级别似乎不正确。例如，地图可能缩小得太小，无法识别任何地理位置。	DBMS 图层的 MBR 是由 MapInfo_MapCatalog 表决定的。 MapCatalog 的表区域导致不同于您所希望的输出缩放级别。	使用 MapInfo Professional MDX 工具 MISETMBR.MDX 编辑 MapInfo_MapCatalog 中的区域 (DB_X_LL, DB_X_UR, DB_Y_LL, DB_Y_UR)。

使用坐标系统

在本章中，您将了解如何使用坐标系统 （有时候称作“地图投影”）来更改地图的外观，或者了解如何使用坐标系统来更改 MapX 处理地图坐标所用的单位。

坐标系统的基本概念

每一地图都具有一个坐标系统，该坐标系统以各种方式影响地图绘制软件：

坐标系统影响 X-Y 坐标的处理方式。例如，一个地点可以通过经度 / 纬度的度数来表示，或者可以通过其它单位 （例如米）来表示。

坐标系统影响地图的外观。更改地图的坐标系统可能会使地图显得伸展或扭曲。坐标系统代表计算机地图绘制的复杂、高级的方面。本章论述的内容假设您已熟悉了地图绘制概念，例如假东向和地图投影。

获取坐标系统对象

MapX 对象模型通过 CoordSys 对象公开坐标系统信息。通过此对象的属性，您可以了解与一个坐标系统有关的各方面的情况。

有若干方法来获取 CoordSys 对象。您所使用的技术取决于您尝试完成的内容。

Map.DisplayCoordSys 属性返回读 - 写 CoordSys 对象。您可以使用此对象来控制用来显示该地图的坐标系统和地图投影。有关详细信息，请参见本章后面第 178 页的 “在不同 CoordSys 中显示地图”。

Map.NumericCoordSys 属性也返回读 - 写 CoordSys 对象。使用此对象来设置 MapX 用来处理 X-Y 坐标的坐标系统。有关详细信息，请参见第 179 页的 “在不同 CoordSys 中指定 X-Y 坐标”。

Layer.CoordSys 属性返回只读 CoordSys 对象，该对象指示在特定地图图层中用来存储图元的坐标系统。如果您设置 Map.DisplayCoordSys 来匹配 Layer.CoordSys，则可以最大限度地加快该图层的显示速度。（当 DisplayCoordSys 不同于用来保存图层的 CoordSys 时， MapX 会即时转换图元，这将减慢显示速度。）

查询 CoordSys 对象的属性

对于每一类型的坐标系统，只有某些属性是适用的。例如，如果 CoordSys.Type 属性是 miRobinson (12)，则只有 Datum、Units 和 OriginLongitude 属性适用。Robinson 地图投影完全不使用任何其它属性。

若要了解某一特定投影是否使用了某一属性，请参见第 181 页的 “概要介绍坐标系统所使用的参数”。

您可以查询任何 CoordSys 对象的所有属性，甚至可以查询不适用于该坐标系统类型的属性。如果某一属性不适合于给定的 CoordSys 类型，则 MapX 将返回默认值。如果您需要坚持使用某一 CoordSys 对象，则可以通过查询并保存其所有属性来这样做，而不必担心每一属性是否适用。

在不同 CoordSys 中显示地图

一个应用程序可有两种方法来指定用其显示地图的坐标系统 （或 “投影”）：

- 使用 Map.DisplayCoordSys.Set 方法来设置新的显示 CoordSys。

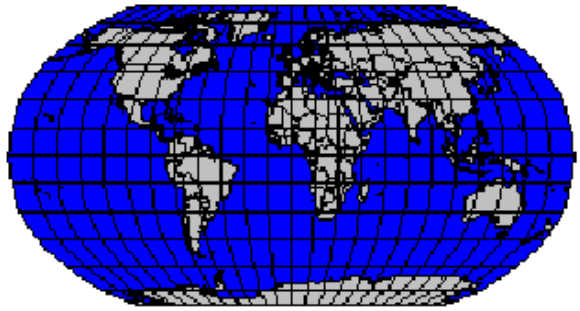
  - 或者 -

- 将新的 CoordSys 对象赋予 Map.DisplayCoordSys 属性。

注：如果一个地图包含一个或多个光栅图图层，则 MapX 自动在最可见的光栅图指定的投影中显示该地图。如果具有不同投影的其它光栅图成为最可见的光栅图，则坐标系统可以随着地图视图的更改 （由于缩放或平移）而进行更改。在此情况下，您无法更改地图的显示坐标系统。

显示 Robinson 地图

以下所示的 Visual Basic 示例更改地图的 DisplayCoordSys，以便通过 Robinson 地图投影来显示该地图：



```
Dim iProjectionType As
Integer
Dim iDatumNumber As Integer
Dim iUnits As Integer
Dim dOriginLongitude As
Double
    Initialize the variables
to be used by coordsys.set.
  ' Values were obtained from
the "Robinson" entry
  ' in the file MAPINFOW.PRJ, which looks like this:
  ' "Robinson", 12, 62, 7, 0
  iProjectionType = miRobinson  '(value: 12)
  iDatumNumber = 62             'North American 1927 (NAD 27)
  iUnits = miUnitMeter          '(value: 7)
  dOriginLongitude = 0
  Map1.DisplayCoordSys.Set iProjectionType, iDatumNumber, _
  iUnits, dOriginLongitude
```

在不同 CoordSys 中指定 X-Y 坐标

Map 对象具有 NumericCoordSys 属性，表示用来处理数字坐标的坐标系统。该属性同时影响 X-Y 坐标的输入和输出，如下所示：

- 在指定地图坐标（指定为对象属性或方法参数）时， MapX 假定这些坐标以此坐标系统为准。
- 当 MapX 返回地图坐标 （作为对象属性或事件）时，这些坐标以此坐标系统为准。

默认情况下，该坐标系统是经度 / 纬度 WGS-84。换言之，如果您没有修改 NumericCoordSys，则 MapX 假定地图坐标以此经度 / 纬度度数为准。

应用程序可以使用两种方法指定用来处理数字坐标的坐标系统：

- 使用 Map.NumericCoordSys.Set 方法来设置新的数字 CoordSys。

  - 或者 -

- 将新的 CoordSys 对象赋予 Map.NumericCoordSys 属性。

在不同 CoordSys 中检索坐标

以下 Visual Basic 示例更改地图的 NumericCoordSys，以便以不同的坐标系统返回坐标。

```
Dim iProjectionType As Integer
Dim iDatumNumber As Integer
Dim iUnits As Integer
Dim dOriginLongitude As Double

' Display map's center X-Y in original coordsys
Debug.Print "Original Center: " & Map1.CenterX & ", " & _
Map1.CenterY

' Initialize the variables to be used by coordsys.set.
' Values were obtained from the "Robinson" entry
' in the file MAPINFOW.PRJ, which looks like this:
' "Robinson", 12, 62, 7, 0

iProjectionType = miRobinson  '(value: 12)
iDatumNumber = 62             'North American 1927 (NAD 27)
iUnits = miUnitMeter          '(value: 7)
dOriginLongitude = 0

Map1.NumericCoordSys.Set iProjectionType, iDatumNumber, _
iUnits, dOriginLongitude

' Display map's center X-Y in the new coordsys
Debug.Print "New Center: " & Map1.CenterX & ", " & Map1.CenterY
```

显示 “选择投影”对话框

若要显示让用户选择坐标系统的对话框，请使用 CoordSys.PickCoordSys 方法。

该对话框自动初始化，这样突出显示的说明将描述 CoordSys 对象。如果用户在对话框中单击 “确定”，则该方法返回 True，并且更新 CoordSys 对象以匹配用户所选的坐标系统。

要在设计阶段显示该对话框：

1. 右击地图控件。
2. 从快捷菜单中选择 “属性”。
3. 在 “常规”选项卡上，单击 “投影”按钮。

MapX 将显示 PickCoordSys 方法所调用的相同的 “选择投影”对话框。

示例 (PickCoordSys)

以下语句在运行时显示 “选择投影”对话框，使用户可以更改显示该地图所用的坐标系统。

```
Map1.DisplayCoordSys.PickCoordSys
```

概要介绍坐标系统所使用的参数

CoordSys 对象支持若干参数 （数据、单位、原始经度等）。但是，每一投影只使用某些参数。下表显示参数所应用于的投影。

投影	数据	Units	Origin Long.	Origin Lat.	Std. Par. 1	Std. Par. 2	Azim.	Scale Fact.	False East.	False Nor.	Range
亚尔勃斯等面积圆锥投影	X	X	X	X	X	X			X	X	
等距方位投影	X	X	X	X *							X

投影	数据	Units	Origin Long.	Origin Lat.	Std. Par. 1	Std. Par. 2	Azim.	Scale Fact.	False East.	False Nor.	Range
圆柱等面积投影	X	X	X		X						
双立体照相投影	X	X	X	X				X	X	X	
Eckert IV 投影	X	X	X								
Eckert VI 投影	X	X	X								
等距圆锥投影	X	X	X	X	X	X			X	X	
Gall 投影	X	X	X								
Hotine 斜轴墨卡托投影	X	X	X	X			X	X	X	X	
朗伯方位等面积投影	X	X	X	X *							X
朗伯正形投影	X	X	X	X	X	X			X	X	
经度 – 纬度	X										
墨卡托投影	X	X	X								
米勒投影	X	X	X								
摩尔魏特投影	X	X	X								
新西兰地图栅格投影	X	X	X	X					X	X	
鲁宾逊投影	X	X	X								
正弦曲线投影	X	X	X								
立体照相投影	X	X	X	X				X	X	X	
瑞士墨卡托投影	X	X	X	X					X	X	
横轴墨卡托投影	X	X	X	X				X	X	X	
区域墨卡托投影	X	X	X		X						
多圆锥投影	X	X	X	X					X	X	
Cassini - Soldner 投影	X	X	X	X					X	X	

* MapX 只在极地方面支持等距方位角投影和朗伯方位角等面积投影。这些投影的纬度原点必须是 90 或 -90。

使用来自 MAPINFOW.PRJ 的设置

如果您使用 MapInfo Professional，您将会在 MapInfo 可执行文件的相同目录中找到称作 MAPINFOW.PRJ 的文件。 MAPINFOW.PRJ 是一个文本文件，包含定义 MapInfo 的坐标系统的参数。 MAPINFOW.PRJ 的副本随 MapX 一起提供。

假定您已在 MapInfo 中使用了特定的投影 （例如鲁宾逊投影），并且您想要在 MapX 应用程序中指定该投影。 MAPINFOW.PRJ 文件向您提供填写用于 MapX 方法的各种不同参数 （例如 CoordSys.Set）所需的信息。

MAPINFOW.PRJ 中的各项具有以下参数。参数是用逗号分隔的。

部分	说明
说明	描述坐标系统的字符串。此说明出现在对话框中。
类型	通常准确匹配某一 CoordSysTypeConstants 的数字。用户可以以两种方式定制此数字：若要指定仿射转换，用户将此数字加上 1000 ；若要指定边界，用户将此数字加上 2000。
数据	表示 MapX 支持的某一数据的数字，或者指示定制数据的特殊数字 （999 或 9999）。
单位 [4 个定制数据参数的集合]	可选；只有在 “数据” 数字是 999 时才适用。
[9 个定制数据参数的集合]	可选；只有在 “数据” 数字是 9999 时才适用。
投影参数集	从 0 到 8 个参数，取决于指定的类型。请参阅 **“概要介绍坐标系统所使用的参数”**，了解此类型使用哪些参数。
[7 个仿射转换参数的集合]	可选；只在 “类型” 数字恰好为 1000 或 3000 （大于 CoordSysTypeConstant）时才适用。
[4 个边界参数的集合]	可选；只在 “类型” 数字恰好为 2000 或 3000 （大于 CoordSysTypeConstant）时才适用。

MAPINFO.PRJ 示例

在文本编辑器中打开 MAPINFOW.PRJ，或者在字处理程序包中打开它。

注：MapX 在解释和设置地图坐标系统时使用此文件。在进行任何更改前应备份此文件。

在 MAPINFOW.PRJ 中，搜索 MapInfo 投影的名称，例如 “Robinson”。您找到的行包含该鲁宾逊投影的定义：

```
"Robinson", 12, 62, 7, 0
```

该行说明与鲁宾逊坐标系统有关的若干情况：

- 该行起始处的字符串是说明。该说明后的第一个数字 （在此示例中是 12）指示适用的坐标系统的类型。将该数字 12 与 MapX CoordSysType 常数表相核对，您将看到类型 12 与 MapX 常

数 miRobinson 相匹配。（在此情况中，您可以通过该说明清楚地了解坐标系统使用鲁宾逊投影；但在其它一些情况下，您可能不会清楚地了解所使用的投影。）

- 请参阅 **“概要介绍坐标系统所使用的参数”**，了解鲁宾逊地图使用哪些参数。鲁宾逊地图仅使用三个参数：数据、单位和经度原点。现在，您知道如何解释该行中的其余数字：62 是数据数， 7 是单位数， 0 是经度原点。从 MapX 支持的数据列表中，您可以确定数据 62 表示用于美国大陆的 NAD 27。
- 从 MapUnit 常数表中，您可以确定单位类型数字 7 匹配 MapX 常数 miUnitMeter 的值。

您可以在调用 CoordSys.Set 方法时将此组值 (12, 62, 7, 0) 用作参数。例如：

```
objCoordSys.Set 12, 62, 7, 0
```

如何解释特殊数据数字

MapInfo 用户可以通过创建定制的数据定义定制 MAPINFOW.PRJ 文件。有两种类型的定制数据定义：

- 完整的定制数据定义由数据数字 9999 组成，后面紧随 9 个用逗号分隔的其它定制数据参数。
- 简化的定制数据定义由数据数字 999 组成，后面紧随 4 个用逗号分隔的其它定制数据参数。

如果在 MAPINFOW.PRJ 的某一行中指定数据数字 999 或 9999，则定制数据参数在其余坐标系统参数（单位、经度原点等）之前出现。有关定制数据语法的更多信息，请参见本章后面的 “定义定制数据” 一节。

如何解释特殊 CoordSys 类型数字

在 MAPINFOW.PRJ 中出现的坐标系统类型数字 （例如在上例中为 12）可能不与任何 MapX CoordSysType 常数值相匹配。您可能需要从该数字中减去 1000、 2000 或 3000，以获得表示有效 MapX CoordSysType 的数字。

如果 MAPINFOW.PRJ 中的某行包含仿射转换，则坐标系统类型数字将在对其加上 1000，并且在该行末尾添加 7 个仿射转换参数。

如果该行包括边界定义，则投影类型数字将在其上加上 2000，并且在该行末尾添加 4 个边界参数。

例如，来自假定的 MAPINFOW.PRJ 文件的以下行表示同时具有仿射转换和边界定义的坐标系统。此坐标系统使用 UTM Zone 10 （坐标系统类型：miTransverseMercator，其值为 8）。但是，因为此坐标系统同时包括仿射转换和边界说明，所以说明后的数字为 3008 (8 + 1000 + 2000)。

```
"DCS", 3008, 74, 7, -123, 0, 0.9996, 500000, 0, 3, 1.57, -0.21, _
84120.5, 0.19, 2.81, -20318.0, 70000, 0, 80000, 50000
```

尽管数字 3008 在 MAPINFOW.PRJ 内有效，但您必须在 MapX 中使用未改变的数字 8。

上面的 “DCS” 示例可以按如下所示加以解释：

```
name,
CoordSysType + 3000,
the projection parameters (Transverse Mercator uses 7 _
parameters),
the 7 affine transformation parameters,
The 4 bounds parameters (in the order: xmin, ymin, xmax, ymax)
```

应用仿射转换

如果您需要定义旋转后或扭曲后的坐标系统，则需要指定仿射转换。在 MapX 中，通过创建 AffineTransform 对象，然后将该对象传递给 CoordSys.Set 方法，可以定义此类转换。

注：如果您只是想要旋转地图，则不需要指定仿射转换。而需设置 Map 对象的 Rotation 属性。

仿射转换具有以下形式：

$x' = Ax + By + C$

$y' = Dx + Ey + F$

在上述等式中，对基坐标 (x, y) 进行转换以得出派生的坐标 (x', y')。从 A 到 F 的 6 个常数确定该转换的结果，如下所示：

A	沿 X 轴进行缩放或伸展。
B	沿 X 轴进行旋转或扭曲。
C	沿 X 轴进行移位。
D	沿 Y 轴进行旋转或扭曲。
E	沿 Y 轴进行缩放或伸展。
F	沿 Y 轴进行移位。

定义定制数据

大多数坐标系统使用 MapX 支持的标准数据之一。如果您需要使用未在该列表中列出的数据，并且您知道该数据的数学参数，则可以使用定制的数据定义坐标系统。

此论述提供可同时用于 MapX 和 MapInfo Professional 的示例，以供使用这两种产品的那些用户。如果您已完成了在 MapInfo 中创建定制数据的过程，则应发现令您的数据适合于 MapX 是很容易的。如果您没有使用 MapInfo Professional，则可以完全忽略特定于 MapInfo 的示例。

什么是数据?

数据是地球的形状和方位的数学描述。由于地球的形状是不规则的，因此，有许多不同的本地数据用于世界的不同部分。这些本地数据提供了在特定区域中地球表面的近似值。

每一地球坐标系统均使用特定的数据来近似地表示地球的表面。如果两个坐标系统使用不同的数据，则地图绘制软件必须在其将坐标从一个坐标系统转换为另一个坐标系统时执行数据转换。 MapX 和 MapInfo 使用 Bursa-Wolfe 数据转换方法，这一方法通常会精确到 10 米内。（当 MapInfo 在使用相同数据的两个坐标系统之间进行转换时，并不执行任何数据转换，并且结果通常精确到 0.1 米内。）

定义数据的参数

MapX 和 MapInfo Professional 使用以下信息定义数据：

- 椭圆体，也称作球状体。椭圆体是绕其短轴旋转以构成三维表面的椭圆。椭圆体是由两个数学参数描述的：以米为单位的其半长轴的长度 （用字母 a 表示）或其扁度 （用字母 f 表示）。支持超过 40 种预定义的椭圆体；请参见 “MapX 支持的椭圆体”。
- 以米为单位指定椭圆体沿其各个轴移位的距离的三个移位参数。这些参数通常用 dX、 dY 和 dZ 表示。您还可以用 u、 v 和 w 表示它们。 MapX Datum.Set 方法将这些参数标识为 ShiftX、 ShiftY 和 ShiftZ。
- 以弧度 - 秒为单位指定椭圆体沿其各个轴旋转的角度的三个旋转参数。这些参数通常用 EX、 EY 和 EZ 表示。 Datum.Set 方法将这些参数标识为 RotateX、 RotateY 和 RotateZ。
- 以百万分之几为单位指定用来调整椭圆体的大小量的比例修正因子。此参数用字母 m 表示。
- 以东格林尼治度数为单位的本初子午线的经度。本初子午线指定在地球上将哪一位置指派给经度 0°。大多数数据使用格林尼治作为本初子午线，因此该参数通常为 0。但是，某些数据使用不同的地点作为本初子午线。例如， NTF 数据使用巴黎作为其本初子午线，它是东格林尼治 2.33722917 度。如果您在坐标系统中使用 NTF 数据，则该坐标系统中的所有经度都是相对于巴黎的，而不是相对于格林尼治的。

在 MapInfo Professional 中定义数据

在 MapInfo 中，您可以按下面的顺序使用数据数字 9999，后随数据参数，来在坐标系统中定义定制数据：

```
9999, EllipsoidNumber, dX, dY, dZ, EX, EY, EZ, m, PrimeMeridian
```

某些数据只指定椭圆体和移位参数 (dX, dY, dZ)，而不指定旋转参数、比例修正或本初子午线。在这些情况中，您可以使用数据数字 999 来代替 9999，以简化定义：

```
999, EllipsoidNumber, dX, dY, dZ
```

这些参数集然后被复制到文件 MAPINFOW.PRJ 中；请参见下面的示例。

在 MapX 中定义数据

在 MapX 中，您可以通过创建 Datum 对象并使用其 Set 方法来设置参数，定义定制数据。然后，您将此 Datum 对象用作 CoorsSys.Set 方法的参数之一。

常见错误以及如何避免这些错误

与 WGS 84 数据相比，各种移位和旋转参数描述椭圆体在空间中的方位。确保这些参数具有正确的符号（正或负）是十分重要的。通常，描述本地数据的文档将列出将坐标从本地数据转换为 WGS 84 所需的参数。（这与通过从 WGS 84 减去本地数据派生的参数相同。）在该情况下，您可以完全像它们在文档中显示那样使用这些参数。但是，如果您具有列出用于以相反方向 （从 WGS 84 到本地系统）转换坐标的参数的文档，则必须颠倒这些移位、旋转和比例修正参数的正负符号。

以正确顺序列出这些参数也是非常重要的。某些文档以 EZ 为先列出这些旋转参数，例如：EZ, EY, EX。在这些情况中，您必须在定义定制数据时颠倒旋转参数的顺序。在您的文档使用希腊字母来表示参数时非常容易忽略上述工作。

数据转换

在将坐标从一种数据转换为另一种数据时， MapX 使用 Molodensky （3 参数）和 Bursa-Wolfe （7 参数）方法。它们都是可以将坐标从任何数据转换为其它任何数据的常用方法。

在推出了 NAD 83 数据后， NOAA 开发了称作 NADCON （表示 North American Datum CONversion，北美数据转换）的程序。这是仅用于将坐标从 NAD 27 转换为 NAD 83 （反之亦然）的专用性极强的程序。为执行此专门的任务，它比 Molodensky 常用方法精确得多； NADCON 可以精确到 0.1 米，而 Molodensky 只精确到 10-30 米。大多数美国政府机构 （包括人口调查局）都将 NADCON 作为在 NAD 27 和 NAD 83 之间进行转换的标准。

从 MapX 3.5 开始，如果这些坐标位于 NADCON 所涵盖的区域 （美国、波多黎各和维尔京群岛）内，就使用 NADCON 算法来在 NAD 27 和 NAD 83 之间转换坐标。如果坐标不位于这些区域内，或者它们使用非 NAD 27 或 NAD 83 的其它数据，则 MapX 使用 Molodensky 或 Bursa-Wolfe 转换方法。

由于需要进行文件访问， NADCON 方法可能稍慢于 Molodensky 方法。如果您要禁用该 NADCON 方法，则在注册表中添加 “NADCON”项。该注册表项应具有以下路径：

- HKEY_LOCAL_MACHINE\Software\\MapInfo\\MapX\\3.0\NadCon

如果将该项设置为 0，将使用 Molodensky 转换方法来代替 NADCON。若要启用 NADCON，则将该项设置为 1 （默认值）。 MapX 在其装载时读取此项，并且所有地图均使用相同的设置。您不能为特定地图启用或禁用 NADCON。

更多信息 ...

有关坐标系统和地图投影的更多信息，请查看以下资源：

相关的小册子

- 美国制图协会。选择 “世界地图 – 属性、扭曲、类别、方面”。弗吉尼亚州 Falls Church 市：美国测量和地图绘制大会。特殊出版号 2. 1988。
- 美国制图协会。根据需要匹配地图投影。弗吉尼亚州 Falls Church 市：美国测量和地图绘制大会。特殊出版号 3. 1991。
- 美国制图协会。哪一地图是最好的？世界地图投影。弗吉尼亚州 Falls Church 市：美国测量和地图绘制大会。特殊出版号 1. 1986。

若要获取这些小册子，请与下面的地址联系：

美国测量和地图绘制大会
5410 Grosvenor Lane, Suite 100
Bethesda, MD 20814-2212
301-493-0200

相关手册

- John P. Snyder。地图投影 － 工作手册。华盛顿：美国地理调查专业文献 1395. 1987.
- John P. Snyder 和 Philip M. Voxland。地图投影册。华盛顿：美国地理调查专业文献 1453. 1989.

若要获取这些手册，请与下面的地址联系：

Earth Science Information Center
U.S. Geological Survey
507 National Center
Reston, VA 20192
1.800.ASK.USGS - 或 - 1.703.648.6045

14 章

- 理解示例应用程序
- 从早期的 MapX 版本升级 C++ 应用程序
- 访问 C++ 中的 MapX 属性和方法
- 在您的项目中包括 MapX.cpp
- 使用 C++ 创建 MapX 控件
- 使用 C++ 创建菜单项
- 使用 C++ 处理 MapX 事件
- 使用定制工具 （C++ 示例）
- 使用 C++ 进行数据绑定
- 使用 C++ 添加快捷菜单
- 使用 C++ 的内置帮助程序对话框
- 使用 C++ 处理 MapX 异常错误
- 在 C++ 对话框中创建地图

使用 Visual C++

掌握 MapX 的一个方法就是学习示例应用程序。在文件夹 MapInfo MapX 5.0\Samples50 中可以找到示例应用程序。不过，本章将帮助您了解 MapX 和 Visual C++ 的入门知识。

理解示例应用程序

在您学习示例 C++ 应用程序 mapxsamp.cpp 的过程中，以下主题将帮助您理解 MapX。

- 从早期的 MapX 版本升级 C++ 应用程序
- 访问 C++ 中的 MapX 方法和属性
- 在您的项目中包括 MapX.cpp
- 创建 MapX 控件
- 创建菜单项
- 处理 MapX 事件
- 使用定制工具
- 数据绑定
- 添加快捷菜单
- 使用内置帮助程序对话框
- 使用 C++ 处理 MapX 异常错误
- 在 C++ 对话框中创建地图

注：这些主题假设您正使用 Microsoft 的文档 / 视图模型 （标准 MFC AppWizard 应用程序）。此示例应用程序可以与 mapxsamp.mdp 一起内置于 Developer Studio 中。

C++ 开发人员提示

如果您正使用 Visual C++ 5 或更高版本，则将示例 mapxsamp.cpp 项目添加到您的工作区中。这样您就可以迅速访问自己的项目文件右侧的示例文件和类向导。您可以随时从工作区中删除示例项目。

如果您使用早期的 MapX 版本创建了 MapX 应用程序，请参见 **“从早期的 MapX 版本升级 C++ 应用程序”**。

从早期的 MapX 版本升级 C++ 应用程序

如果您使用早期的 MapX 版本编写了 C++ 应用程序，将需要使用新的 MapX 打包程序类 （mapx.h 和 mapx.cpp）将您的应用程序升级到 MapX 的当前版本。

您可能需要修改传递给 CreateDispatch 的字符串。对于早期的 MapX 版本，您最好使用字符串来指定对象名，例如：

```
Flds.CreateDispatch("MapX.Fields")
```

对于当前的 MapX 版本，您需要指定不同的字符串：

```
Flds.CreateDispatch("MapX.Fields.4")
```

若要令您的代码与 MapX 的将来版本的兼容性更高，您最好使用 GetClsid 来代替字符串。GetClsid 返回的结果将用于 MapX 的当前和将来版本中。例如：

```
Flds.CreateDispatch(Flds.GetClsid())
```

此更改适用于您创建的所有对象。

在 MapX 对象模型中，您可以使用以下对象类创建独立的对象：

AffineTransform	BindLayer	BitmapSymols
CoordSys	数据	Feature
Fields	LayerInfo	Map
ODBCQueryInfo	部分	Point
点	Rectangle	RowValue
RowValues	Style	Variables

访问 C++ 中的 MapX 属性和方法

对象

每一 MapX 对象均由文件 MapX.h 和 MapX.cpp 中的 C++ 类实现。C++ 类的名称与 MapX 对象的名称相同，只是以 “CMapX”为前缀。例如， DataSet 对象具有名为 CMapXDataset 的类。

属性

MapX 对象的属性是由 C++ 类中的成员函数实现的。 Dataset 对象的像 “Name”属性这样的读 / 写属性将在 CMapXDataSet 中具有两个成员函数，一个成员函数用来设置属性值，另一个成员函数用来获取该属性的值。这两个成员函数名将在属性名前加前缀 “Get”或 “Set”。

```
CString GetName();
void SetName(LPCTSTR);
```

只读属性将只具有 “Get”成员函数，而不具有 “Set”成员函数。

方法

MapX 对象的方法是由与 MapX 对象的方法具有相同名称的成员函数实现的。您将注意到，其中许多参数的类型将是 “const VARIANT &”。这意味着该成员函数应为要传递给它的 VARIANT 类型的变量。

可选参数

在 Visual Basic 中或其它脚本语言中，许多方法都具有视为 “可选”的参数。在 C++ 中，必须在调用这些方法时指定所有参数。所有可选参数都是作为 VARIANT 实现的。若要使用可选参数调用 MapX 方法，您必须按如下所示设置该变量：

```
VARIANT vtOptional;
vtOptional.vt = VT_ERROR;
vtOptional.scode = DISP_E_PARAMNOTFOUND;
```

MapX.h 文件包括帮助程序类 **COptionalVariant**，它在其构造器中执行上述功能。只要您要使用可选参数调用一个方法，就可以使用该文件。

使成员函数超载

为了使用标准 C++ 类型 （而不是 VARIANTS）来更容易地调用方法，通过构造 VARIANT 并调用 “真实”成员函数的默认值，使用更简单的参数令大多数 MapX 方法超载。由于不是所有可能的组合均得到处理，因此可以随时从 CMapX 类派生您自己的类，并添加您自己的超载的成员函数。

OLE Dispatch 驱动程序

MapX.h 中的 C++ 类是从 MFC 类 COleDisplatchDriver 派生的。所有属性和方法最终以调用 IDispatch->Invoke() 来告诉 MapX 要执行的任务为结束。 COleDisplatchDriver 通常在从属性或方法返回的 IDispatch 指针上正确处理调用 Release()，但还有一些地方需要特别注意。当 MapX 对象通过其 IDispatch 接口传递给一个事件时，您必须确保不释放该指针，因为传递给事件的对象尚未由 MapX 进行 AddRef()。

返回其它对象的属性和方法

您可以将直接返回对象的方法或属性的返回值赋予同一类型的变量。

在您的项目中包括 MapX.cpp

在您的项目中包括 MapX.cpp 和 .h 文件。这两个文件包含用于对 MapX 控件进行访问的类定义和方法实现。 MapX.h 和 MapX.cpp 文件可能位于安装有 MapX 的 Samples50\CPP 子目录中。

使用 Visual C++ 版本 4

从 “插入”菜单中，选择 “将文件插入项目”。选择 MapX.cpp 作为要插入的文件。

注：不要选择 “插入” > “组件”命令。这样做将会创建一个 .cpp 文件，但该文件将是不完整的。

使用 Visual C++ 版本 5 或更高版本

从 “项目”菜单中，选择 “添加到项目” > “文件”。选择 MapX.cpp 作为要添加的文件。

警告！不要选择 “项目” > “组件和控件”命令。这样做将会创建一个 .cpp 文件，但该文件将是不完整的。

使用 C++ 创建 MapX 控件

在将包含该控件的视图中包括它：

```
#include "MapX.h"
    :
```

```
    :
    :
class CMapxSampleView : public CView
{
        :
        :
        :
    protected:
        CMapX  m_ctrlMapX;
        :
        :
    }
```

要声明表示用于 MapX 的控件 ID 的常数：

1. 转到 “视图” > “资源符号”。
2. 单击 “新建”。
3. 键入 “IDC_MAP” 来作为名称。

要在类向导中为 WM_SIZE 和 WM_CREATE 消息创建处理程序：

1. 转到 “视图” > “类向导”
2. 从 “类名称” 组合框中选择您的视图类。
3. 在消息框中，单击 “WM_CREATE”，然后单击 **“添加函数”**。
4. 接着还在该消息框中选择 “WM_SIZE” 并单击 **“添加函数”**。
5. 然后，单击 **“编辑代码”**。

在创建视图时创建该控件。在 CMapXSampView::OnCreate 中：

```
  // create map with default size
  // resize message will cause it to be
  // size to the client area of the view
  if (!m_ctrlMapX.Create(NULL, WS_VISIBLE, CRect(0,0,100,100),
    this,IDC_MAP))
      return -1;

Keep the control's size in sync with the containing window:
  // resize the map to be the same size as our client area
  void CMapxSampleView::OnSize(UINT nType, int cx, int cy)
  {
      CView::OnSize(nType, cx, cy);

      if (cx != 0 && cy != 0)
          m_ctrlMapX.MoveWindow(0,0,cx,cy,TRUE);
  }
```

像您为 WM_CREATE 消息所作的一样为 WM_SETFOCUS 消息创建新消息标头。

在我们的示例中，我们想要确保只要窗口被激活 MapX 就获得焦点：

```
  void CMapxSampleView::OnSetFocus(CWnd* pOldWnd)
  {
      CView::OnSetFocus(pOldWnd);
```

```
        m_ctrlMapX.SetFocus();
    }
```

使用 C++ 创建菜单项

您可以通过为 Doc/View 创建包含 MapX 控件的菜单来公开 MapX 功能。请参考在其中创建了 doc 模板的菜单 (mapxsamp.cpp)：

```
CMultiDocTemplate* pDocTemplate;
pDocTemplate = new CMultiDocTemplate(
    IDR_MAPXSATYPE,                 //Menu items to control MapX
    RUNTIME_CLASS(CMapxSampleDoc),
    RUNTIME_CLASS(CChildFrame),     //custom MDI child frame
    RUNTIME_CLASS(CMapxSampleView));

AddDocTemplate(pDocTemplate);
```

添加用于 MapX 菜单项的处理程序。在该示例中，这些菜单项是在 mapxsamp.rc 中的菜单部分中定义的。请注意，在该示例中，我们使用菜单命令来快速演示地图交互工具 （放大工具、半径选择工具等）。在更接近成品的应用程序中，您可以使用工具栏按钮作为用户界面，而不是使用菜单命令。

```
"Previous &View",        ID_CONTEXT_PREVIOUSVIEW
"View &Entire Map",      ID_VIEW_VIEWENTIREMAP
SEPARATOR
"&Properties...",        ID_VIEW_PROPERTIES

"&Layer Control...",     ID_VIEW_LAYERCONTROL
SEPARATOR
"&Toolbar",              ID_VIEW_TOOLBAR
"&Status Bar",           ID_VIEW_STATUS_BAR

"&Arrow",                ID_MAP_TOOL_ARROW
SEPARATOR
"Zoom &In",              ID_MAP_TOOL_ZOOMIN
"Zoom &Out",             ID_MAP_TOOL_ZOOMOUT
"&Pan",                  ID_MAP_TOOL_PAN
"&Center",               ID_MAP_TOOL_CENTER
SEPARATOR
"&Select",               ID_MAP_TOOL_SELECT
"&Radius Select",        ID_MAP_TOOL_RADIUSSELECT
"R&ectangle Select",     ID_MAP_TOOL_RECTANGLESELECT
```

一旦定义了这些菜单项 ID，就可以通过类向导添加处理程序。在您的包含 MapX 控件的类中，为 ID_MAP_TOOL_ARROW、 ID_MAP_TOOL_ZOOMIN 等创建处理程序。

```
// tell MapX what the current tool is
void CMapxSampleView::OnMapToolArrow()
{
    m_ctrlMapX.SetCurrentTool(miArrowTool);
}
void CMapxSampleView::OnMapToolZoomin()
{
    m_ctrlMapX.SetCurrentTool(miZoomInTool);
}
// switch to the previous view
void CMapxSampleView::OnContextPreviousview()
{
    m_ctrlMapX.ZoomTo(m_dPrevZoom, m_dPrevX, m_dPrevY);
}
```

一旦选择了该工具，内置的 MapX 功能就可以处理缩放、选择等操作。

使用 C++ 处理 MapX 事件

若要处理 MapX 事件，您首先需要为感兴趣的事件建立 eventsink 地图。用于事件 DISPATCH id 的常数是在 MapX.h 中为 MapX 定制事件定义的，并且是在 <olectl.h> 中为 OLE 普通事件定义的。

```
// From MapX.h
#define MAPX_DISPID_SELECTION_CHANGED 0x1
#define MAPX_DISPID_RESOLVEDATABIND 0x2
#define MAPX_DISPID_TOOLUSED 0x3
#define MAPX_DISPID_REQUESTDATA 0x4
#define MAPX_DISPID_DATAMISMATCH 0x5
#define MAPX_DISPID_MAPVIEWCHANGED 0x6
#define MAPX_DISPID_ANNOTATIONADDED 0x7
#define MAPX_DISPID_ANNOTATIONCHANGED 0x8
#define MAPX_DISPID_THEMEMODIFYREQUESTED 0x9
#define MAPX_DISPID_DRAWUSERLAYER 0x0a
#define MAPX_DISPID_POLYTOOLUSED 0x0b
// From <olectl.h>
#define DISPID_CLICK                    (-600)
#define DISPID_DBLCLICK                 (-601)
#define DISPID_KEYDOWN                  (-602)
#define DISPID_KEYPRESS                 (-603)
#define DISPID_KEYUP                    (-604)
#define DISPID_MOUSEDOWN                (-605)
#define DISPID_MOUSEMOVE                (-606)
#define DISPID_MOUSEUP                  (-607)
```

```
#define DISPID_ERROREVENT                (-608)
```

EVENT_SINK 中的 ON_EVENT 宏也为 MapX 控件指定一个 ID （在该示例中为 IDC_MAP），指定该事件的参数，并且指定事件处理程序方法的名称。

在视图标头文件 (her, MapXSampView.h) 中，将 “DECLARE_EVENTSINK_MAP ()”一行放置于“DECLARE_MESSAGE_MAP”行之下。

从 mapxsampview.cpp：

```
BEGIN_EVENTSINK_MAP(CMapxSampleView, CView)
    ON_EVENT(CMapxSampleView, IDC_MAP, DISPID_MOUSEMOVE,
    OnMouseMoveInMap,VTS_I2 VTS_I2 VTS_XPOS_PIXELS VTS_YPOS_PIXELS)
    ON_EVENT(CMapxSampleView, IDC_MAP, MAPX_DISPID_MAPVIEWCHANGED,
    OnMapViewChanged, VTS_NONE)
    ON_EVENT(CMapxSampleView, IDC_MAP, DISPID_MOUSEUP,
    OnMouseUpInMap, VTS_I2 VTS_I2 VTS_XPOS_PIXELS VTS_YPOS_PIXELS)
    ON_EVENT(CMapxSampleView, IDC_MAP, MAPX_DISPID_TOOLUSED,
    OnToolUsed, VTS_I2 VTS_R8 VTS_R8 VTS_R8 VTS_R8 VTS_R8 VTS_BOOL
    VTS_BOOL VTS_PBOOL)
    ON_EVENT(CMapxSampleView, IDC_MAP,
    MAPX_DISPID_THEMEMODIFYREQUESTED, OnThemeModifyRequested,
    VTS_DISPATCH)
END_EVENTSINK_MAP()
```

随后是用于 OnToolUsed 事件的事件处理程序代码。来自 mapxsampview.h 的声明：

```
void OnToolUsed(short ToolNum, double X1, double Y1, double X2,
double Y2, double Distance, BOOL Shift, BOOL Ctrl, BOOL*
EnableDefault);
```

… 以及来自 mapxsampview.cpp 的实现方式 （我们只使用 TRACE 宏将参数输出到调试窗口）：

```
void CMapxSampleView::OnToolUsed(short ToolNum, double X1, double Y1,
double X2, double Y2, double Distance, BOOL Shift, BOOL Ctrl, BOOL*
EnableDefault)
{
    CString str;
    str.Format("Tool=%d, [%f,%f] [%f, %f], dist=%f, %s %s\n",
        ToolNum, X1,Y1,X2,Y2,Distance,
       (Shift)?"Shift":"",(Ctrl)?"Ctrl":"");
    TRACE(str);
}
```

您通常应通过此函数处理定制工具，或者覆盖用于内置工具的 MapX 的默认行为。有关示例，请参见 **“使用定制工具”**。

注：对于其对象作为参数传递的事件，事件处理程序不应更改该对象的引用计数。有关示例，请参见 **“使用 C++ 处理 MapX 异常错误”**。

使用定制工具 （C++ 示例）

一旦您已使用 CreateCustomTool 方法创建了定制工具后，就可以在用户使用定制工具时使用 ToolUsed 事件执行操作。

此示例进行测试以了解正使用哪一工具。然后，根据正在使用的工具，该示例或者：

- 更改光标下的地图图元的样式，或者
- 在用户单击处放置新符号。

```
void CMapxSampleView::OnToolUsed(short ToolNum, double X1,
          double Y1, double X2, double Y2, double Distance,
             BOOL Shift, BOOL Ctrl, BOOL* EnableDefault)
{
    CString str;
    CMapXPoint pnt;

    str.Format("Tool=%d, [%f,%f] [%f, %f], dist=%f, %s %s\n",
        ToolNum, X1,Y1,X2,Y2,Distance,
        (Shift)?"Shift":"",(Ctrl)?"Ctrl":"");
    TRACE(str);
    // change the style of the feature under the cursor
    if (ToolNum == MAP_TOOL_CHANGESTYLE) {
        try {
            // Need the dispatch to use the point
            if (pnt.CreateDispatch(pnt.GetClsid())) {
                pnt.Set(X1, Y1);
            }
            else {
                // something went wrong, can't use the point...
                AfxThrowOleException(CO_E_CLASS_CREATE_FAILED);
            }

            CMapXLayers layers = m_ctrlMapX.GetLayers();
            // Get the USA feature under the cursor
            CMapXFeatures    ftrs =
             layers.Item("USA").SearchAtPoint(LPDISPATCH(pnt));
            // work on only the first feature
            CMapXFeatureftr = ftrs.Item(1);
            // get the style object from the feature
            CMapXStylestyle = ftr.GetStyle();

            style.SetRegionBackColor(255);
            // update the feature in the layer
            ftr.Update();
        }
        catch (COleDispatchException *e) {
            e->ReportError();
```

```
                e->Delete();
            }
            catch (COleException *e) {
                e->ReportError();
                e->Delete();
            }
        }
        // place a new symbol at the point clicked on
        else if (ToolNum == MAP_TOOL_NEWPOINT) {
            try {
                CMapXLayers layers = m_ctrlMapX.GetLayers();
                CMapXFeatureftr;
                // Need the dispatch id to use the feature
                if (ftr.CreateDispatch(ftr.GetClsid())) {
                    // Symbol feature
                    ftr.SetType(miFeatureTypeSymbol);
                    // Get the point object from the feature
                    // and call the Set method
                    ftr.GetPoint().Set(X1, Y1);
                    // Add it to the layer
                    layers.Item("USA").AddFeature(ftr);
                }
                else {
                    AfxThrowOleException(CO_E_CLASS_CREATE_FAILED);
                }
            }
            catch (COleDispatchException *e) {
                e->ReportError();
                e->Delete();
            }
            catch (COleException *e) {
                e->ReportError();
                e->Delete();
            }
        }
    }
```

使用 C++ 进行数据绑定

CMapXSample 应用程序具有用于以下数据集类型的数据绑定示例：miDataSetDAO、 miDataSetODBC、miDataSetUnbound、 miDataSetGlobalHandle。这些示例还说明如何使用动态数据绑定、添加 XY 点的图层、 ZIP 代码匹配和自动匹配。

请参见 mapxsampview.cpp 文件来查看这些示例。下面是使用 miDataSetGlobalHandle 类型从文件添加数据的菜单命令处理程序之一。这些数据的格式需要是：用引号括起字符串，在字段之间用制表符，并且在记录末尾使用回车符 / 换行符。例如：

```
"\"NY\"\t105.34\t100\t1\r\n"
"\"MA\"\t245.19\t200\t2\r\n"
"\"NY\"\t195.0\t300\t3\r\n"
"\"AK\"\t195.0\t125\t4\r\n"
"\"CA\"\t56.453\t200\t5\r\n";
```

请注意，此类型的数据绑定 (miDataSetGlobalHandle) 只是支持的若干数据集类型中的一种。

CMapxSampleView::OnMapAdddata() 函数处理 “地图”菜单上的菜单项。它以上述形式提示包含数据的文件名、读取该文件并将其添加到地图的数据集集合中。

```
void CMapxSampleView::OnMapAdddata()
{
  :
  :
  CFileDialog dlgFile(TRUE, "*.txt", NULL, 0, szDataFilter, this);

  if (dlgFile.DoModal() == IDCANCEL)  // User cancelled the dialog
    return;

  // Read file into a string, and copy it to a global memory buffer
  :
  :
  // Allocate the memory buffer, copy the string into it
  :
  :
  // Declare the variables that will be parameters to
  // DataSets.Add()
  short Type;
  VARIANT SourceData, Name, GeoField,
                      SecondaryGeoField, BindLayerName, Fields;
  CString strName= "TestData";

// set up optional parameters; most will not be used
// Note: you could also use the line
//COptionalVariant SecondaryGeoField;
// instead
  SecondaryGeoField.vt = VT_ERROR;
  SecondaryGeoField.scode = DISP_E_PARAMNOTFOUND;

  // let mapx auto detect geofield
  GeoField.vt = VT_ERROR;
  GeoField.scode = DISP_E_PARAMNOTFOUND;

  // let mapx find which layer to bind to
  BindLayerName.vt = VT_ERROR;
  BindLayerName.scode = DISP_E_PARAMNOTFOUND;
```

```
    // use all fields with defaults
    Fields.vt = VT_ERROR;
    Fields.scode = DISP_E_PARAMNOTFOUND;

    // set the name of our dataset
    Name.vt = VT_BSTR;
    // Remember to SysAlloc() the string; it's going into a BSTR
    Name.bstrVal = strName.AllocSysString();

    // set up source data - no error checking on alloc
    Type = miDataSetGlobalHandle;
    SourceData.vt = VT_I4;
    SourceData.lVal = (long)hGlobalData;
    try {
      // now add the dataset to the datasets collection
      CMapXDataset ds = m_ctrlMapX.GetDatasets().Add(Type,
      SourceData, Name, GeoField, SecondaryGeoField, BindLayerName,
      Fields);
    }
    catch (COleDispatchException *e) {
      e->ReportError();
      e->Delete();
    }
    SysFreeString(Name.bstrVal)
  }
```

使用 C++ 添加快捷菜单

DISPID_MOUSEUP 事件可用于添加快捷 （鼠标右键按钮）菜单。为您的快捷菜单创建菜单资源 （在示例应用程序中为 IDR_CONTEXTMENU）。 OnMouseUpInMap 处理程序类似于以下代码：

```
    // if right mouse button, display the context menu
    BOOL CMapxSampleView::OnMouseUpInMap(
        short Button,short Shift,OLE_XPOS_PIXELS x,OLE_YPOS_PIXELS y)
    {
      if (Button == 2) {  // right button
          CMenu menu;         // top-level menu
          CMenu *pMenu=NULL;  // pop-up menu

          // Load the menu resource.
          menu.LoadMenu(IDR_CONTEXTMENU);

          // TrackPopupMenu cannot display the top-level menu, so get
          // the handle of the first pop-up menu.
          pMenu = menu.GetSubMenu(0);
          if (!pMenu) {
              return TRUE;
          }
```

```
            SetMenuDefaultItem(pMenu->m_hMenu, ID_VIEW_PROPERTIES,
            FALSE);

            // Display the floating pop-up menu. Track the right mouse
            // button on the assumption that this function is called
            // during WM_CONTEXTMENU processing.
            POINT pt;
            GetCursorPos(&pt);

             pMenu->TrackPopupMenu(TPM_LEFTALIGN | TPM_RIGHTBUTTON,
             pt.x, pt.y, this, NULL);

            // Destroy the menu.
            menu.DestroyMenu();
        }
        return TRUE;
    }
```

使用 C++ 的内置帮助程序对话框

使用内置的帮助程序对话框十分简单。有一个用来调用 “MapX 属性”对话框的菜单项。它可以使用 ClassWizard 挂钩到该处理程序上；下面说明如何将 ClassWizard 放置于 mapxsamp.cpp 中：

```
ON_COMMAND(ID_VIEW_PROPERTIES, OnViewProperties)
```

并且处理程序：

```
void CMapxSampleView::OnViewProperties()
{
    // easiest way to bring up property page
  try {
  m_ctrlMapX.PropertyPage();
      }
      catch (COleDispatchException *e) {
          e->ReportError();
          e->Delete();
      }
      catch (COleException *e) {
          e->ReportError();
          e->Delete();
      }
}
```

注：建议您不要在完成的应用程序中包括属性页。它在编写和调试应用程序时可以起到很大帮助作用，但它不适合由最终用户使用，因为它提供对过多属性的直接访问，并且不可配置。应代之以使用普通对话框 （例如 LayersDlg）。

同样，“图层控件”对话框：

```
void CMapxSampleView::OnViewLayercontrol()
```

```
{
    try {
        // mark as optional since we don't have a helpfile
        COptionalVariant vHelpFile, vHelpID;

        CMapXLayers layers = m_ctrlMapX.GetLayers();
        layers.LayersDlg(vHelpFile, vHelpID);
    }
    catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    }
    catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

使用 C++ 处理 MapX 异常错误

MapX 通过报告（引发）COleDispatchException 来报告大多数错误。在调用 MapX 时捕获异常错误是十分重要的，因为 MFC 的默认异常错误处理程序不捕获它们并且您的应用程序将退出。在消息框中显示对错误的说明也是非常有用的。错误消息的文本将帮助您确定是否正确使用了 MapX 图元，或者是否已发生某一其它问题。

COleDispatchException 类还在公共成员 m_wCode 中包括错误代码，以便您的程序可以标识和处理不同类型的错误。对于错误代码及其说明的列表，请参见联机帮助中的 **MapX 错误代码**。

如果在尝试调用 MapX 时具有一般的 OLE 错误 （在您使用过时的 MapX.h 或 MapX.cpp 版本时可能发生），则 MFC 将引发 COleException。建议你在调用 MapX 时捕获全部两个异常错误类型。

MapX 也可能在 Error 事件中传递错误 （尽管非常少见）。只有在不从 MapX 属性或方法直接调用的异步处理 （像 redraw）期间出现某种类型的错误时，才可能发生上述情况。

异常错误处理示例

在此示例的 ThemeModifyRequested 事件的事件处理程序中，我们显示普通的 “主题” 属性对话框以让用户更改主题颜色等。请注意异常错误处理的方式。

```
// Note: in objects passed to events, the event handler
// does not change the reference count
// ie: do not call release on the object
void CMapxSampleView::OnThemeModifyRequested(LPDISPATCH Theme)
{
    try {
        CMapXTheme theme;
        COptionalVariant vHelpFile, vHelpID;
        // mark as optional since we don't have a helpfile

        theme.AttachDispatch(Theme, FALSE); // don't auto release
        theme.ThemeDlg(vHelpFile, vHelpID);
        // could decide to bring up legend dlg here instead
        //CMapXLegend leg(theme.GetLegend());
        //leg.LegendDlg(vHelpFile, vHelpID);
    }
    catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    }
    catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
}
```

在 C++ 对话框中创建地图

1. 将 MapX 插入对话框中。
   - 在用于您要插入 MapX 的对话框模板的资源编辑器中，右击并从菜单中选择 “插入 OLE 控件”。然后选择 “MapInfo MapX v5.0”。
2. 为该对话框创建 CDialog 派生类。
   - 突出显示 MapX 并按下 Ctrl+W 来显示 ClassWizard。在出现提示后选择 “创建新类”，并且输入对话框的名称 （该示例使用 “CSampleDlg”）。按 “确定” 将创建 SampleDlg.h 和 SampleDlg.cpp 文件。
3. 将 CMapX 作为对话框类的成员添加。
   - 首先，在对话框的标头文件中添加 `#include "MapX.h"`，然后添加用于 MapX 的成员变量。

注：不要将类向导生成的打包程序类用于 MapX。它们是不完全的。使用 Sample50\CPP 目录中随

MapX 一起包括的 MapX.h 和 MapX.cpp 文件。

在 C++ 对话框中创建地图

```
#include "MapX.h"

/////////////////////////////////////////////////////////////////////////////
// CSampleDlg dialog

class CSampleDlg : public CDialog
{
// Construction
public:
    CSampleDlg(CWnd* pParent = NULL);  // standard constructor

// Dialog Data
    //{{AFX_DATA(CSampleDlg)
    enum { IDD = IDD_MAPXDLG };
    //}}AFX_DATA
.
:
    CMapX m_ctrlMapX;
.
:
};
```

使用 DDX_Control 将打包程序类链接到 MapX 控件

使用 DDX_Control 按如下所示将一行插入对话框类的 DoDataExchange 方法中：

```
void CSampleDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CSampleDlg)
    DDX_Text(pDX, IDC_STATICX, m_strX);
    DDX_Text(pDX, IDC_STATICY, m_strY);
    //}}AFX_DATA_MAP

    // IDC_MAP1 is the ID given to the mapx control when it
    // was inserted into the dialog template.
    DDX_Control(pDX, IDC_MAP1, m_ctrlMapX);
}
```

使用 MapX 属性、方法和对象来控制地图

```
// example use of mapx to create a temporary layer and create
// some custom mapx tools to be used later.
BOOL CSampleDlg::OnInitDialog()
```

```
{
    CDialog::OnInitDialog();

    try {
        m_ctrlMapX.SetZoom(1000);  // zoom in to 1000 miles

        // create temporary layer on top for drawing on.
        CMapXLayer layer =
        m_ctrlMapX.GetLayers().CreateLayer("scratch layer",NULL,1);

        // make it the animation layer
        m_ctrlMapX.GetLayers().SetAnimationLayer(layer);

        // create some custom tools for object drawing
        m_ctrlMapX.CreateCustomTool(MYTOOL_SYMBOL,
        miToolTypePoint, miCrossCursor);
        m_ctrlMapX.CreateCustomTool(MYTOOL_LINE, miToolTypePoly,
        miCrossCursor);
        m_ctrlMapX.CreateCustomTool(MYTOOL_REGION, miToolTypePoly,
        miCrossCursor);
        m_ctrlMapX.CreateCustomTool(MYTOOL_TEXT, miToolTypePoint,
        miIBeamCursor);
        m_ctrlMapX.SetCurrentTool(MYTOOL_REGION);
    }
    catch (COleDispatchException *e) {
        e->ReportError();
        e->Delete();
    }
    catch (COleException *e) {
        e->ReportError();
        e->Delete();
    }
    return TRUE; // return TRUE unless you set the focus to a
                 //control. EXCEPTION: OCX Property Pages should
                 // return FALSE
}
```

使用类向导处理事件

在对话框中处理 MapX 事件是非常简单的。只需使用类向导 “消息地图”选项卡，突出显示 MapX 的 ID (IDC_MAP1)，将列出事件。突出显示一个事件并按下 “添加函数”按钮，类向导会将事件处理程序和 EVENT_SINK 项自动添加到您的对话框类中。

15 章

- 标准工具概述
- 对象编辑工具
- 节点选择和编辑
- 创建定制工具
- 创建多边形绘制工具 (Polytools)

MapX 工具

大多数地图绘制应用程序提供各种工具来协助完成常见的绘图任务 （例如在地图上绘制线条）和导航任务 （例如放大）。 MapX 提供若干常见地图绘制工具，并且您还可以创建自己的定制工具。

标准工具概述

使用标准工具

使用 MapX，您可以很容易地将常见工具栏按钮并入应用程序中。 MapX 提供对若干常见地图绘制工具的内置支持，包括：

- 令用户更改地图的比例和 / 或位置的导航工具 （放大、缩小、平移、居中）。
- 让用户单击地图图元以给它加标签的加标签工具。
- 为用户提供各种方法来选择地图图元的一组选择工具。
- 对象创建工具，用于创建新的地图图元。

提供对修改键 （SHIFT 键、 CTRL 键）的内置支持的选择工具：使用选择工具的同时按住 SHIFT 键；该工具将取消选择图元；使用选择工具的同时按住 CTRL，该工具会将图元添加到选择中。只要按下修改键 MapX 就会自动显示不同的光标 （加号或减号出现在该光标旁），以便用户可以理解该键的用途。

控制哪一工具作为当前工具

若要设置所使用的工具，请设置 **Map.CurrentTool 属性**。

若要激活某一标准工具，请将该属性设置为某一 ToolConstant。例如，要将工具更改为 “放大”工具：

```
Map1.CurrentTool = miZoomInTool
```

若要激活一个定制工具，请使用您在使用 CreateCustomTool 方法时指定的 **ToolNumber** 值。

```
Map1.CurrentTool = 99
```

可用的标准工具

不同的工具将使鼠标可以执行多种任务。例如，如果当前工具被设置为 miLabelTool，则在您单击鼠标时，它会给该特定地图对象加标签。鼠标光标将基于您所使用的工具进行更改。

随 MapX 提供了一些标准工具：

工具	常数	说明
添加线条	miAddLineTool	将线条图元添加到插入图层中。
添加点	miAddPointTool	单击该工具可将点图元添加到插入图层中。
添加折线	miAddPolyLineTool	将折线图元添加到插入图层中。
添加区域	miAddRegionTool	将区域图元添加到插入图层中。
箭头	miArrowTool	单击标题或注释此外，在可编辑图层中移动选定图元或调整选定图元的大小。
居中	miCenterTool	单击该工具可以重新将地图居中。
加标签	miLabelTool	在一个图元上单击可以给该图元加标签。
平移	miPanTool	拖动该工具可以重新将地图居中。
多边形选择	miPolygonSelectTool	单击该工具可以绘制一个多边形；该多边形内的对象将被选定。

工具	常数	说明
半径选择	miRadiusSelectTool	拖动该工具可以选择半径内的图元。
矩形选择	miRectSelectTool	拖动该工具可以选择矩形内的图元。
选择工具	miSelectTool	单击该工具可以选择图元。
符号	miSymbolTool	放置符号注释。
文本	miTextTool	放置文本注释。
放大	miZoomInTool	放大。
缩小	miZoomOutTool	缩小。

对象编辑工具

通过对象编辑工具，用户可以在地图图层中创建和修改图元。有四种标准的对象创建工具：“添加点”、“添加线条”、“添加折线”和 “添加区域”。

注：若要设置所使用的工具，请设置 **Map.CurrentTool 属性**。

对象编辑工具将新图元添加到在 **Layers.InsertionLayer 属性**中指定的任何图层中。只能有一个插入图层，并且默认情况下是没有插入图层。在没有任何插入图层时将当前工具设置为对象创建工具将导致错误。

MapX 还支持对现有地图图元的修改。若要编辑图元，则对于您要更改的任何图层，必须将 Layer.Editable 属性设置为真。然后，可以使用内置的 “箭头”工具，在当前选择中移动图元或调整图元的大小。若要移动选定图元，只需单击并在选择中拖动。若要调整选定图元的大小，只需单击并用编辑手柄拖动。若要删除选定图元，按 Delete 键。

节点选择和编辑

节点选择和编辑是添加到选择工具中的功能。通过该功能，用户可以添加、更改或删除 “节点对象”中的节点。即，节点编辑功能使用户可以控制具有节点的对象的形状。

在地图处于 “节点编辑模式”时，可以从任何选择的、可编辑的节点对象选择节点。这包括多个图层上来自多个对象的多个不连续节点。默认情况下， MapX 处于 “图元编辑模式”。为了选择和编辑节点对象，必须将 MapX 设置为 “节点编辑模式”。

注：当前，在节点编辑中不支持事务处理。在选择节点时，请务必小心，不要无意中敲击 Delete 键。此时，没有 “撤消”功能。

下表显示在 “节点编辑模式”下可以使用哪些对象：

节点对象	可以选择节点	可以添加节点	无节点对象 （将不在 “节点编辑模式”下工作）
• 点 • 线条 • 折线 • 区域	• 点 • 线条 • 折线 • 区域	• 线条 • 折线 • 多边形	• 弧线 • 椭圆 / 圆形 • 矩形 • 圆角矩形 • 文本

使用 Map.FeatureEditMode 属性设置模式

地图对象可以设置为 “图元编辑模式”或 “节点编辑模式”（其默认值是 “图元编辑模式”）。在 “图元编辑模式”中，用调整大小手柄绘制选定的可编辑对象 （通常突出显示对象的 MBR 的 4 个角）。在 “节点编辑模式”中，绘制为进行编辑选定的对象的节点，并且只有节点对象 （点、线条、折线、区域）将显示可选择的节点。只突出显示非节点对象。

下面是用于设置 **Map.FeatureEditMode 属性**的 Visual Basic 命令。

图元编辑模式

```
Map1.FeatureEditMode = miEditModeFeature
```

节点编辑模式

```
Map1.FeatureEditMode = miEditModeNode
```

具有移动和删除重复节点的节点编辑模式

```
Map1.FeatureEditMode = miEditModeNode | miMoveDuplicateNodes |
miDeleteDuplicateNodes
```

节点编辑模式和添加节点模式

```
Map1.FeatureEditMode = miEditModeNode | miEditModeAddNode
```

节点选择和编辑条件

若要选择节点，地图必须处于 “节点编辑模式”，而非 “添加节点模式”。下面是用于使用选择工具选择节点的 “规则”：

- 在未选择的节点上单击以选择该节点。这将清除当前选择的任何节点。或者，如果您在当前选择的节点上单击，其它选择的节点将不更改。
- 按住 Ctrl 键的同时单击可以在当前节点的各状态 （已选择和未选择）之间切换。这将不会影响以前选择的任何节点。使用该功能可以禁用单独的节点，或者指示节点的起始范围 （请参见下面的 “选择节点范围”）。
- 按住 Shift 键的同时单击可以指定节点的结束范围。对于多边形，这将选择节点的最短范围。
- 按住 Ctrl 和 Shift 键的同时单击将在多边形或区域中选择节点的最长范围。

注：此操作仅适用于区域的多边形。它选择与按住 Shift 键的同时单击相反的节点范围。

- 在用户单击或按住 Ctrl 键的同时单击一个节点时定义 “锚节点”。也就是说，当选择了一个节点范围时，该范围中每一节点的当前状态由锚节点 （即，节点的 “路径”中的第一个节点）的状态决定。如果选择了该锚节点，则该范围中的所有节点均被选定。

选择节点范围

下面介绍如何选择多个节点范围：

1. 单击 （或在按住 Ctrl 键的同时单击）对象中的一个节点。这是该范围的 “锚节点”（即起始点）。
2. 按住 Shift 键的同时单击该对象中的另一个节点。这将定义第一个范围的结束点。
3. 按住 Ctrl 键的同时单击该对象中的第三个节点。这将定义第二个范围的起始锚节点。
4. 按住 Shift 键的同时单击该对象中的第四个节点。这将完成第二个范围。
5. 为您要选择的其它每一范围重复第一步和第二步。

为了定义节点的范围，起始点和结束点必须位于同一多边形或折线对象中。对于多 – 多边形 / 多 – 折线，该范围只能在单多边形 / 折线内。在多对象内范围不能跨对象或单独的折线 / 多边形。

移动节点

在选择了一个或多个节点后，就可以移动它们。要进行移动，在以前选择的节点上单击，然后将它拖到新位置。在拖动时，绘制连接其新位置内所有当前选定节点的虚线。在松开鼠标按钮后，所有选定的节点都将移到该新位置。如果在拖动时按下 ESC，则操作将中止并且所有节点都将返回到其原始位置。

在移动后，已移动的所有节点仍被选定。

注：如果任何修改的对象是多 - 多边形，并且其 polyid 被重新排序，将取消选择所有节点。例如，如果多边形的相对大小发生更改，则其顺序可能更改，从而将取消选择所有节点。

删除节点

在选择了一个或多个节点后，可以通过按 Delete 键删除它们。在删除了这些节点后，将没有任何选定的节点。在删除了节点后，如果剩余的节点不足以定义其类型的对象，将删除该对象。例如，多边形要求 3 个或更多节点。如果从包含 6 个节点的多边形上删除 4 个节点，则该多边形将被删除，因为剩余的 2 个节点无法定义一个多边形。

地图必须保持 “焦点”，才可以删除节点。即，如果用户从可编辑图层上选择一个节点对象，而地图处于 “节点编辑模式”（选择了一个或多个节点），然后该用户单击一个控件 （例如用来切换删除重复节点状态的复选框），则地图将失去焦点。因此，在用户按下 Delete 键时 （即使选择点工具处于活动状态并且选择了节点），节点也将不删除。为了使 Delete 键正确工作，应用程序必须在激活该控件后设置指向地图的焦点 （称作 setFocus）。

移动重复节点

在移动节点时，可以选择只移动选定的节点，或者同时移动选定的节点和重复节点。（重复节点是在同一图层中属于其它对象的节点。）在启用 “移动重复节点”后，也将移动与选定节点完全匹配的同一图层中相邻对象内的所有节点。

删除重复节点

在删除节点时，可以选择只删除选定的节点，或者删除同一图层内选定的节点和重复节点。在启用“删除重复节点”后，也将删除与选定节点完全匹配的同一图层中相邻对象内的所有节点。

添加节点

若要添加节点，地图必须处于 “节点编辑模式”和 “添加节点模式”。节点只能添加到线条、折线和多边形上。如果将节点添加到线条上，该线条将自动转为折线。节点只能添加到选定的可编辑对象上。用户必须在线条、折线或多边形对象的边缘上的两个节点之间单击。在单击鼠标按钮时鼠标光标的位置将确定新节点将插入其间的两条虚线。鼠标的移动将显示表示新对象的形状的两条虚线。松开鼠标按钮时鼠标光标的位置将确定新节点的位置。在拖动期间按下 ESC 键 （在松开鼠标按钮前）将中止该操作，并且将不添加节点。

粘附节点

若要使节点选择和编辑操作更容易，您可以启用 MapX 的粘附节点功能。使用 Map 对象的两个属性来启用和定义此功能。Map.SnapToNodeSupport 用来启用或禁用此功能。Map.SnapTolerance 用于以像素为单位确定粘附节点操作适用的距离，即光标在其自动移动到最近节点前距一个节点的远近程度。

创建定制工具

定制工具

如果需要 MapX 未提供的工具按钮类型，只需通过使用 Map.CreateCustomTool 方法创建一个定制工具。

在您创建定制工具时，可以控制所创建的工具的类型 -- 换言之，您选择此工具是否允许用户单击、单击和拖动以绘制线条、或者单击和拖动以绘制矩形。您还选择在使用某一定制工具时显示哪一光标。

在您为任何应用程序创建一个定制工具时，通常要执行三步：

1. 创建工具。
2. 编写工具处理程序 （工具实际执行功能的代码）。
3. 使用该工具 （令用户开始使用工具）。

创建工具

若要创建一个定制工具，应调用 CreateCustomTool 方法。

此示例创建定制的标尺工具。该标尺工具用于确定地图上两点之间的距离。首先，我们声明常数 RULERTOOLID 等于 500 来表示我们的定制工具。然后，在我们装载该应用程序的主要形式时，创建该工具。

```
Const RULERTOOLID = 500   his goes in the General declarations.

Private Sub Form_Load()
Map1.CreateCustomTool RULERTOOLID, miToolTypeLine, miSizeCursor
```

```
End Sub
```

在上面对 CreateCustomTool 的调用中，我们指定了三个必需的参数：*ToolNumber*、 *Type* 和 *Cursor*。 ToolNumber 是我们创建用来表示该工具的 RULERTOOLID 常数。 Type 是确定该工具的行为的 ToolTypeConstants 值。在此例子中，该值为 miToolTypeLine，使用户可以使用该工具单击并拖动以绘制线条。 Cursor 为 miSizeCursor，表示在选择该工具后它将显示为大小光标。

CreateCustomTool 具有两个也采用 CursorConstants 值的可选参数。 CursorConstants 指定您可以用于定制工具的光标。它们在定制工具为 CurrentTool 时定义光标形状。

ShiftCursor 指定在按住 <SHIFT> 键时工具的光标形状。 CtrlCursor 指定在按住 <CTRL> 键时的光标形状。它们在您要将您的工具行为与这些键相关联时会非常有用。

若要使您的定制工具成为活动工具，应设置 CurrentTool 属性。例如，您可以将一个按钮放置于 Visual Basic 窗体上，并且当用户单击该按钮时，将设置 CurrentTool 属性。

可用的定制工具类型

ToolTypeConstants 描述您在创建定制工具时可以使用的工具类型。它们描述工具的行为 （例如，miToolTypeLine 令用户绘制线条； miToolTypeCircle 令用户绘制圆形等）。

常数	行为
miToolTypePoint	指示特定的点。
miToolTypeLine	绘制线条。
miToolTypeCircle	绘制圆形。
miToolTypeMarquee	绘制选择该框内的地图对象的矩形框。
miToolTypePoly	绘制折线。
miToolTypePolygon	绘制多边形。

创建了该标尺定制工具后，需要编写工具实际执行功能的代码。

编写工具处理程序

有两个不同的时间可用来执行工具的代码：在该工具的使用期间，或在使用该工具后。对于我们的定制标尺工具，我们希望在工具使用期间执行代码，因为需要在该期间执行工作。

该定制的标尺工具的操作方法如下：当用户在某一地图位置上用它单击时，将标记该工具的距离单位的起始点。用户按下鼠标按钮并在地图上移动鼠标，然后在另一位置停止该鼠标。这将标记工具的距离度量的结束点。该标尺工具计算两点之间的距离。为此，我们需要在用户按下鼠标按钮时以及用户在地图上移动鼠标并在其它点停止时执行该标尺工具的代码。若要在用户单击时捕获起始点，我们在 MouseDown 事件中编写了以下代码：

```
Dim XDown As Double
Dim YDown As Double

Private Sub Map1_MouseDown(Button As Integer, _
          Shift As Integer, X As Single, Y As Single)
```

```
        If Map1.CurrentTool = RULERTOOLID And Button = vbLeftButton _
        Then
            Map1.ConvertCoord X, Y, XDown, YDown, miScreenToMap
        End If
    End Sub
```

在用户单击鼠标时，将引发 MouseDown 事件。 MouseDown 事件中的节点采用用户单击处的 X,Y 点，然后将其从屏幕坐标转换为地图坐标，并且将其存储在全局变量 Xdown 和 Ydown 中。 Xdown 和 Ydown 表示距离度量的起始点。

为了当用户在地图上移动鼠标和在其它点停止时捕获结束点，我们在 MouseMove 事件中编写了以下代码：

```
    Private Sub Map1_MouseMove(Button As Integer, _
              Shift As Integer, X As Single, Y As Single)

        If Map1.CurrentTool = RULERTOOLID And Button = vbLeftButton _
        Then
          Dim MapCoordX As Double, MapCoordY As Double
          Map1.MapUnit = miUnitMile
          Map1.ConvertCoord X, Y, MapCoordX, _
                MapCoordY, miScreenToMap
          fMainForm.sbStatusBar.SimpleText = Map1.Distance(XDown, _
                 YDown, MapCoordX, MapCoordY) & " miles"
        End If
    End Sub
```

此代码将继续执行，直到用户停止移动鼠标并松开鼠标按钮。最后执行此代码是在鼠标停止移动前用于地图上的最后一点，该点将是我们的距离度量的结束点。我们将该点从屏幕坐标转换为地图坐标，然后使用 Map 对象的 Distance 方法来计算这两个点之间的距离。结果将以状态栏的形式显示。

ToolUsed 事件

您可能经常要在使用一个工具后执行该工具的代码。例如，允许用户绘制一个圆形，并在松开鼠标后，计算该半径内客户的数目。如果这是该工具需要的功能，则使用 Map 对象的 ToolUsed 事件放置您的代码。

令用户开始使用工具

若要 “令用户开始使用工具”，请将 Map 对象的 CurrentTool 属性设置为该工具：

```
    Map1.CurrentTool = miZoomInTool
```

- 或者 -

```
    Map1.CurrentTool = 99
```

创建多边形绘制工具 (Polytools)

polytool 是一种让用户通过重复单击 （举例）来绘制多边形或折线的工具。

标准 Polytool

MapX 提供标准的多边形选择工具。通过该工具，用户可以绘制多边形，然后选择其质心属于该多边形的所有图元。只搜索可选择的图层。忽略 Seamless、 Raster 和 Userdraw 层。

若要激活此工具，请将 CurrentTool 属性设置为 miPolygonSelectTool （值：1010）。

当使用此工具时，引发 PolyToolUsed 事件。如果用户以双击、空格或回车结束该多边形 (miPolyToolEnd)，则执行搜索，并且将 miPolyToolEnd (1) 传递给 PolyToolUsed 事件。如果用户以 ESC 键结束该多边形，则取消搜索； miPolyToolEndEscaped (2) 被传递给 PolyToolUsed 事件。用户可以删除一个线条的所有节点 （退格键或 Delete 键）。在删除了最后一个节点后，它将发送 (2) 标记。

定制 Polytool

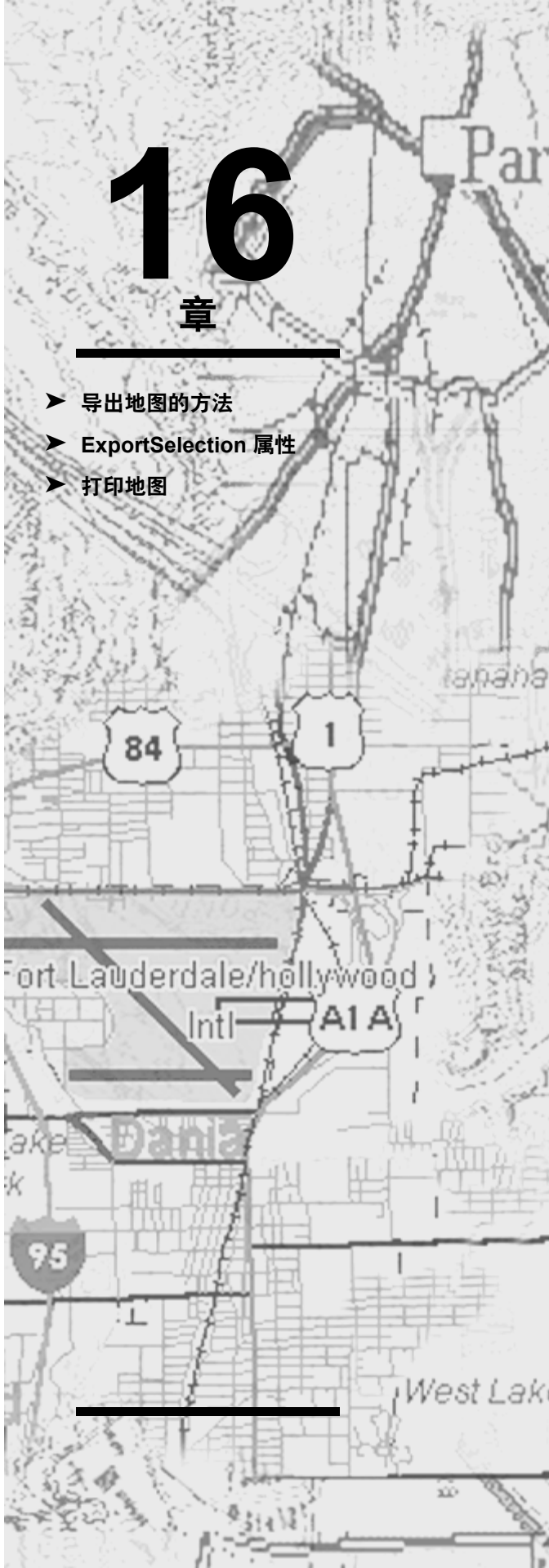
若要创建定制 polytool，应调用 CreateCustomTool 方法，并且指定 miToolTypePoly 的 ToolTypeConstants 值。

若要使您的定制 polytool 成为活动工具，应设置 CurrentTool 属性。

只要用户选择您的定制 polytool 并在地图上使用它， MapX 就调用 PolyToolUsed 事件。因此，您需要将代码添加到 PolyToolUsed 事件过程中，以使您的工具生效。

导出地图

用户常常需要打印地图或合成另一应用程序中的地图的可视图像。 MapX 提供专用的方法，允许您将地图内容发送到剪贴板、打印机或图形文件。



导出地图的方法

要将地图导出到图形文件或将图形内容复制到剪贴板，可使用 Map.ExportMap 方法。

要正确导出标题，导出图像的宽度和高度之比必须和地图的宽 / 高比相同。要将地图导出为原始高 / 宽的一半，可使用以下代码 （Visual basic 示例）：

```
Map1.ExportMap "BMP" Map1.MapPaperWidth/2,map1.MapPaperHeight/2
```

方法	说明	代码示例
ExportMap (graphic file)	将地图导出到图形文件。	Map1.ExportMap “C:\Map.TIF”,miFormatTIF
ExportMap (clipboard)	将地图导出到剪贴板。	Map1.ExportMap “clipboard”

另请参阅

Map.ExportMap 方法，第 401 页

Map.FeatureFactory 属性，第 406 页

Map.MapPaperWidth 属性，第 406 页

导出格式常数

格式	说明
miFormatWMF	元文件
miFormatBMP	位图
miFormatJPEG	JPEG 图像
miFormatTIF	TIF
miFormatGIF	GIF 图像
miFormatPNG	便携网络图形
miFormatPSD	PhotoShop

```
' 此示例展示 Map.ExportMap 方法。它使用
' 方法来将地图作为 BMP 文件置于剪贴板中。
Map1.PaperUnit = miUnitCentimeter

' 将 12 × 9 cm 的地图导出到剪贴板，采用 BMP 格式
Map1.ExportMap "clipboard", miFormatBMP, 12, 9
```

ExportSelection 属性

利用地图的 ExportSelection 属性将允许您控制是否随地图导出选择模式。有关此属性的使用方法可参阅以下代码示例：

```
' 导出用于显示的 jpg 图像，并包含在图像中突出显示的选择
Map1.ExportSelection=True
Map1.ExportMap "c:\temp\map.jpg", miFormatJPEG
```

打印地图

要打印地图，可使用 PrintMap 方法。有关详细信息，请参阅第 412 页的 “Map.PrintMap 方法”

注：当前地图的绘制符合给定的矩形。在维持宽高比时可以获取最佳结果。此方法只打印 MapX 地图的内容，不显示任何图标、工具栏或菜单项。如果需要在输出上查看附加控件，可能需要打印包含 MapX 控件的表单。

方法	说明	代码示例
PrintMap	打印地图	Map1.PrintMap Printer.hDC, 0, 0, Map1.Width * 100, Map1.Height * 100

17 章

- MapX 客户安装
- 安装 MapX
- 空间服务器访问安装支持
- 安装光栅格式处理程序
- 安装地图和 Geoset
- 向 Windows 注册表添加注册表项
- 传入 MapX 许可证字符串中
- 重新通过 MrSID 分发您的 MapX 应用程序
- 重新通过 ECW 分发您的 MapX 应用程序

分发您的 MapX 应用程序

您从 MapInfo 购买的 MapX 产品是开发人员套件。它包含 MapX、示例程序、联机帮助系统、示例地图和 geoset、以及各种实用程序和其它支持文件。本章介绍如何向客户分发您的 MapX 应用程序。请注意，您必须如 MapInfo 标准许可证 / 开发环境中所述，为您随 MapX 功能分发的每一应用程序的每一用户购买用户许可证 （即客户授权）作为 MapInfo 许可证书的凭据。

MapX 客户安装

在您的 MapX 应用程序中提供 MapX 控件 （即，需要在您的应用程序内分发以便它们正确工作的 MapX 部分）对于没有经验的程序员而言可能十分困难，因为涉及大量相关文件及其复杂结构。因此，MapInfo Corporation 已着手分发过程的自动化。基本上，有两种通过您的 MapX 应用程序分发 MapX 控件的方法。包括：

- 使用在 MapX CD 上提供的 MapX Installer。这是最简单的分发方法， MapX 开发人员只需进行非常少的工作，因为某些过程和任务已自动化。
- 将 MapX 集成到您要分发的应用程序内。此分发方法需要大量的人力，并且需要大量而详细的定位，但向程序员提供最高的控制程度以及最可能的定制性能。

随安装程序自动安装的文件

运行 [CD\Install\ setup.exe] 将在默认的 MapX 安装目录中安装以下内容：
C:\Program Files\MapInfo\MapX 5.0

- MapX 控件 mapx50.dll
- 它的支持的 dll
- 光栅和网格 dll 以及处理程序
- 默认的数据集驱动程序

系统 dll 也将安装在适当的位置 （请参见第 223 页的 “安装 MapX”）。

您可以使用 setup.exe 安装随 MapX 一起提供的默认数据，方法是向其传递 “-data”命令行参数。

在运行时将 MapX 集成到应用程序中

在您向客户提供应用程序时，将需要安装：

- 您创建的软件
- MapX 控件 （通过 MapX Installer 或您定制的版本）
- 相关的支持文件
- 您的应用程序使用的地图和 geoset

理想上，您将 MapX 的安装并入您的安装程序中，以便您的最终用户将不需要执行单独的安装来安装 MapX。

有四个主要步骤来安装和分发 MapX 应用程序：

1. 将必需的文件安装到您的用户的计算机上。
2. 运行实用程序 （例如 regsvr32.exe）来在用户的系统上注册特定文件。
3. 向 Windows 注册表添加特定的注册表项
4. 在应用程序运行时传入 MapX 许可证字符串。

注：如果您在对话框中与 Visual C++、 VisualBasic、 Delphi 或 PowerBuilder 一起使用 MapX，则它们将负责使用正确的许可证字符串创建 MapX，因此您可以忽略此部分。如果您在 Visual C++ 中使用 CMapX::Create() 方法创建 MapX 控件，则必须采取这一额外步骤。

随 MapX 一起安装的文件可以分为以下几类：

必需的文件：

- MapX DLL 及其相关 DLL
- Windows 系统文件：包括字体、 MFC 和 OLE dll

可选文件：

- 数据集驱动程序 （用于数据绑定）
- 导入 / 导出光栅处理程序
- 地图和 Geoset

安装 MapX

如果您已在计算机上安装 MapX，则可以很容易大致了解 MapX 使用多少文件。 MapX v4.x 在路径 “\Program Files\Common Files\MapInfo Shared\MapX Common”下安装程序文件。默认情况下， MapX v5.0 在 “\Program Files\MapInfo\MapX 5.0”中存储所有必需文件。此目录应用于 MapX 必需的文件，不过您的应用程序所需的所有文件都应放置于对您的最终用户和特定的应用程序有意义的目录下。

注：MapX 使用某些不位于 MapX 程序目录中的文件。例如， MapX 使用 MFC 和 OLE dll，它们驻留在 Windows System 目录中。

我们还建议 MapX 程序目录是：“\Program Files\MapInfo\MapX 5.0”。

下表描述需要安装哪些文件以及应将这些文件安装到的位置。

文件	安装文件的位置	其它特殊要求
Mfc42.dll、 msvcp60.dll、 msvcrt.dll、 ALT.dll、 MFC42.dll、 MFC42U.dll （用于 unicode 平台）	\Windows\System	在替换这些文件前进行版本检查。如果替换文件期间可能正使用较旧的版本，在这种情况下将要求重新启动。
Olepro32.dll、 oleaut32.dll	\Windows\System	必须使用 regsvr32.exe 注册
Mapx50.dll	MapX 公共目录	必须使用 regsvr32.exe 注册。在注册前，确保安装 MapX 相关文件。
mdatasetint.tlb	MapX 公共目录	必须使用 regtyplib.exe 注册

文件	安装文件的位置	其它特殊要求
MapX 相关文件： ALLTYPE.DLL, AllTypeRes.dll, AppSelection.dll ChangeManager.dll, ColLookupSystem.dll, CommandProcessor.dll, CommandProcessorRes.dll, COORDSYS.DLL, CoordSysRes.dll, CustomProperties.dll, DAENGINE.DLL, DAEngineRes.dll, DBINFO.DLL, DBInfoRes.dll, DBLAYER.DLL, DBLayerRes.dll, DELPHIMM.DLL, ExprPacket.dll, ExprPacketCreator.dll, ExprPacketCreatorRes.dll, ExprPacketRes.dll, FcnInfoServer.dll, FcnInfoServerRes.dll, FIND.DLL, FINDRES.DLL, GEO.DLL, GeoDictionaryManager50.exe, GeoDictionaryManagerIntl50.dll, GeoObject.dll, GeoObjectProcess.dll, GeoObjectProcessRes.dll, GeoObjectRes.dll, GEORES.DLL, GEOSET.DLL, GeosetManager50.exe, GeosetManagerIntl50.dll, GeosetRes.dll, GRIDDLL.DLL, LEGEND.DLL, LegendRes.dll, libspw_mi.dll, MapBasicInternalFcn.dll, MapBasicInternalFcnRes.dll, MapBasicTranslator.dll, MapBasicTranslatorRes.dll, MAPINFOW.FNT, MAPINFOW.PRJ, MAPPER.DLL, MapperRes.dll, MAPX.ABB, MAPX.PEN, MAPX50.DLL, MapXADODS.dll, MapXDAODS.dll, MapXRDODS.dll, mapxstate.dll, mapxstateres.dll, mdatasetint.tlb, MIAPP.DLL, MIAPPRES.DLL, MIDLG50.DLL, MIDLIN50.DLL, MILEXER.DLL, MILexerRes.dll, MIMetadata.dll, MIOCI.DLL, MIODBC.DLL, MIRDB.DLL, MIRDBRES.DLL, mirdbspatial.dll, mirdbspatialRes.dll, MIWINDOW.DLL, MIWindowRes.dll, MOCIDataset50.dll, MODBCDataset.dll, MSafeArrayDataset.dll, MXINTL50.DLL, mxmldataset.dll, RASTER.DLL, RasterRes.dll, REGSVR32.EXE, RegTypLib.exe, STYLES.DLL, TextFileReader.dll, TextFileReaderRes.dll Thematics.dll, ThematicsRes.dll, TOKENS.DLL, TOOLS.DLL, UTILITY.DLL, UtilityRes.dll, WINMGR.DLL, WinMgrRes.dll, XMLUTIL.DLL, XMLUTILRES.DLL,	MapX 程序目录	

文件	安装文件的位置	其它特殊要求
MapX 字体文件： ARIAL.TTF、 MAPIS___.TTF、 MAPSYM.TTF、 TTMIAR__.TTF、 TTMICG__.TTF、 TTMIMI__.TTF、 TTMIOG__.TTF、 TTMIOS__.TTF、 TTMIRE__.TTF、 TTMITC__.TTF、 TTMIWE__.TTF	Windows\Fonts 目录	这些字体必须安装并向操作系统注册。有关详细信息，请参见下文。
Nadcon 支持文件： ALASKA.LAS, ALASKA.LOS, CONUS.LAS, CONUS.LOS, HAWAII.LAS, HAWAII.LOS, PRVI.LAS, PRVI.LOS, STGEORGE.LAS, STGEORGE.LOS, STLRNC.LAS, STLRNC.LOS, STPAUL.LAS, STPAUL.LOS	MapX 程序目录	只在将图层与 NAD 27 和 NAD 83 坐标系统一起使用时才是必需的。
位图符号： TOWE1-32.BMP, POLI1-32.BMP, MBOX2-32.BMP, GOLF1-32.BMP, TOWE2-32.BMP, RAIL1-32.BMP, MOSQ1-32.BMP, HOSP1-32.BMP, IEL2-32.BMP, RAIL2-32.BMP, ONEW1-32.BMP, HOUS1-32.BMP, ADG1-32.BMP, ONEW2-32.BMP, HOUS2-32.BMP, ADG2-32.BMP, HOUS3-32.BMP, ANK1-32.BMP, ANK2-32.BMP, OOK1-32.BMP, AMP1-32.BMP, TRAF1-32.BMP, AR1-32.BMP, TRUC1-32.BMP, RAIL3-32.BMP, AUT1-32.BMP, TRUC2-32.BMP, REST1-32.BMP, PENC1-32.BMP, HUR1-32.BMP, YIEL1-32.BMP, STAT1-32.BMP, PIN1-32.BMP, HYDR1-32.BMP, AMBU1-32.BMP, STOP1-32.BMP,PIN2-32.BMP, INTE1-32.BMP, COMP1-32.BMP, SYNA1-32.BMP, PIN3-32.BMP, LITE1-32.BMP, FARM1-32.BMP, TARG1-32.BMP, PIN4-32.BMP, LITE2-32.BMP, FAST1-32.BMP, TAXI1-32.BMP, PIN5-32.BMP, MAIL1-32.BMP, FIRE1-32.BMP, TEMP1-32.BMP, PIN6-32.BMP, MBOX1-32.BMP, GLOB1-32.BMP,	在 MapX Program 目录下名为 Custsymb 的目录中。例如：\Program Files\MapInfo\MapX\Custsymb	只在使用定制位图符号时是必需的。

安装字体

MapX 也使用某些 TrueType 字体。这些字体文件 (*.ttf) 必须被复制到 Window 的 Font 文件夹中。

字体也必须向 Windows 注册。如果您正使用第三方软件包来创建您自己的安装程序，则该程序包也必须为您处理字体注册。不过，通过调用 Win32 AddFontResource 例程，您可以手动注册字体。一旦正确注册一种字体后，注册表就将在以下目录下包含相应的注册表项：

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Fonts

- 或者 -

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Fonts

空间服务器访问安装支持

如果在支持空间的 RDBMS （例如 SpatialWare for Oracle、 Informix 或 DB2 或者 Oracle 8*i* Spatial）中访问数据，则需要下面的驱动程序。

空间服务器	数据集驱动程序	安装要求
SpatialWare MapInfo ODBC	Miodbc.dll、 mirdb.dll、mirdbspatialres.dll、mirdbspatial.dll	Odbc32.dll 必须存在于系统路径中
Oracle 8i Spatial	Mioci.dll、 mirdb.dll、mirdbspatialres.dll	Odbc32.dll 必须存在于系统路径中

注册文件并安装数据集驱动程序

如上表中所述，某些文件需要 “注册”。需要注册的大多数文件

可以通过运行 regsvr32.exe 实用程序进行注册：

- regsvr32 /s <mapx50.dll 的 filespec>
- regsvr32 /s <olepro32.dll 的 filespec>
- regsvr32 /s <oleaut32.dll 的 filespec>

如果您的应用程序使用 ODBC 数据驱动程序，则必须注册 MODBCDataset.dll。如果您的应用程序使用 Notes 数据驱动程序，则必须注册 MNotesDataset.dll 和 MMapXColumnInfo.dll。如果您的应用程序使用 ODBC 或 Notes 数据驱动程序，则您必须按如下所示注册类型库。

regtyplib <mdatasetint.tlb 的 filespec>

MapX 允许您将数据绑定到地图图层。有关详细信息，请参见第 85 页的第 6 章：“将数据加入地图”。

注：上述所有数据集驱动程序都应与 MapX 安装在同一目录中。

该表阐释用于数据绑定的不同选项以及安装要求。下面的所有文件都必须向 regsvr32.exe 注册。

数据集源类型	数据集驱动程序	安装要求
Lotus Notes	Mnotesdataset.dll	Nnotes.dll 必须安装在系统路径中
ODBC	MODBCDataset.dll	Odbc32.dll 必须安装在系统路径中
Delphi v3	MgenDSetDrvr.dll、 Dslibp.dll	Delphimm.dll 必须安装在系统路径中
Delphi v4	MgenDSetDrvr.dll、 Dslibp4.dll	Borlndmm.dll 必须位于在系统路径中。在 Dslibp4.dll 上运行 RegisterDS4.exe
Delphi v5	MgenDSetDrvr.dll、 Dslibp5.dll	Borlndmm.dll 必须位于系统路径中。在 Dslibp5.dll 上运行 RegisterDS5.exe
Delphi v6	MgenDSetDrvr.dll、 Dslibp6.dll	Borlndmm.dll 必须位于系统路径中。在 Dslibp6.dll 上运行 registerDS6.exe
安全数组	MsafeArrayDataset.dll	
ADO v2.0	MapXADODS.dll、 ATL.dll	必须安装 ADO
RDO v2.0	MapXRDODS.dll、 ATL.dll	必须安装 RDO
OCI	MOCIDataset50.dll	必须安装 Oracle 客户程序、 mioci.dll 和 mirdb.dll
DAO	MapXDAODS.dll	MDAC v2.5 或更高版本
XML	mxmldataset.dll	必须安装 msxml4

安装光栅格式处理程序

MapX 还使开发人员可以将光栅图像并入其应用程序中。 MapX 应用程序可以打开或写入各种光栅图像类型。在分发您的应用程序时必须包括适当的驱动程序。

MapX 可以使用多种不同的库中的一种来装载光栅图像。在 MapX 装载光栅图像时，它将搜索这些 DLL 并且询问该 DLL 是否可以读取给定的文件。一旦某一 DLL 回答 yes， MapX 就将知道哪一 DLL 将可为该文件执行格式处理。格式处理程序将命名为 “xxxxxxxx.RHx”。名称的主要部分将基于该格式。扩展名将始终以 RH 为起始，但可以以任何字母 (A-Z) 为结束。在搜索格式处理程序时， MapX 以 RHA 为起始，然后是 RHB，直到 RHZ。这使得 MapX 可以确定优先使用哪些处理程序。例如，需要在任何其它格式之前检查 SPOT 文件，因为只有它们才是可能与其它格式造成混乱的原始数据。 SPOT 处理程序的扩展名是 RHD。 Halo 格式处理程序 （标准内置处理程序）命名为 RHV。 Lead 工具命名为 RHX。

随 MapX 一起提供 EAD Technologies, Inc 所提供的 LEADTOOLS Win32 Pro 以及 Media Cybernetics 提供的 HALO 图像库。您可以使用上述工具之一在 MapX 中显示大多数光栅文件。 LeadTools 会在 MapX 中引用图像时将整个光栅图像装入内存中。这意味着，将需要更多的时间来装载图像，但平移和缩放的速度将更快。 HALO 只将需要显示的内容装入内存中，因此它会更快地装载图像，但平移和缩放速度将

放慢。默认情况下，将首先尝试 HALO。您可以通过以下两种方法之一更改此顺序：在分发您的应用程序时不包括 HALO 库，或者通过重命名 LeadTools 处理程序，将其从 .rhx 重命名为 .rhu 或字母 “x” 之前未使用的其它任何字母。还可能有一些上述库不支持的格式，因此将具有其自己的独立的库。

注：上述所有光栅处理程序都应与 MapX 安装在同一目录中。

下面是与 MapX 一起提供的处理程序表：

格式处理程序	格式处理程序	所需的支持的光栅类型和 DLL
所有光栅类型	MIRASTER.DLL （用于 unicode 的 MIRASTERU.DLL）	
Lead 工具	LTFIL12N.DLL, LTKRN12N.DLL, LTDIS12N.DLL, LEADTOOL.RHX	BMP: LFBMP12N.DLL JPG: LFCMP12N.DLL GIF: LFGIF12N.DLL J2K: LFJ2K12N.DLL PNG: LFPNG12N.DLL PSD: LFPSD12N.DLL TIF: LFTIF12N.DLL, LFFAX12N.DLL WMF: LFWMF12N.DLL
Halo 库	Halo.rhv, Mihiffl.dll	BMP: miffbmp.dll GIF: miffgif.dll JBPG: miffjpeg.dll PCX: miffpcx.dll TGA: mifftga.dll TIF: mifftiff.dll
TIF	TIFF.RHL	或者您可以使用 Lead 工具 或 Halo TIF 库
SPOT	Spot.rhd	BIL
ECW	ECW.RHL NCSCNET.DLL NCSECW.DLL NCSECWEX.DLL NCSUTIL.DLL	ECW
Northwood Grid （作为光栅）	MIGRID.DLL （MIGRIDU.DLL - 用于 Unicode） MIRASTER.DLL （MIRASTERU.DLL - 用于 Unicode）	GRD
MapInfo Grid	GRIDDLL.DLL、 MIG.GHL	MIG
政府	ADAGASRP.RHL	ADRG、 ASRP
政府	CADRGCIB.RHL	CADRG、 CIB、 NITF

格式处理程序	格式处理程序	所需的支持的光栅类型和 DLL
Northwood Grid （作为网格）	VMGRID.DLL、 NWGRD30.GHL	GRD
DTED	DTED.GHL	DT?
USGS Digital Elevation Grid	DEM.GHL	DEM
GTOPO Grid	GTOPO30.GHR	DEM

安装地图和 Geoset

注册 geoset 是一种将与该 geoset 相关联的每一 MapInfo 表注册到 GeoDictionary 中的便捷方法。通过在 GeoDictionary 中注册 MapInfo 表，该表可用于自动绑定。若要获得与自动绑定有关的更多信息，请参阅第 85 页的第 6 章：“将数据加入地图”。

如果确定您需要使用 GeoDictionary，则当您注册 geoset 时， GeoDictionaryManager50.exe 将适当的项添加到 Geodictionary (geodict.dct) 中。如果该 Geodictionary 不存在，则 GeoDictionaryManager50.exe 将创建一个。

此步骤必须在所有 Geoset 及其关联的表都安装在用户的计算机上之后执行。复查您的 geoset 以查看用于每一表的预期路径。例如，如果您使用 MapInfo 表在硬盘驱动器的不同区域上创建了 geoset，则该 geoset 将指定指向与该 geoset 不在同一目录中的任何内容的完整路径。如果重新分发此 geoset，则这些表将不得不位于与原始 geoset 相同的路径中。若要避免此问题，应在创建该 geoset 前将所有 MapInfo 表复制到同一目录中。然后在该同一目录中创建该 geoset。

向 Windows 注册表添加注册表项

MapX 还使用以下 5 个注册表项 （如果这些注册表项不存在，则您的安装程序必须在最终用户的系统上创建它们）。 MapX 安装程序在您在系统上安装 MapX 时创建这些注册表项。因此，如果您想要查看这些注册表项的大概情形，则使用 regedit.exe 之类的实用程序查看您的系统的注册表。

注册表项	说明
HKEY_LOCAL_MACHINE\Software_MapInfo\MapX 5.0\GeoDictionary	**字符串** - GeoDictionary 注册表项具有用于 geodictionary 文件的文件说明。示例：C:\Program Files\myappdir\Maps\GeoDict.DCT
HKEY_LOCAL_MACHINE\Software_MapInfo\MapX 5.0 \SearchPaths	**字符串** - SearchPaths 注册表项具有地图文件和 geoset 可以存在于其中的分号分隔的文件说明；它默认为空字符串 ("")，而不位于在 GeoDictionary 注册表项中指定的目录中。

注册表项	说明
HKEY_LOCAL_MACHINE\Software_ MapInfo\MapX 5.0\ CommonDLLDir	**字符串** - 具有 MapX 和支持的文件位于其中的文件夹的位置。示例：\Program Files\Common Files\MapInfo Shared\MapX Common
HKEY_LOCAL_MACHINE\Software\MapInfo_ MapX\5.0\Program Dir	**字符串** - MapX 控件和支持文件的位置。
HKEY_LOCAL_MACHINE\Version Code\	**字符串** - MapX 版本。

注：在将 geoset 添加到 MapX 对象时 （没有指定 Geoset 的完整路径）和在使用自动绑定时，使用 GeoDictionary 路径。此外，如果没有使用 GeoDictionary，则 SearchPaths 注册表项不是必需的。请参见 “安装地图和 Geoset”，可以确定这些注册表项是否是您的应用程序所必需的。

传入 MapX 许可证字符串中

为了在用户的计算机上实例化 MapX 的运行时版本时，必须在执行您的应用程序期间在对象创建时传递许可证字符串。

注：如果您在对话框中与 Visual C++、 VisualBasic、 Delphi 或 PowerBuilder 一起使用 MapX，则它们将负责使用正确的许可证字符串创建 MapX，因此您可以忽略此部分。如果您在 Visual C++ 中使用 CMapX::Create() 方法创建 MapX 控件，则必须采取这一额外步骤。

Visual C++

例如，如果在 Visual C++ 的对话框中使用 MapX 时，则不需要执行此步骤。

为了成功创建该 MapX 对象，您必须将许可证字符串作为 bstrLicKey 参数传递给 CMapx::Create()。

重新通过 MrSID 分发您的 MapX 应用程序

在许可证获得者想要部署支持 MrSID 文件格式 (*.sid) 的应用程序时，许可证获得者应包括以下文本：

MrSID 软件 （尤其是 mrsid32.dll）依据许可证使用，并且版权所有 © 1995-2002, LizardTech, Inc., 1008 Western Ave., Suite 200, Seattle, WA 98104。保留所有权利。 MrSID 受到美国专利号 5,710,835 的保护。外国专利权正在申请中。禁止未授权的使用或复制。

软件中获得专利权的技术是部分通过 Los Alamos National Laboratory 的项目开发的，该实验室由美国政府提供资金并由加利福尼亚州立大学进行管理。美国政府具有对该技术的保留权利，包括在全球范围内为美国政府或代表美国政府要实施或已实施的非专有、不可转让、不能取消、已付的许可证，该专利涵盖的发明，并且具有基于 35 U.S.C. ß 200-212 的其它权利以及适用的实行法规。

有关进一步的信息，请与 Lizardtech 联系。

重新通过 ECW 分发您的 MapX 应用程序

如果您需要部署支持 ECW 光栅文件格式 (*.ecw) 的应用程序，则必须包括以下版权信息：

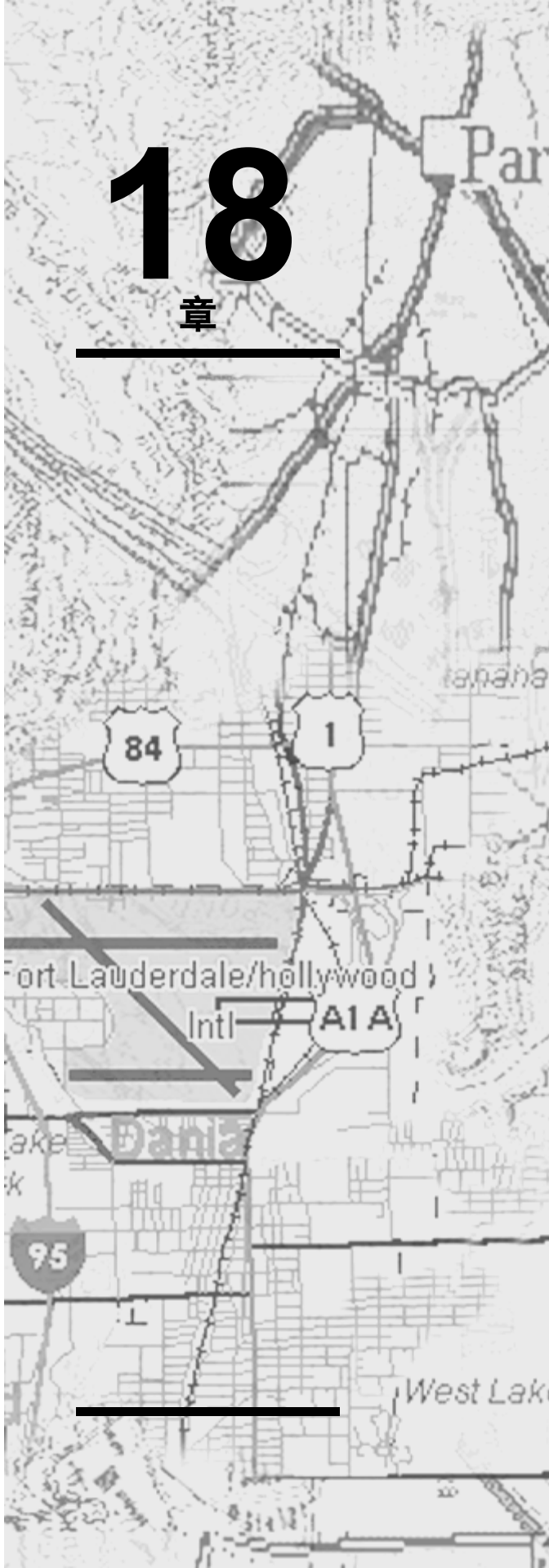
ECW for MapInfo 版权所有 MapImagery Limited。有关更多信息，请参见 MapImagery Web 站点 http://www.mapimagery.com。

ECW 压缩格式版权所有 Earth Resource Mapping。有关更多信息，请参见 Earth Resource Mapping 站点 http://www.ermapper.com。美国专利号 6,201,897 中的一个或多个涵盖 ECW 压缩技术

18 章

MapX 对象模型

本章是按照字母排序的参考，说明 MapInfo MapX 的对象、属性和方法。



Affine Transform 对象

AffineTransform 对象允许您定义旋转或扭曲的坐标系。

对象属性

- **AffineTransform.A, B, C, D, E, F 属性**

对象方法

- **AffineTransform.Set 方法**
- **AffineTransform.Units 方法**

注解

所有属性均为只读。要修改 **AffineTransform** 对象，可使用 Set 方法。

要获取 **AffineTransform** 对象，可参考 **CoordSys** 对象的 **AffineTransform** 属性，或声明新的独立 Affine Transform 对象。

另请参阅

有关仿射转换的介绍，请参阅第 177 页的第 13 章：“使用坐标系统”。

AffineTransform.A, B, C, D, E, F 属性

目的

这些属性均为只读，表示仿射转换的参数，返回双精度值。

注解

仿射转换具有以下形式：

$x' = Ax + By + C$

$y' = Dx + Ey + F$

在上述等式中，对基坐标 (x, y) 进行转换以得出**派生**的坐标 (x', y')。从 A 到 F 的 6 个常数确定该转换的结果，如下所示：

A	沿 X 轴执行缩放或伸展。
B	沿 X 轴执行旋转或扭曲。
C	沿 X 轴执行移位。
D	沿 Y 轴执行旋转或扭曲。
E	沿 Y 轴执行缩放或伸展。
F	沿 Y 轴执行移位。

这些属性均为只读，要设置这些属性，可使用 Set 方法。

AffineTransform.Set 方法

目的

此方法设置仿射转换的属性。

语法

OBJECT.**Set** UNITS [A, B, C, D, E, F]

部分	说明
OBJECT	AffineTransform 对象。
UNITS	**MapUnitConstants** 值，例如 miUnitMeter (7)。
A	双精度值，沿 X 轴执行缩放或伸展。
B	双精度值，沿 X 轴执行旋转或扭曲。
C	双精度值，沿 X 轴执行移位。
D	双精度值，沿 Y 轴执行缩放或伸展。
E	双精度值，沿 Y 轴执行旋转或扭曲。
F	双精度值，沿 Y 轴执行移位。

注解

所有参数均为必需。

仿射转换具有以下形式：

$$x' = Ax + By + C$$

$$y' = Dx + Ey + F$$

在上述等式中，对基坐标 (x, y) 进行转换以得出派生的坐标 (x', y')。

AffineTransform.Units 方法

目的

这是一个只读属性，指定仿射转换使用的地图单位。此函数返回匹配一个 **MapUnitConstants** 的短值。

AllFeaturesConstraint 对象

这一对象允许程序员指定远程图层中的所有图元应该存储在缓存中。通过调用 Layer.CreateAllFeaturesConstraint 方法创建了 AllFeaturesConstraint 对象。图层可以有多个与其关联的 AllFeaturesConstraint 对象。AllFeaturesConstraint 对象的 Enable 和 Disable 方法可在它创建之后用于添加或去除此约束。 Enabled 属性可用于确定当前的约束状态。

要强制刷新缓存，可使用 Layer.Refresh 方法。

对象属性

- **AllFeaturesConstraint.Enabled 属性**

对象方法

- **AllFeaturesConstraint.Disable 方法**
- **AllFeaturesConstraint.Enable 方法**

示例

- **Visual Basic 示例**

另请参阅

LayerInfo 对象，第 373 页

使用缓存，第 166 页

Layer.CreateAllFeaturesConstraint 方法，第 344 页

BoundsConstraint 对象，第 249 页

FeaturesConstraint 对象，第 285 页

AllFeaturesConstraint.Disable 方法

目的

在 AllFeaturesConstraint 对象创建之后，此方法和 Enable 方法一起用于控制是否使用由对象定义的约束。禁用之后， AllFeaturesConstraint 对象不用于确定将哪些记录存储在缓存中。启用之后，AllFeaturesConstraint 对象用于确定将哪些记录存储在缓存中。与其他约束对象不同，在初始化之后，AllFeaturesConstraint 对象将在默认情况下禁用。

语法

OBJECT.**Disable**

OBJECT － AllFeaturesConstraint 对象

另请参阅

使用缓存，第 166 页

AllFeaturesConstraint.Enabled 属性，第 237 页

AllFeaturesConstraint.Enable 方法，第 237 页

AllFeaturesConstraint.Enable 方法

目的

在 AllFeaturesConstraint 对象创建之后，此方法和 Disable 方法一起用于控制是否使用由对象定义的约束。在禁用的 AllFeaturesConstraint 对象不用于确定将哪些记录存储在缓存中。在启用的 AllFeaturesConstraint 对象用于确定将哪些记录存储在缓存中。在初始化之后， AllFeaturesConstraint 对象将在默认情况下禁用，因此，在最初使用该约束时必须使用此方法。

语法

OBJECT.**Enable**

OBJECT － AllFeaturesConstraint 对象

另请参阅

使用缓存，第 166 页

AllFeaturesConstraint.Enabled 属性，第 237 页

AllFeaturesConstraint.Disable 方法，第 236 页

AllFeaturesConstraint.Enabled 属性

目的

这是一个布尔值，只读属性。在为 TRUE 时， AllFeaturesConstraint 对象用于确定将哪些地图图元存储在缓存中。在为 FALSE 时，将不会使用 AllFeaturesConstraint 对象。

另请参阅

使用缓存，第 166 页

AllFeaturesConstraint.Enable 方法，第 237 页

AllFeaturesConstraint.Disable 方法，第 236 页

Annotation 对象和 Annotations 集合

每个地图都有一个 Annotations (Map.Annotations property) 集合。注释是符号或文本对象，并且绘制在地图顶层。

注释通常用于向地图添加消息 （文本），或在地图上放置符号。这些注释随地图的大小缩放。它们不一定要与地图的特定图层关联，并且始终位于顶部。

注意 Annotation 对象没有用于设置位置、符号样式或文本的属性。要控制这些注释的特性，您必须使用 Annotation.Graphic 属性来获取 **Graphic 对象**，然后进行修改。

对象属性

- **Annotation.Graphic 属性**
- **Annotation.Type 属性**

集合属性

- **Annotations.Count 属性**
- **Annotations.Editable 属性**
- **Annotations.Item 属性**

集合方法

- **Annotations.ActiveAnnotation 方法**
- **Annotations.AddSymbol 方法**
- **Annotations.AddText 方法**
- **Annotations.Remove 方法**
- **Annotations.RemoveAll 方法**

另请参阅

Map 对象，第 388 页

Annotation.Graphic 属性

目的

此属性包含 Graphic 对象，具有用于 Annotation 的属性。请参阅 Graphic 对象说明。

另请参阅

Graphic 对象，第 324 页

Annotation.Type 属性

目的

此属性指定注释类型。这是 **AnnotationTypeConstants** 的值，可以是 miSymbolAnnotation 或 miTextAnnotation。这是一个只读属性。

Annotations.ActiveAnnotation 方法

目的

这一方法将返回激活 （当前所选）的注释。

语法

OBJECT.**ActiveAnnotation**

Annotations.AddSymbol 方法

目的

此方法向 Annotations 集合添加符号。使用默认样式 （正如在 **Map.DefaultStyle** 指定）。

语法

Annotations.AddSymbol [X, Y]

部分	说明
X	要添加的符号的 X 坐标。这是一个双精度值，用于指定地图坐标 （经度）。
Y	要添加的符号的 Y 坐标。这是一个双精度值，用于指定地图坐标 （纬度）。

另请参阅

Map.DefaultStyle 属性，第 399 页

Graphic.X 属性，第 325 页

Graphic.Y 属性，第 325 页

Annotations.AddText 方法

目的

此方法向 Annotations 集合添加文本注释。使用默认样式 （正如在 Map.DefaultStyle 指定）。

语法

OBJECT.**AddText** (Text, X, Y, [Position])

部分	说明
OBJECT	表示 Annotations 对象的占位符。
文本	字符串值。成为新注释的说明。
X	放置文本的 X 坐标。这是一个双精度值，指定地图坐标 （经度）。
Y	放置文本的 Y 坐标。这是一个双精度值，指定地图坐标 （纬度）。
Position	和 x,y 坐标有关的添加文本参考点。取值为 **PositionConstants** 值。

另请参阅

Map.DefaultStyle 属性，第 399 页

Graphic.Caption 属性，第 324 页

Graphic.Position 属性，第 324 页

Graphic.X 属性，第 325 页

Graphic.Y 属性，第 325 页

Annotations.Count 属性

目的

此属性指定集合中的 **Annotation** 对象数量。该属性是一个整数值，是只读的。

Annotations.Editable 属性

目的

此属性控制注释是否可以最终用户编辑。可编辑注释可以由最终用户移动、重新调整大小并可以在当前位置编辑。该属性是一个布尔值，默认值为 True。

Annotations.Item 属性

目的

此属性从集合获取 Annotation 对象。索引用于指定要获取哪一 Annotation。该索引是从 1 到 Annotations.Count 的整数值。这是用于 Annotations 集合的默认属性。

另请参阅

Annotations.Count 属性，第 240 页

Annotations.Remove 方法

目的

此方法从集合去除 Annotation 对象。

注：如果您去除项目，集合索引将再次计数，填充由已删除项目留下的空洞。

语法

OBJECT.**Remove** (Index)

部分	说明
OBJECT	表示 Annotations 对象的占位符。
Index	要使用的注释的索引。这是一个整数，介于 1 和 AnnotationCount 之间。

另请参阅

Annotations.Count 属性，第 240 页

Annotations.RemoveAll 方法

目的

从集合中删除所有注释对象。

语法

OBJECT.**RemoveAll**

部分	说明
OBJECT	表示 Annotation 对象的占位符。

BindLayer 对象

BindLayer 对象用作 Datasets.Add 方法的参数。用于指定如何执行数据绑定的信息。在绑定带有 X 和 Y 坐标的数据或类似邮政编码的数据时，以及需要在新的图层中于这些位置创建的点时，这是必需的。

对象属性

- **BindLayer.CoordSys 属性**
- **BindLayer.Filespec 属性**
- **BindLayer.KeyLength 属性**
- **BindLayer.LayerName 属性**
- **BindLayer.LayerType 属性**
- **BindLayer.OverwriteFile 属性**
- **BindLayer.RefColumn1 属性**
- **BindLayer.RefColumn2 属性**
- **BindLayer.ReferenceLayer 属性**
- **BindLayer.ReferenceLayerField 属性**

另请参阅

Datasets.Add 方法，第 263 页

BindLayer.CoordSys 属性

目的

此属性使用 CoordSys 对象控制您创建图层时使用的坐标系统。

注解

在您执行 x/y 或将点参考数据绑定来创建点图层时， BindLayer.CoordSys 属性指定将要用于创建图层的坐标系。此外，对于 x/y 数据绑定， MapX 将使用此坐标系来解释数据源的 x 和 y 坐标。例如，如果您知道数据源包括州平面坐标中的 x 和 y 坐标，可将 BindLayer.CoordSys 设置为使用州平面坐标系。

如果没有设置此属性，则当前 Map.NumericCoordSys 属性用作坐标系。

BindLayer.Filespec 属性

目的

此属性允许指定文件的名称和位置，使 Datasets.Add 方法可以创建永久图层而不是临时图层。它是一个字符串属性，应该设置为要创建的图层的文件路径。

注解

此属性应用于 x/y 数据绑定和 pointref 数据绑定（例如 BindLayer.LayerType 属性是 miBindLayerTypeXY 或 miBindLayerTypePointRef）。

如果您将文件路径和名称分配给此属性，则 Datasets.Add 方法将创建一个永久层，存储于指定的路径中。如果未指定该属性，图层则为临时图层。

BindLayer.KeyLength 属性

目的

这是一个整数属性，指定由 Datasets.Add 生成图层中的列的字符长度。

注解

如果您没有为此属性分配值，则默认的关键字长度为 254。换而言之，生成的图层将包含一个 254 个字符宽的列。

BindLayer.LayerName 属性

目的

如果 BindLayer.LayerType 是 miBindLayerTypeNormal，或新建图层的名称，如果 BindLayer.LayerType 是 miBindLayerTypeXY 或 miBindLayerTypePointRef，则此属性指定要将数据绑定到的图层的名称。这是字符串值。

另请参阅

BindLayerTypeConstants，第 559 页

BindLayer.ReferenceLayer 属性，第 245 页

BindLayer.LayerType 属性，第 243 页

BindLayer.RefColumn1 属性，第 244 页

BindLayer.RefColumn2 属性，第 245 页

BindLayer.LayerType 属性

目的

此属性指定数据绑定到的图层类型。绑定数据是指创建 Dataset 的过程，该过程是使用 Datasets.Add 方法创建的。此属性取 BindLayerTypeConstant 的值，如下所述。

讨论

如果设置为 miBindLayerTypeNormal，则表示您正将数据绑定 （添加）到 GeoDictionary 中的图层。

如果设置为 miBindLayerTypeXY，则表示为数据中具有的 X 和 Y 坐标，并且您希望要创建新的点图层。在使用时，您需要指定新建点图层的名称 （BindLayer.LayerName），数据中的字段包含 X 坐标 (BindLayer.RefColumn1)，并且字段包含 Y 坐标 (BindLayer.RefColumn2)。

如果设置为 miBindLayerTypeXY，则表示为数据中具有的 X 和 Y 坐标，并且您希望要创建新的点图层。新建图层对于源数据中各行均包含一个点，匹配点参考文件中的条目。例如，如果源数据包括邮政编码信息，并且您有一个点参考文件，其中包含安装在 GeoDictionary 中的美国邮政编码中心，您可以创建新的图层，对于每个包含在源数据中的邮政编码，包含一个点 （如果邮政编码出现多次，则为多个点）。在使用时，您需要指定新建点图层的名称 （BindLayer.LayerName），数据中的字段包含参考数据 (BindLayer.RefColumn1)，并且点参考文件的名称 – 包含上例中的邮政编码位置的文件 (BindLayer.ReferenceLayer)。点参考文件必须安装在 GeoDictionary 之中。

另请参阅

Datasets.Add 方法，第 263 页

BindLayerTypeConstants，第 559 页

BindLayer.ReferenceLayer 属性，第 245 页

BindLayer.LayerType 属性，第 243 页

BindLayer.RefColumn1 属性，第 244 页

BindLayer.RefColumn2 属性，第 245 页

BindLayer.OverwriteFile 属性

目的

此属性允许用户指定 MapX 是否覆盖由 Bindlayer.Filespec 属性指定的文件 （如已存在）。该属性是一个布尔值属性，默认值为 False。

注解

此属性应用于 X/Y 和点参考数据绑定 （例如 BindLayer.LayerType 属性是 miBindLayerTypeXY 或 miBindLayerTypePointRef）。

此属性和 BindLayer.FileSpec 属性一起使用，并且由 Datasets.Add 方法使用，创建永久图层，存储在指定路径中。如果此属性设置为 false，但由 BindLayer.Filespec 属性指定的文件确实存在，出现 #1230 错误。

另请参阅

BindLayer.LayerType 属性，第 243 页

BindLayer.Filespec 属性，第 242 页

Datasets.Add 方法，第 263 页

BindLayer.RefColumn1 属性

目的

此属性指定字段名或字段编号 （从 1 开始）中的一个：

- BindLayer.LayerType 为 miBindLayerTypeXY 时字段包含 X 值 （经度）。
- BindLayer.LayerType 为 miBindLayerTypeRef 时字段包含参考值 （例如邮政编码）。

另请参阅

BindLayerTypeConstants，第 559 页

BindLayer.RefColumn2 属性

目的

BindLayer.LayerType 为 miBindLayerTypeXY 时，此属性指定包含 Y 值 （纬度）字段的字段名或字段编号 （从 1 开始）。

另请参阅

BindLayer.LayerType 属性，第 243 页

BindLayerTypeConstants，第 559 页

BindLayer.ReferenceLayer 属性

目的

此属性指定要使用的参考文件的名称 （如果 BindLayer.LayerType 为 miBindLayerTypePointRef）。在进行图层绑定是将使用参考图层。这将指定要使用的参考文件的名称。包含项目 （例如邮政编码）的 x 和 y 坐标的参考文件。 MapX 附带 “美国 5 位邮政编码中心”图层，可用于邮政编码层级上的点参考绑定。

ReferenceLayer 属性可以设置为图层的文件名 （完整文件名说明）或设置为图层的 “友好名称”（通过 Geodictionary 为图层分配的说明）。

PointReference 绑定 （miBindLayerTypePointRef 类型的数据绑定），可以使用任意表作为参考图层，而不只是 CPF 文件。在进行 PointReference 绑定时，指定参考图层作为提供给 Datasets.Add 方法的 BindLayer 对象的 ReferenceLayer 属性。

另请参阅

BindLayer.LayerType 属性，第 243 页

BindLayerTypeConstants，第 559 页

BindLayer.ReferenceLayerField 属性

目的

指定要绑定到 MapInfo 的表中的字段 （按照数字，从 1 开始）。如未指定， MapX 将采集用于列的源数据，选择匹配率最高的列。如果没有匹配，则选择第一个列。在使用指定字段绑定之前必须检索字段。

注解

此属性可选 （不需要设置）。它只对 BindLayerTypeNormal 有效，并且只在要绑定到的图层在 LayerName 属性中指定时有效。

BitmapSymbol 对象 和 BitmapSymbols 集合

BitmapSymbol 对象代表在用户系统上可用的位图符号之一。 BitmapSymbol 集合表示在用户系统上可用的所有位图符号的集合。

注解

符号位图的大小从 1 至 48 点。如果 BitmapSymbol 的大小小于 1 点，则 Size 属性设置为 1。如果 BitmapSymbols 的大小大于 48 点，则 Size 属性将设置为 48。

请注意可以支持 2、 16 和 256 位颜色位图，但是不支持 24 位颜色。

注：BitmapSymbols.Item 是 BitmapSymbols 集合的默认属性。

对象属性

- **BitmapSymbol.Name 属性**

集合属性

- **BitmapSymbols.Count 属性**
- **BitmapSymbols.Item 属性**

集合方法

- **BitmapSymbols.Refresh 方法**
- **BitmapSymbols.Unload 方法**

BitmapSymbol.Name 属性

目的

这一只读字符串表示文件名 （例如 filename.bmp），表示位图符号。

注解

Name 属性为只读属性。使用此属性来确定位图符号文件 (*.bmp) 的名称，您可以将该名称分配给 Style 对象的 SymbolBitmapName 属性。

BitmapSymbols.Count 属性

目的

这一只读属性返回集合中的项目数量 （即可用作位图符号的位图的数量）。

注：请注意可以支持 2、 16 和 256 位颜色位图，但是不支持 24 位颜色。

BitmapSymbols.Item 属性

目的

这一只读属性返回集合中的 BitmapSymbol 对象，只使用数字索引。这是 BitmapSymbols 集合的默认属性。

语法

[BitmapSymbol=]OBJECT.**Item** (index)

部分	说明
OBJECT	表示 BitmapSymbol 对象。
index	表示要返回的 BitmapSymbol 对象的整数。

BitmapSymbols.Refresh 方法

目的

这将更新 **BitmapSymbols** 集合，如有必要确保该集合处于当前状态。

注解

BitmapSymbols 集合表示在 CUSTSYMB 目录中可见的 .bmp 文件的集合。但是，如果在 MapX 运行时添加 .bmp 文件，MapX 不会自动更新集合。要确保 **BitmapSymbols** 集合处于当前状态，可使用 Refresh 方法。

语法

OBJECT.**Refresh** ()

BitmapSymbols.Unload 方法

目的

卸载此前加载的所有 .bmp 文件，释放关联的资源。

注解

当 **BitmapSymbols** 集合消失时，资源将不会自动释放。当您使用 BitmapSymbols 集合时，使用 Unload 方法来显式释放关联的资源。

当您使用此方法来显式释放集合的资源，则资源将在 MapX 库卸载时释放。

语法

OBJECT.**Unload** ()

BoundsConstraint 对象

这一对象允许程序员通过指定矩形的边界，指定将哪些图元缓存用于远程数据库图层。任意用于图元的 MinimumBoundingRectangle (MBR) 重叠矩形 MBR 的图元，用于定义此约束，将保存在缓存中。通过调用 Layer.CreateBoundsConstraint 方法创建了 BoundsConstraint 对象。图层可以有多个与其关联的 BoundsConstraint 对象。 BoundsConstraint 对象的 Enable 和 Disable 方法可在它创建之后用于添加或去除此约束。Enabled 属性可用于确定当前的约束状态。Bounds 属性可用于设置或检查定义此约束的 MBR。

要强制刷新缓存，可使用 Layer.Refresh 方法。

对象属性

- **BoundsConstraint.Bounds 属性**
- **BoundsConstraint.Enabled 属性**

对象方法

- **BoundsConstraint.Disabel 方法**
- **BoundsConstraint.Enable 方法**

示例

- **Visual Basic 示例**

另请参阅

LayerInfo 对象，第 373 页

使用缓存，第 166 页

Layer.CreateBoundsConstraint 方法，第 345 页

Rectangle 对象，第 437 页

FeaturesConstraint 对象，第 285 页

AllFeaturesConstraint 对象，第 236 页

BoundsConstraint.Bounds 属性

目的

这是读 / 写属性，可以取或返回 MapX 矩形对象。矩形定义 BoundsConstraint 对象的地理范围，确定哪些图元存储在缓存中。任意用于图元的 MinimumBoundingRectangle (MBR) 重叠矩形 MBR 的图元，用于定义此约束，将保存在缓存中。通过更改 Bounds 属性，您可以潜在更改存储在缓存中的图元。

语法

[Rectangle=] OBJECT.**Bounds**

[Rectangle=] － A returned MapX Rectangle object

OBJECT － BoundsConstraint 对象

另请参阅

Rectangle 对象，第 437 页

使用缓存，第 166 页

BoundsConstraint.Disabel 方法

目的

在 BoundsConstraint 对象创建之后，此方法和 BoundsConstraint.Enable 方法一起用于控制是否使用由对象定义的约束。在禁用的 BoundsConstraint 对象不用于确定将哪些记录存储在缓存中。在启用的 BoundsConstraint 对象用于确定将哪些记录存储在缓存中。初始化之后， BoundsConstraint 对象将在默认情况下启用。

语法

OBJECT.**Disable**

OBJECT － BoundsConstraint 对象

另请参阅

使用缓存，第 166 页

BoundsConstraint.Enabled 属性，第 251 页

BoundsConstraint.Enable 方法，第 250 页

BoundsConstraint.Disabel 方法，第 250 页

BoundsConstraint.Enable 方法

目的

在 BoundsConstraint 对象创建之后，此方法和 Disable 方法一起用于控制是否使用由对象定义的约束。在禁用的 BoundsConstraint 对象不用于确定将哪些记录存储在缓存中。在启用的 BoundsConstraint 对象用于确定将哪些记录存储在缓存中。在初始化之后， BoundsConstraint 对象将在默认情况下启用，此方法只需在 BoundsConstraint 通过 BoundsConstraint.Disable 方法禁用之后。

语法

OBJECT.**Enable**

OBJECT － BoundsConstraint 对象

另请参阅

使用缓存，第 166 页

BoundsConstraint 对象，第 249 页

BoundsConstraint.Enabled 属性，第 251 页

BoundsConstraint.Disabel 方法，第 250 页

BoundsConstraint.Enabled 属性

目的

这是一个布尔值，只读属性。在为 TRUE 是， BoundsConstraint 对象用于确定将哪些地图图元存储在缓存中。在为 FALSE 时，将不会使用 BoundsConstraint 对象。

另请参阅

使用缓存，第 166 页

BoundsConstraint.Enable 方法，第 250 页

BoundsConstraint.Disabel 方法，第 250 页

CoordSys 对象

CoordSys 对象表示坐标系。

对象属性

- **CoordSys.AffineTransform 属性**
- **CoordSys.Azimuth 属性**
- **CoordSys.Bounds 属性**
- **CoordSys.Datum 属性**
- **CoordSys.FalseEasting 和 CoordSys.FalseNorthing 属性**
- **CoordSys.OriginLatitude、 OriginLongitude 属性**
- **CoordSys.Range 属性**
- **CoordSys.ScaleFactor 属性**
- **CoordSys.StandardParallelOne、 StandardParallelTwo 属性**
- **CoordSys.Type 属性**
- **CoordSys.Units 属性**

对象方法

- **CoordSys.Clone 方法**
- **CoordSys.PickCoordSys 方法**
- **CoordSys.Set 方法**

注解

所有属性均为只读。要修改坐标系，可使用 **Set** 方法或 **PickCoordSys** 方法，或将其分配为等于另一个 CoordSys 对象。

要获取 **CoordSys** 对象，可参考 **Map.DisplayCoordSys** 属性、 **Map.NumericCoordSys** 属性或 **Layer.CoordSys** 属性。

另请参阅

第 177 页的第 13 章：“使用坐标系统”

Map.DisplayCoordSys 属性，第 399 页

Map.NumericCoordSys 属性，第 411 页

Layer.CoordSys 属性，第 346 页

CoordSys.AffineTransform 属性

目的

这是一个只读属性，返回 AffineTransform 对象。

另请参阅

Affine Transform 对象，第 234 页

CoordSys.Azimuth 属性

目的

这是只读属性，返回表示坐标系方位角的双精度值。

CoordSys.Bounds 属性

目的

这是只读属性，返回表示坐标系边界的矩形对象。

CoordSys.Clone 方法

目的

将坐标系复制为新的 CoordSys 对象。

语法

[CoordSys=]OBJECT.**Clone** ()

CoordSys.Datum 属性

目的

这是一个只读属性，返回 **Datum** 对象。

另请参阅

Datum 对象，第 267 页

CoordSys.FalseEasting 和 CoordSys.FalseNorthing 属性

目的

这是只读属性，返回表示坐标系的错误东向 / 错误北向设置的双精度值。

注解

此单位由 **CoordSys.Unit** 属性决定。

CoordSys.OriginLatitude、 OriginLongitude 属性

目的

这是只读属性，返回作为双精度值的坐标系纬度原点 / 经度原点设置。

CoordSys.PickCoordSys 方法

目的

这将显示一个对话框，允许用户选择坐标系。

语法

[Boolean=]OBJECT.**PickCoordSys** ([HelpFile], [ID])

部分	说明
OBJECT	CoordSys 对象
HelpFile	**Variant**：标识 Windows 帮助文件位置和文件名的字符串。可选，如果忽略，则对话框出现时将不会有 “帮助”按钮。
HelpID	**Variant**：整数 ID 号，标识要显示的帮助主题。可选，对话框出现时将不会有“帮助”按钮。

注解

该对话框自动初始化，这样突出显示的说明将匹配 CoordSys 对象。

如果用户在对话框中单击 “确定”，则该方法返回 True，并且更新 CoordSys 对象以匹配用户所选的坐标系统。如果用户单击 “取消”，则方法返回 False，并且 CoordSys 对象保持不变。

通过读取文件 mapinfow.prj 的内容， MapX 初始化可用坐标系列表，该文件位于和 mapx50.dll 相同的目录中。

CoordSys.Range 属性

目的

这是只读属性，返回表示坐标系范围的双精度值 （从 1 到 180 的数字，标识地球在多大程度上可见）。

CoordSys.ScaleFactor 属性

目的

这是只读属性，返回表示坐标系比例因子的双精度值。

CoordSys.Set 方法

目的

此方法设置 CoordSys 对象的属性。

语法

OBJECT.**Set (**Type, [Datum], [Units], [OriginLongitude], [OriginLatitude],

[StandardParallelOne], [StandardParallelTwo], [Azimuth], [ScaleFactor],

[FalseEasting], [FalseNorthing], [Range], [Bounds], [AffineTransform])

部分	说明
OBJECT	CoordSys 对象
类型	**CoordSysTypeConstants** 短精度值，例如 miRobinson (12)。
数据	**Datum** 对象或支持的数据标号 （例如 62 代表美国大陆的 “NAD 27” 数据）。
Units	**MapUnitConstants** 值，例如 miUnitMeter (7)。
OriginLongitude OriginLatitude	双精度值，表示标准平行 （纬度）。
StandardParallelOne StandardParallelTwo	双精度值，表示标准平行 （纬度）。
Azimuth	双精度值，表示方位角。
ScaleFactor	双精度值，表示比例因子。
FalseEasting FalseNorthing	双精度值，表示 Falseeasting 和 FalseNorthing （采用 Units 参数指定的单位）。
Range	双精度值：从 1 至 180 度的纬度，表示范围 （表示地球的可见范围）。
Bounds	一个 Rectangle 对象，表示坐标系的外部边界，采用 Units 参数指定的单位。如果类型为 miNonEarth 则为必需，否则为可选。
AffineTransformation	AffineTransform 对象，此参数始终为可选。

注解

Type 参数始终为必需。所有其他参数为变量，这些参数为必需与否取决于所选投影 / 坐标系。

如果参数不是必需，则可以传递任何值，或将其完全忽略。

CoordSys.StandardParallelOne、 StandardParallelTwo 属性

目的

这是只读属性，返回双精度值的纬度值，表示坐标系标准平行设置。

CoordSys.Type 属性

目的

这是只读属性，返回匹配 **CoordSysTypeConstants** 之一的短精度值。

CoordSys.Units 属性

目的

这是只读属性，返回匹配 **MapUnitConstants** 之一的短精度值。

Dataset 对象和 Datasets 集合

每个地图都具有一个 Datasets 集合。 Datasets 集合具有用于从集合添加或移除 Dataset 对象的方法和属性。数据绑定的概览如下。

数据绑定

数据绑定是将数据源中的数据导入 MapX 的过程。数据源可以是 Visual Basic 数据控件或 ODBC 数据源。在 MapX 中，数据显示为 Dataset 对象。

数据绑定可以通过以下两种方法完成。如果使用 Visual Basic 开发绑定数据控件，在设计阶段，可以使用 Map 对象的 Dataset 属性。或者，使用 Datasets.Add 方法通过程序绑定数据。使用该方法需要通知 MapX 要使用的数据源、数据源的某些信息以及要绑定到的地图图层。

绑定过程会创建 Dataset 对象。该 Dataset 加入 Datasets 集合，包含数据绑定到的地图图层中的图元的计算值。例如，如果数据绑定到 US_States 地图上，每个州将有一个新数据值，用于控制州的绘制方式。如果数据源中包含某个州的多个记录，值将求和、求平均值或计数。 Dataset 具有 Value 方法，可以使用该属性访问地图每一行 （或图元）的计算值。

DataSource （Datasets.Add 的第二个参数）实际上是 OLE 接口。 MapX 使用该接口直接从数据源中访问数据。数据实际上不传递给 Datasets.Add。

有关由 MapX 支持的数据类型的详细信息，请参见第 85 页的第 6 章：“将数据加入地图”。

对象属性

- **Dataset.Fields 属性**
- **Dataset.GeoField 属性**
- **Dataset.Layer 属性**
- **Dataset.Name 属性**
- **Dataset.ReadOnly 属性**
- **Dataset.RowCount 属性**
- **Dataset.SecondaryGeoField 属性**
- **Dataset.SourceRows 属性**
- **Dataset.Themes 属性**
- **Dataset.Type 属性**

对象方法

- **Dataset.AddField 方法**
- **Dataset.Refresh 方法**
- **Dataset.RowValues 方法**
- **Dataset.Value 方法**

集合属性

- **Datasets.BuildSourceRows 属性**
- **Datasets.Count 属性**

- **Datasets.Item 属性**

集合方法

- **Datasets.Add 方法**
- **Datasets.Contains 方法**
- **Datasets.Remove 方法**
- **Datasets.RemoveAll 方法**
- **Datasets.Restore 方法**

Dataset.AddField 方法

目的

这将允许将字段 （列）添加到 Dataset，该 Dataset 是包含函数、运算符和 Datasets 字段的表达式 （只从当前 Dataset）。这将向用于标记、主题等的 Dataset 字段添加新字段。

语法

OBJECT.**AddField** (Name, Expression)

部分	说明
OBJECT	表示 **Dataset** 对象。
Name	新字段的名称。
Expression	已创建的表达式。

注解

新字段的类型或者为 miTypeNumeric 或者为 miTypeString，具体取决于表达式的返回类型。尽管聚集类型设置为 miAggregationSum，但新字段将不会保存任何字段的聚集值。

对于数字字段，新字段的聚集类型为 miAggregationSum ；对于字符串字段，新字段的聚集类型为 miAggregationIndividual。

字段添加到 Dataset 的字段集合的末尾，并且将是临时字段。即在应用终止时，字段将会消失。此外，添加的字段不可更新。如果要尝试更新添加的字段，将不会出现异常错误。

对于无效的表达式、无效的名称参数或非唯一的字段名，将会出现异常错误。

另请参阅

请参阅第 575 页的 “创建表达式”

MapX 字段命名规范，第 507 页

Datasets.BuildSourceRows 属性

目的

Datasets 创建自 Datasets 集合，将此属性设置为 TRUE，将支持 Dataset.SourceRows 方法。相应 Datasets 创建自 Datasets 集合，将此属性设置为 FALSE，取消相应的对象上的 Datasets.SourceRows 将导致出现异常错误。

注解

给定 Dataset 在其生存期内或者支持或者不支持 Datasetset.SourceRows 方法。更改 Datasets 集合的属性不会影响此前从该 Dataset 集合创建的 Datasets 对象。

Dataset.Fields 属性

目的

用于 Dataset 的 Fields 集合

另请参阅

Field 对象和 Fields 集合，第 302 页

Dataset.GeoField 属性

目的

Fields 集合中的地理关键字列的列名。这是一个整数值，是只读属性。

Dataset.Layer 属性

目的

此 Dataset 绑定到的 Layer 对象。

注解

添加 Dataset 之后，数据将始终附属到图层。这将指向该图层。

一个图层可以有多个绑定到其的 Dataset 对象。删除图层之后，绑定到其的 Datasets 也将被删除。

Dataset.Name 属性

目的

Dataset 的名称。这是字符串值。

注解

该名称是自在使用 Datasets.Item 或 Datasets.Remove 等方法时引用此 Dataset 的方法之一。该名称必须在整个 Datasets 集合的所有 Dataset 中唯一。

Dataset.Refresh 方法

目的

强制从数据源重新读取数据。

语法

Dataset.Refresh

注解

此方法用于在数据发生更改时更新地图。

在调用 Refresh 方法之后，数据将从原始数据源重新读取，并使用由 Dataset 中的字段定义的聚集方法重新聚集。

任何基于 Dataset 的主题将更新以体现新的数据。

注：此方法不适用于 miBindLayerTypeXY 类型的 Datasets。

Dataset.RowValues 属性

目的

此属性返回用于给定行的 RowValues 对象。它将作为行 ID 号 （Long 类型），串行化关键字 （String 类型），或提取关键字的图元对象 （Feature 类型）。

语法

[RowValues=]OBJECT.**RowValues**(Row)

部分	说明
OBJECT	表示 Dataset 对象。
Row	**Variant：**可以为 FeatureID （整数）、 FeatureKey （字符串）或 Feature 对象。

Dataset.RowCount 属性

目的

此属性将返回 Dataset 中的行数。这是一个整数值。这是一个只读属性。

注解

由于数据绑定到图层，Dataset.RowCount 实际将返回图层中的图元数量。如果没有数据绑定到图元，则 Dataset 中的项可以具有 NULL 值。

对于从 DBMS 服务器上创建的数据集， Dataset.RowCount 返回的值将为零。

另请参阅

Layer.AllFeatures 方法，第 343 页

Features.Count 属性，第 282 页

Dataset.ReadOnly 属性

目的

只读属性，表示 Dataset 是否可以更新。目前，只有 miDatasetLayer Datasets 可更新。 Datasets 使用 Layers.UpdateFeature 或 Feature.Update 通过 RowValues 集合更新。

另请参阅

RowValue 对象和 RowValues 集合，第 442 页

Layer.UpdateFeature 方法，第 363 页

Feature.Update 方法，第 280 页

Datasets.Restore 方法

目的

此方法可用于从已经使用 OLE 串行化存储到文件的地图恢复 Dataset。由于 Datasets 依赖于外部数据，Datasets 的内容将不会和地图的其余信息一起串行化到文件。 Datasets.Restore 可用于通过令未串行化的地图指回外部数据，恢复 Dataset （及其主题）。

语法

[Dataset=]OBJECT.**Restore** (Name, SourceData)

部分	说明
OBJECT	Datasets 集合。
Name	字符串参数。是要挥发的 Dataset 的名称，在地图串行化时应为为现有的 Dataset 名称。
SourceData	变量。应该是储存 Dataset 添加到原始 Datasets.Add 时的相同参数。此数据源随后用于恢复 dataset 的内容。

另请参阅

Datasets.Add 方法，第 263 页

Dataset.SecondaryGeoField 属性

目的

Fields 集合中的备用地理关键字列的列名。如果在添加数据时没有指定 SecondaryGeoField，则该值为 0。这是一个整数值，是只读属性。

Dataset.SourceRows 属性

目的

Dataset 的 SourceRows 集合包括源自源数据的行列表，该数据源匹配指定的行。

另请参阅

SourceRow 对象和 SourceRows 集合，第 454 页

Feature.KeyValue 属性，第 275 页

Dataset.Themes 属性

目的

用于 Dataset 的 Themes 集合

另请参阅

Theme 对象和 Themes 集合，第 481 页

Dataset.Type 属性

目的

这是一个只读属性，返回数据集类型。此属性的值将为 **DatasetTypeConstants** 之一。

另请参阅

DataSetTypeConstants，第 565 页

Dataset.Value 方法

目的

此方法检索用于 Dataset 中的指定行和列的值。这是 Dataset 对象的默认属性。

语法

[value=] OBJECT.**Value** (Row, Column)

部分	说明
OBJECT	表示 Dataset 对象。
Row	MapX 中用于获取值的行。 **Variant**：可以为以下项目之一： *FeatureKey *FeatureID *Feature object **注释**：如果您正在使用远程表，则必须使用 FeatureKey 或 Feature 对象。不得使用 FeatureID。
列	**Variant**：可以是列名或索引。

注解

返回的值来自 MapX Dataset，而非原始源数据。由于原始源数据聚集或绑定到图层来形成 MapX Dataset，行数相应于图层中的图元数，并且不是原始的源行数。因此，如果没有数据绑定到图元，则 Dataset 中的单元格可以具有 NULL 值。通过将变量类型设置为 VT_NULL，变量中将返回 NULL 值。

您可以使用 **Feature object** 作为 Row 参数，由于图元对象唯一表示数据集绑定到 Layer 中的单一的行。

如果您正在使用远程表，则必须使用 FeatureKey 或 Feature 对象。不得使用 FeatureID。

另请参阅

Feature.KeyValue 属性，第 275 页

Feature.Name 属性，第 277 页

Datasets.Add 方法

目的

此方法创建指定的 Dataset 并将其添加到集合。

语法

OBJECT.**Add** (Type, SourceData, [Name], [Geofield], [SecondaryGeofield], [BindLayer], [Fields], [Dynamic])

部分	说明
OBJECT	表示 Dataset 对象。
类型	正在添加的 Dataset 对象取 **DatasetTypeConstants** 值。
SourceData	该参数引用数据，因 TYPE 的不同而异 （例如 IDAORecordset、ODBCQueryInfo、 Layer 等对象）。
Name	该参数是唯一标识数据集的字符串。该参数是可选参数，如果未指定，将使用 DatasetN 格式的名称，其中 N 是唯一编号。

部分	说明
Geofield	该参数是包含地理信息的列的名称或索引。该参数是可选参数，如果未指定， MapX 将搜索字段以确定匹配 GeoDictionary 中的条目的字段。 另请参阅：**Map.MatchNumericFields 属性**。
SecondaryGeofield	包含优化地理信息的列的名称或索引。这是一个可选参数。
BindLayer	指定要将此数据附属到的图层，或者如果点参考文件要地理引用导入数据 （例如邮政编码）或包含 Long/Lat 值，则为 **BindLayerObject**。该参数是可选参数，如果未指定， MapX 将在 GeoDictionary 中搜索要连接到的图层。 **注释**：如果在 DataSets.Add 中使用 miBindLayerTypeXY，将创建一个临时表，其中包含一个名为 “GeoName”的列，以便必须在永久图层上创建要成为新数据绑定的 Fields 集合。
Fields	该参数是属于 **Field 对象**集合的 **Field** 对象。Field 对象用于描述数据源中要导入的字段，以及在数据源的多个记录与特定的地图图元匹配时，要使用的合计函数。该参数是可选参数，如果未指定，所有列均将导入，如果每个图元遇到多个记录，则对数据值求和。
Dynamic	**Variant**：控制数据绑定是否为动态的布尔值。该参数是可选参数；如果省略，则默认值为 False，表示绑定是静态的 （即 MapX 将在数据库打开时复制所需的数据）。如果指定 True， MapX 将仅在需要数据时 （例如标记时）实时访问数据。如果指定 True 但是数据集不支持动态列，将发生例外。

注：如果存在二义性 （例如多个 geoset），则用户可以指定时间处理程序来拾取一个。如果没有指定时间处理程序，则 MapX 将拾取第一个。

注：数据集类型：miDatasetLayer － 用于创建数据集，引用源自 MapInfo 表的的字段。 SourceData 参数是一个 Layer 对象。

注：在进行 **BindLayer** 匹配时， Geofields 必须唯一。对于不唯一的数据集，只有第一项将匹配。其余项将忽略。

注解

如果使用自动检测逻辑，则将会发现多个匹配结果。如果要处理这一情况，则可以实施 **ResolveDataBind 事件**处理程序。否则， MapX 将拾取第一个匹配项。

只有 Dataset 绑定到的图层包含的关键字列不唯一时，才需要 SecondaryGeofield。备用 geofield 随后即可唯一匹配图层中的对象 （行）。

如果在 DataSets.Add 中使用 miBindLayerTypeXY，将创建一个临时表，其中包含一个名为 “GeoName”的列，以便必须在永久图层上创建要成为新数据绑定的 Fields 集合。

如果要创建点的新图层，则 GeoField 列必须唯一 （BindLayer.LayerType 是 miBindLayerTypeXY 或 miBindLayerTypePointRef）。GeoField 列将用于新的点图层中的图元。如果值不是唯一的，将导致单个点加入新的点图层，从而出现第一个重复的关键字值，重复行中的数据值将合计。

如果指定了 Fields 集合， Geofield 和 SecondaryGeofield 参数引用 Fields 集合中的列，而不是源数据中的列。

如果使用动态数据绑定，则所有字段均将忽略 Field.AggregationFunction 属性，并且所有字段均将单独聚集。如果有多个匹配记录，则将采用第一个记录。

另请参阅

Field 对象和 Fields 集合，第 302 页

MapX 字段命名规范，第 507 页

Datasets.Contains 方法

目的

如果由该参数指定的数据集当前在集合中，则此属性返回 True。参数可以为 Dataset 名称或基于 1 的索引编号。此方法对于通过 State 对象恢复数据集非常实用。

语法

OBJECT.Contains(Index)

部分	说明
OBJECT	这是一个 Datasets 对象。
Index	这是 Dataset 名称或基于 1 的索引编号。

另请参阅

State 对象，第 456 页

Datasets.Count 属性

目的

集合中的 **Dataset** 对象数。这是一个只读属性。

Datasets.Item 属性

目的

此属性从集合检索特定的 **Dataset** 对象。这是一个变量，您可以指定 Dataset 名称或基于 1 的索引编号。这是 Datasets 集合的默认属性。

Datasets.Remove 方法

目的

从 **Datasets** 集合去除指定的 **Dataset** 对象。

注：如果您去除项目，集合索引将再次计数，填充由已删除项目留下的空洞。

语法

Datasets.Remove (index)

部分	说明
index	Dataset 名称或基于 1 的索引编号。

注解

此外还删除了所有基于此 Dataset 的主题。

Datasets.RemoveAll 方法

目的

将所有 **Dataset** 对象从集合中删除。

语法

Datasets.RemoveAll

Datum 对象

Datum 对象表示由 **CoordSys** 对象使用的所有数据资料。数据是地球的形状和方位的单数数学描述。

对象属性

- **Datum.Eccentricity 属性**
- **Datum.Ellipsoid 属性**
- **Datum.Flattening 属性**
- **Datum.PrimeMeridian 属性**
- **Datum.RotateX、 RotateY、 RotateZ 属性**
- **Datum.ScaleAdjust 属性**
- **Datum.SemiMajorAxis, SemiMinorAxis 属性**
- **Datum.ShiftX、 ShiftY、 ShiftZ 属性**

对象方法

- **Datum.Set 方法**
- **Datum.SetFromList 方法**

注解

所有属性均为只读。要修改 Datum 对象，可使用 Set 方法或 SetFromList 方法。

要获取 **Datum** 对象，可引用 **CoordSys** 对象的 **Datum** 属性。

Datum.Eccentricity 属性

目的

这是只读属性，返回表示数据离心率的双精度值。

注解

在等式中，离心率通常表示为字母 e。

Datum.Ellipsoid 属性

目的

这是只读属性，返回确定 Datum 椭圆率的短精度值，例如 28 用于 “WGS 84”。

Datum.Flattening 属性

目的

这是只读属性，返回表示数据扁平率的双精度值。

注解

在等式中，扁平率通常表示为字母 f。

Datum.PrimeMeridian 属性

目的

这是只读属性，返回双精度值，以格林威治东经度数表示数据的本初子午线经度。

注解

PrimeMeridian 属性通常为 0，因为大部分数据均将格林威治作为本初子午线。但是，某些数据使用不同的地点作为本初子午线。例如， NTF 数据使用巴黎作为其本初子午线，它是东格林尼治 2.33722917 度。如果您在坐标系统中使用 NTF 数据，则该坐标系统中的所有经度都是相对于巴黎的，而不是相对于格林尼治的。

Datum.RotateX、 RotateY、 RotateZ 属性

目的

这是一个只读属性，返回双精度值，以弧度 - 秒为单位表示数据的椭球体沿其各个轴旋转的角度。

Datum.ScaleAdjust 属性

目的

只读属性，返回双精度值，表示数据的比例修正因子 （百万分之几，用于调整椭球体的大小）。

注解

在等式中，比例调整通常表示为字母 m。

Datum.SemiMajorAxis, SemiMinorAxis 属性

目的

这是只读属性，返回表示数据半长轴和半短轴的双精度值。长度的单位为米。

注解

在等式中，其半长轴通过用字母 a 表示，其半短轴通过用字母 b 表示。

Datum.Set 方法

目的

此方法设置构成 **Datum** 对象的属性。

语法

OBJECT.**Set** (Ellipsoid, ShiftX, ShiftY, ShiftZ, RotateX, RotateY, RotateZ, ScaleAdjust, PrimeMeridian)

部分	说明
OBJECT	**Datum** 对象。
Ellipsoid	短精度值，必须为支持的椭球体数字之一 （例如 28 代表 WGS 84 椭球体）。
ShifX, ShiftY, ShiftZ	双精度值，表示椭球体沿其各个轴移动的距离 （以米为单位）。
RotateX, RotateY, RotateZ	双精度值，以弧度 - 秒为单位表示椭球体沿其各个轴旋转的角度。
ScaleAdjust	双精度值，以百万分之几为单位表示用来调整椭球体大小量值的比例修正因子。
PrimeMeridian	双精度值，表示本初子午线的经度。通常为 0，参阅 PrimeMeridian 属性。

注解

所有参数均为必需。

另请参阅

Datum.SetFromList 方法，第 269 页

Datum.SetFromList 方法

目的

此方法设置 **Datum** 对象的所有属性，匹配数据列表中的数据值之一。

语法

OBJECT.Set DatumNumber

部分	说明
OBJECT	**Datum** 对象。
DatumNumber	短精度值，必须为支持的数据数字之一 （例如 62 代表 NAD 27 数据，表示美国大陆）。

Datum.ShiftX、 ShiftY、 ShiftZ 属性

目的

这些只读属性返回双精度值，表示数据椭球体沿其各个轴移动的距离 （以米为单位）。

Feature 对象和 Features 集合

每个图层都包含所选 Feature 对象的集合 (Layer.Selection)。 Feature 对象对应于图层中的图元 （实际实体），例如纽约或芝加哥。

注：如果所选的地图图元位于另一被选图层的另一对象之上，则原始图元将不会显示为选中。例如在 “世界地图” geoset 上选择 “美国”，然后按住 CTRL 键，选择海洋。则 “美国”将不会显示为选中。但是两个项目，均将处于其各自图层的 Selection 集合中。

Features 集合对象表示一个图层的 Feature 对象的集合。集合中的所有图元必须源自相同图层。您可以通过调用 Layer Search 方法之一来获取图元，使用该方法返回 Features 集合。在调用 Search 方法时，Features 对象包含处于该图层中的图元。从图层增减图元不会影响任何现有的 Feature 集合。

对象属性

- **Feature.Bounds 属性**
- **Feature.Caption 属性**
- **Feature.CenterX 属性 , CenterY 属性**
- **Feature.FeatureID 属性**
- **Feature.FeatureKey 属性**
- **Feature.HasMultipoint 属性**
- **Feature.HasPolyline 属性**
- **Feature.HasRegion 属性**
- **Feature.KeyValue 属性**
- **Feature.LabelPoint 属性**
- **Feature.Layer 属性**
- **Feature.Length 属性**
- **Feature.Multipoint 属性**
- **Feature.Name 属性**
- **Feature.Nodes 属性**
- **Feature.Parts 属性**
- **Feature.Perimeter 属性**
- **Feature.Point 属性**
- **Feature.Polyline 属性**
- **Feature.Region 属性**
- **Feature.Smooth 属性**
- **Feature.Style 属性**
- **Feature.Type 属性**

对象方法

- **Feature.Attach 方法**
- **Feature.Clone 方法**
- **Feature.Offset 方法**

- **Feature.Update 方法**

集合属性

- **Features.Bounds 属性**
- **Features.Count 属性**
- **Features.Item 属性**

集合方法

- **Features.Add 方法**
- **Features.AddByID 方法**
- **Features.Clone 方法**
- **Features.Common 方法**
- **Features.Remove 方法**
- **Features.RemoveByID 方法**
- **Features.Replace 方法**

另请参阅

Selection 集合，第 446 页

Layer.Search 方法，第 357 页

Feature.Area 属性

目的

只读属性，返回区域图元的面积，对于其他类型的图元无效。

注解

要控制返回的面积所用的单位，可设置 **Map.AreaUnit** 属性。

这对于区域有效。该值采用面积单位。如果您修改了图层中的图元，并且没有执行更新以保存更改，则此属性随后将返回一个基于 “已修改”版本对象的的值 （该对象尚处于内存中）。

另请参阅

Map.AreaUnit 属性，第 390 页

Feature.Attach 方法

目的

将独立的图元连接到地图，以便地图坐标系应用于相应图元。

语法

OBJECT.Attach (Map)

部分	说明
OBJECT	一个图元对象。
Map	一个地图对象。

注解

在创建独立图元对象时，必须将该图元对象连接到地图上，然后才能引用其方法或属性。将图元连接到地图上就是将地图的坐标系与图元关联。但是，如果通过调用 FeatureFactory 对象的方法创建独立图元，即无需调用 Attach 方法。

Feature.Clone 方法

目的

将图元复制为新的独立 Feature 对象。

语法

[=Feature]**Feature.Clone**

Feature.Bounds 属性

目的

返回 **Rectangle** 对象，表示图元的地理范围 （其最小边界矩形）。

注解

在 MapX 的早期版本中，此属性称为 **MBR**。

如果您修改了图层中的图元，并且没有执行更新以保存更改，则此属性随后将返回一个基于 “已修改” 版本对象的的值 （该对象尚处于内存中）。

另请参阅

Rectangle 对象，第 437 页

Feature.Caption 属性

目的

读写字符串值，表示构成 Text 图元的文本字符串。如果不是 Text 图元，则引用此属性将会导致异常错误。

语法

[= string] OBJECT.**Caption**

部分	说明
OBJECT	一个 Feature 对象。
= string	一个文本字符串，最长 255 个字符。

另请参阅

Feature 对象和 Features 集合，第 271 页

Feature.CenterX 属性 , CenterY 属性

目的

这是只读属性，返回作为双精度值，表示图元质心的 X 和 Y 地图坐标。

注解

在需要访问图元质心时，使用 LabelPoint 属性将更加轻松，取代了使用 CenterX 和 CenterY。

Feature.FeatureID 属性

目的

此属性返回图元 ID。图层中的每个图元均包含在图层内唯一的 ID。该属性是一个整数值，是只读的。这是 **Feature** 对象的默认属性。

注：尽管此属性仍然是 MapX 函数部分，但是建议您使用 **Feature.FeatureKey**。 FeatureKey 属性用于确定表中的唯一记录。在此前的版本中， FeatureID 属性用于此目的，但是对于无缝和远程图层无法正常工作。 FeatureKey 属性适用于所有图层类型。

Feature.FeatureKey 属性

目的

此属性返回图元的关键字。图层中的每个图元均包含在图层内唯一的关键字。这是一个字符串值，是只读的。这是 **Feature.FeatureID** 的替代 （此属性和此前的功用相同，但是建议您使用 FeatureKey 属性）。

注：图元的关键字可在 FeatureID 应用时用作参数。 FeatureKey 属性用于确定表中的唯一记录。在此前的版本中， FeatureID 属性用于此目的，但是对于无缝和远程图层无法正常工作。 FeatureKey 属性适用于所有图层类型。

另请参阅

Selection.SelectByRegion 方法，第 452 页

Dataset.Value 方法，第 262 页

Dataset.SourceRows 属性，第 262 页

Layer.UpdateFeature 方法，第 363 页

Layer.DeleteFeature 方法，第 347 页

Features.AddByID 方法，第 281 页

Features.RemoveByID 方法，第 283 页

Selection.SelectByID 方法，第 449 页

Feature.KeyValue 属性

目的

这一读写字符串属性用于在添加或更新图元或图层是，更新或设置 MapInfo 表中的列 （字段）。

注解

设置或检索的字段值由图元源自的图层的 KeyField 属性值确定。尽管该属性是字符串属性，则该字符串将在更新或插入时转换为正确的列类型。

在添加或更新图元只使用 KeyValue 属性最近的值。如果您设置 KeyValue，则将图层的 keyfield 属性设置为不同的字段，然后再次设置 keyvalue，只有用于该图元的 MapInfo 表中的第二个值将会更新。

另请参阅

Layer.KeyField 属性，第 353 页

Layer 对象和 Layers 集合，第 340 页

Feature.HasMultipoint 属性

目的

如果 Collection 对象具有 Multipoint 部分，则此只读属性返回 true，否则返回 false。

语法

[bool=] OBJECT.**HasMultipoint**

Feature.HasPolyline 属性

目的

如果 Collection 对象具有 Polyline 部分，则此只读属性返回 true，否则返回 false。

语法

[bool=] OBJECT.**HasPolyline**

Feature.HasRegion 属性

目的

如果 Collection 对象具有 Region 部分，则此只读属性返回 true，否则返回 false。

语法

[bool=] OBJECT.**HasRegion**

Feature.LabelPoint 属性

目的

一个 Point 对象，是图元的质心 （标记点）的地图坐标中的位置。

另请参阅

Point 对象和 Points 集合，第 429 页

Feature.Layer 属性

目的

只读属性，返回图元源自的 Layer 对象。

另请参阅

Layer 对象和 Layers 集合，第 340 页

Feature.Length 属性

目的

这是只读属性，返回构成多义线的线段总长。如果图元有若干组成部分，则返回的值是各个部分的所有线段的总和。

注：该长度计算不考虑地球的曲度，因此对于跨度较长的线段通常不够精确。长度使用修正的迪卡

尔坐标计算：经度 / 纬度坐标将转换为 X, Y 坐标，采用 MapX 的当前地图单位，考虑各线段质心处每一纬度的英里数。在生成的点之间的距离随后将使用毕达哥拉斯定理计算。

注解

要控制返回的长度所用的单位，可设置 **Map.MapUnit** 属性。

如果您修改了图层中的图元，并且没有执行更新以保存更改，则此 Feature.Length 随后将返回一个基于“已修改”版本对象的的值 （该对象尚处于内存中）。

另请参阅

Feature.Perimeter 属性，第 278 页

Map.MapUnit 属性，第 405 页

Feature.Multipoint 属性

目的

这是一个读写属性，返回图元的 Multipoint 部分，其类型为 miFeatureTypeCollection。

注解

从此属性读取，返回集合图元 （如已存在）的 Multipoint 部分。此图元返回独立图元，如果对此图元进行编辑后要保存所做编辑，用户需要更新父级图元。

写入此属性将复制传递的图元，并将其设置到现有的集合图元之中。

Feature.Name 属性

目的

这是一个只读属性，返回图元名称。图元名称是 LabelProperties.DataField 字段 （还显示图元的标签）的值。

另请参阅

LabelProperties.DataField 属性，第 335 页

Feature.Nodes 属性

目的

这一只读属性展示展示节点数据，用户可以在一个对象中通过一次传递查询所有节点，然后令其返回一个单一的连续的内存块。

语法

[SafeArray=] OBJECT.**Nodes** ([CSys])

部分	说明
OBJECT	表示 Feature 对象的占位符。
CSys	CoordSys 对象，定义节点数据返回到的坐标系。默认为 Map.NumericCoordSys。

Feature.Offset 方法

目的

相对于图元的当前位置移动图元。

语法

OBJECT.Offset (deltaX, deltaY)

部分	说明
OBJECT	表示 Feature 对象的占位符。
deltaX	移动点所依据的地图坐标。
deltaY	移动点所依据的地图坐标。

Feature.Parts 属性

目的

只读属性，返回 **Parts** 对象。对于线或区域有效。

另请参阅

Parts 集合，第 427 页

Feature.Perimeter 属性

目的

只读属性，返回区域图元的周长，对于其他类型的图元无效。

注解

要控制返回的周长所用的单位，可设置 **Map.MapUnit** 属性。

如果您修改了图层中的图元，并且没有执行更新以保存更改，则此属性随后将返回一个基于 “已修改” 版本对象的的值 （该对象尚处于内存中）。

另请参阅

Feature.Length 属性，第 276 页

Feature.Point 属性

目的

只读属性，返回 Point 对象。只对点和文本类型的图元有效。

另请参阅

Point 对象和 Points 集合，第 429 页

Feature.Polyline 属性

目的

这是一个读写属性，返回图元的 Polyline 部分，其类型为 miFeatureTypeCollection。

注解

从此属性读取，返回集合图元 （如已存在）的 Polyline 部分，并返回为一个独立对象。如果对此图元进行编辑后要将所做编辑保存到表，用户需要更新父级图元。

写入此属性将复制传递的图元，并将其设置到现有的集合图元之中。

Feature.Region 属性

目的

这是一个读写属性，返回图元的区域部分，其类型为 miFeatureTypeCollection 。

注解

从此属性读取，返回集合图元 （如已存在）的区域部分，并返回为一个独立对象。如果对此图元进行编辑后要将所做编辑保存到表，用户需要更新父级图元。

写入此属性将复制传递的图元，并将其设置到现有的集合图元之中。

Feature.Smooth 属性

目的

Feature.Smooth 对象令您可以平滑多义线，将其制成一条连续的曲线。要令多义线平滑，将此属性设置为 True。

Feature.Style 属性

目的

此读写属性返回图元的样式。

另请参阅

Style 对象，第 463 页

注解

Style 对象默认为 '0'，用于样式对象的每个属性。

Feature.Type 属性

目的

包含图元的类型。这是 **FeatureTypeConstants** 值，为读写属性。

另请参阅

FeatureFactory.CreateText 方法，第 296 页

Feature.Update 方法

目的

此方法使用对 Feature 对象所作更改更新图层。

语法

OBJECT**.Update ([UpdateFeature] , [RowValues])**

部分	说明
OBJECT	表示 Feature 对象的占位符。
UpdateFeature	布尔值，表示是否更新对象列。默认值为 True。
RowValues	RowValues 表示一行数据的属性数据新值。 **RowValues 集合**中的每个值均相应于属性数据的一列。此参数只适用于 miDatasetLayer 类型的 Dataset。

注：对象列存储 Feature 对象的地理数据。

注解

如果 RowValues 集合指向通过 Dataset.AddField 方法添加的字段，则这些字段将为可更新。

Features.Add 方法

目的

将 **Feature** 对象或 **Features** 对象中的所有图元添加到集合 （UNION 集操作）。

语法

OBJECT.**Add** (Source)

部分	说明
OBJECT	表示 Features 对象的占位符。
Source	要添加到 OBJECT 中的 Features 集合的 Feature 对象或 Features 集合。

另请参阅

Feature 对象和 Features 集合，第 271 页

Features.AddByID 方法

目的

将带有指定 FeatureKey 的 **Feature** 对象添加到 **Features** 集合。

语法

OBJECT.**AddByID**(FeatureKey)

部分	说明
OBJECT	表示 Features 对象的占位符。
FeatureKey	要向 Features 集合添加的图元的 FeatureKey

注：Feature.FeatureKey 是 **Feature.FeatureID** 的替代。 FeatureID 和此前的功用类似，但建议使用新的 FeatureKey 属性。 FeatureKey 用于确定表中的唯一记录。在此前的版本中， FeatureID 属性用于此目的，但是对于无缝和远程图层无法正常工作。 FeatureKey 属性适用于所有图层类型。

另请参阅

Feature.KeyValue 属性，第 275 页

Features.Bounds 属性

目的

返回 **Rectangle** 对象，表示集合中所有对象的地理范围 （如其最小边界矩形）。

语法

OBJECT.**Bounds**

OBJECT 占位符表示 **Features** 集合。

注解

如果要缩小地图以显示集合中的全部对象，此属性非常实用：

Features.Clone 方法

目的

此方法将集合复制为另一个 **Features** 对象。

语法

OBJECT.Clone (source)

部分	说明
OBJECT	表示 Features 集合的占位符。
source	表示要使用集合 OBJECT 复制的图元集合。

Features.Common 方法

目的

将此集合与另一个 Feature 对象合并，使该集合只包含两个集合中均存在的图元。（INTERSECT 集操作）。

语法

OBJECT**.Common** (source)

部分	说明
OBJECT	表示 Features 集合的占位符。
source	表示要使用集合 OBJECT 合并的图元集合。

Features.Count 属性

目的

集合中的项数。

Features.Item 属性

目的

从集合返回 **Feature** 对象 － － 当前只支持数字索引。这是 **Features** 集合的默认属性。

Features.Remove 方法

目的

从此集合中删除 **Feature** 对象或 Features 对象中的所有图元 （SUBTRACT 集合运算）。

语法

OBJECT.**Remove** (source)

部分	说明
OBJECT	表示 Features 对象的占位符。
source	表示要从集合 OBJECT 删除的图元集合。

Features.RemoveByID 方法

目的

使用来自 **Features** 集合中的指定 FeatureID 删除 **Feature** 对象。

语法

OBJECT.**RemoveByID** (*FeatureKey)

部分	说明
OBJECT	表示 Features 对象的占位符。
FeatureKey	* 要从 **Features** 集合删除的图元的 FeatureKey 是 FeatureID 参数的替代参数。FeatureID 和此前的功用类似，但建议使用新的 FeatureKey 参数。 FeatureKey 用于确定表中的唯一记录。在此前的版本中， FeatureID 属性用于此目的，但是对于无缝和远程图层无法正常工作。 FeatureKey 属性适用于所有图层类型。

Features.Replace 方法

目的

使用 Feature 对象或 **Features** 对象中的所有图元取代集合中的内容。

语法

OBJECT.Replace (source)

部分	说明
OBJECT	表示 Features 对象的占位符。
source	表示图元或图元集合，将用于替代 “图元”内容的内容。

FeaturesConstraint 对象

这一对象允许程序员通过指定 MapX Features 集合，指定将哪些图元缓存用于远程数据库图层。通过调用 Layer.CreateFeaturesConstraint 方法创建了 FeaturesConstraint 对象。图层可以有多个与其关联的 FeaturesConstraint 对象。 FeaturesConstraint 对象的 Enable 和 Disable 方法可在它创建之后用于添加或去除此约束。 Enabled 属性可用于确定当前的约束状态。 Features 属性可用于设置或检查定义此约束的 Features 集合。

要强制刷新缓存，可使用 Layer.Refresh 方法。

对象属性

- **FeaturesConstraint.Enabled 属性**
- **FeaturesConstraint.Features 属性**

对象方法

- **FeaturesConstraint.Disable 方法**
- **FeaturesConstraint.Enable 方法**

示例

- **VB 示例**

另请参阅

LayerInfo 对象，第 373 页

使用缓存，第 166 页

Feature 对象和 Features 集合，第 271 页

Layer.CreateFeaturesConstraint 方法，第 345 页

AllFeaturesConstraint 对象，第 236 页

BoundsConstraint 对象，第 249 页

FeaturesConstraint.Disable 方法

目的

在 FeaturesConstraint 对象创建之后，此方法和 Enable 方法一起用于控制是否使用由对象定义的约束。在启用之后， FeaturesConstraint 对象用于确定将哪些记录存储在缓存中。初始化之后，FeaturesConstraint 对象将在默认情况下启用。

语法

OBJECT.**Disable**

OBJECT － FeaturesConstraint 对象

另请参阅

使用缓存，第 166 页

FeaturesConstraint.Enabled 属性，第 286 页

FeaturesConstraint.Enable 方法，第 286 页

FeaturesConstraint.Enabled 属性

目的

这是一个布尔值，只读属性。在为 TRUE 时， FeaturesConstraint 对象用于确定将哪些地图图元存储在缓存中。在为 FALSE 时，将不会使用 FeaturesConstraint 对象。

另请参阅

使用缓存，第 166 页

FeaturesConstraint.Enable 方法，第 286 页

FeaturesConstraint.Disable 方法，第 285 页

FeaturesConstraint.Features 属性

目的

这是读 / 写属性，可以取或返回 MapX Features 对象。Features 集合确定源自远程图层的哪些记录存储在缓存中。

语法

[Rectangle=] OBJECT.**Bounds**

Rectangle － MapX Rectangle 对象

OBJECT － FeaturesConstraint 对象

另请参阅

Feature 对象和 Features 集合，第 271 页

使用缓存，第 166 页

FeaturesConstraint.Enable 方法

目的

在 FeaturesConstraint 对象创建之后，此方法和 Disable 方法一起用于控制是否使用由对象定义的约束。在禁用的 FeaturesConstraint 对象不用于确定将哪些记录存储在缓存中。在启用的 FeaturesConstraint 对象用于确定将哪些记录存储在缓存中。在初始化之后， FeaturesConstraint 对象将在默认情况下启用，此方法只需在 FeaturesConstraint 通过 FeaturesConstraint.Disable 方法禁用之后。

语法

OBJECT.**Enable**

OBJECT － FeaturesConstraint 对象

另请参阅

使用缓存，第 166 页

FeaturesConstraint 对象，第 285 页

FeaturesConstraint.Enabled 属性，第 286 页

FeaturesConstraint.Disable 方法，第 285 页

FeatureFactory 对象

此对象允许您创建新的地图图元，或通过对现有图元执行操作 （例如缓冲）来创建图元。

对象方法

- **FeatureFactory.BufferFeatures 方法**
- **FeatureFactory.CombineFeatures 方法**
- **FeatureFactory.CreateArc 方法**
- **FeatureFactory.CreateCircularRegion 方法**
- **FeatureFactory.CreateCollectionFeature 方法**
- **FeatureFactory.CreateEllipticalRegion 方法**
- **FeatureFactory.CreateLine 方法**
- **FeatureFactory.CreateMultipoint 方法**
- **FeatureFactory.CreateRegion 方法**
- **FeatureFactory.CreateSymbol 方法**
- **FeatureFactory.CreateText 方法**
- **FeatureFactory.EraseFeature 方法**
- **FeatureFactory.IntersectFeatures 方法**
- **FeatureFactory.IntersectionPoints 方法**
- **FeatureFactory.IntersectionTest 方法**

注解

其中大多数方法均返回独立图元对象。这些图元对象自动连接到地图上 （即已与坐标系关联）。也就是说，不需要对这些方法返回的图元使用 Attach 方法。

要获取 FeatureFactory 对象，应引用 **Map.FeatureFactory** 属性。

FeatureFactory.BufferFeatures 方法

目的

此方法返回独立的 **Feature** 对象，表示缓冲区域。

语法

[Feature =] OBJECT.**BufferFeatures** (Source, Distance, [Units] , [Resolution])

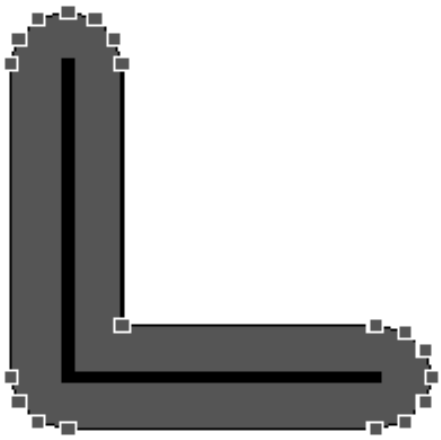
部分	说明
OBJECT	FeatureFactory 对象
Source	Feature 对象或 Features 集合对象。点、线、文本和区域图元可进行缓存。
Distance	缓存的大小，使用由 units 参数指定的单位。如果距离为正，则缓冲区域大于源图元。对区域进行缓存时，可以指定负距离，产生小于源区域的缓存区域。

部分	说明
Units	**Variant**：**MapUnitConstants** 值，例如 miUnitMile (0)，确定 distance 参数的测量单位。如果忽略，则将 MapUnit 用作默认值。
Resolution	**Variant**：正整数，表示创建缓存时使用的节点数。这是缓存操作中每个圆内的节点数，不是整个区域中的节点总数 （如下所示）。如果忽略，则将分辨率将由 DefaultConversionResolution 属性控制。

注解

此方法返回单缓存区域，无论是否一个 Feature 还是 Features 集合上的缓存。如果每集合中的每个图元创建单独的缓存区域，您必须遍历集合，并一次对一个图元使用 BufferFeatures 方法。

如果创建用于在缓存中执行搜索的单一目的的缓存，可以使用 SearchWithinDistance 方法来替代 BufferFeatures。



resolution 参数控制在缓存区域中有多少个节点。Resolution 是就每个圆中的节点数指定的，如示例所示。此图片显示两个图元：黑色的 L 型多义线和灰色缓存区域。区域由创建多义线周围的缓存来生成。在图示中，小方块标记了缓存区域中各个节点的位置。此缓存是使用每个圆中 12 个节点的分辨率生成的。注意多义线的每端均有一个半圆型的帽，每个半圆包括 6 个线段。如果使用较高的分辨率，则在每个半圆中将有多个节点。

分辨率节点无需表示图元中的节点总数。分辨率是按照每个圆中的节点来表示的，而且给定图元可以包含多个半圆。例如，以上显示的缓存区域中的节点多于 12 个，尽管分辨率指定每个圆 12 节点。指定较大分辨率将生成更为平滑的区域，但是生成用时较长，并且生成的对象需要更多的存储空间。

在封闭的对象上允许缓冲宽度为 0：Region、 Rect、 RoundRect、 IsEllipse 和 Text。对于其他对象类型，将会出现 “创建缓存时出错”的异常错误。

另请参阅

FeatureFactory.CreateCircularRegion 方法，第 293 页

FeatureFactory.CombineFeatures 方法

目的

此方法返回独立的 Feature 对象，表示多个线图元或多个区域图元的组合 （并集）。

语法

[Feature =] OBJECT.**CombineFeatures**(feature1, [feature2])

部分	说明
OBJECT	**FeatureFactory** 对象。
feature1	**Feature** 对象或 Features 集合对象。
feature2	**Variant**：A**Feature** 对象或 **Features** 集合对象。

注解

如果 feature1 是 Features 集合，则 feature2 参数可选。如果 feature1 是 Feature 对象，则 feature2 参数为必需。您不能组合线图元和区域图元。支持以下组合：

- 您可以组合两个或多个线图元 （Type 属性为 miFeatureTypeLine 的图元），此时结果为线图元。
- 您可以组合两个或多个区域图元 （类型：miFeatureTypeRegion），此时结果为区域图元。

FeatureFactory.CreateArc 方法

目的

这将创建形似弧的线图元，并将其返回独立的 Feature 对象。

语法

[Feature =] OBJECT.**CreateArc**(Point1, Point2, [Angle] , [Distance] , [Resolution] , [Style])

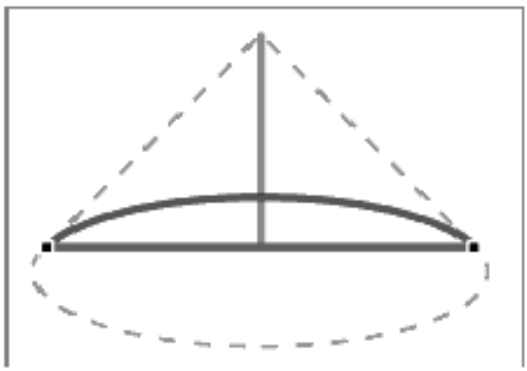
部分	说明
OBJECT	**FeatureFactory** 对象。
Point1, Point2	两个 Point 对象，表示弧的起点和终点。
角度	**Variant**：使用度数表示的角度。可选，如果忽略，则默认角度为 90 度，生成对称的弧。
Distance	**Variant**：补偿线的长度表示为 “弧度单位” 数 （其中一个弧度单位为 Point1 和 Point2 之间的距离）。值越大，弧度就越大。指定 0 将产生水平弧线。可选，如果忽略，则默认为 1。
Resolution	**Variant**：正整数，表示在创建椭圆时使用的节点数，最大数为 32,763。可选，如果忽略，则 defaultConversionResolution 属性控制分辨率。
Style	**Variant**：**Style** 对象定义图元的外观。可选；如果省略，则不设置样式。

注解

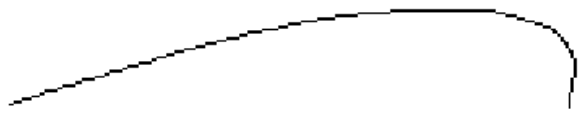
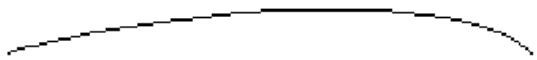
此方法返回线图元 （Type 属性为 miFeatureTypeLine 的图元），形似弧线。注：如果当前的 DisplayCoordSys 不匹配当前的 NumericCoordSys，弧线显示为扭曲。

要计算弧的路径，可假设连接 Point1 和 Point2 线 （图中左侧的水平线）。

其次，假设第二条线 （图中垂直的线），始自第一条线的中点。第二条线的长度由 Offset 参数确定，其相对于第一条线的角度由 Angle 参数确定。要令第二条线正交，如图所示，指定 90 度或 270 度。Point1、 Point2 和第二条线的端点确定一个三角形。这一三角形作为控制多边形，控制弧的形状。弧位于三角形之内，其切入 Point1 和 Point2 处的三角形侧边。该弧位于 Point1 和 Point2 之间的线的一侧。



要令弧位于线的另一侧，可使用不同的 Angle 参数 （例如取代 90 度，指定 270 度）。每个角度和偏移量值的组合将产生唯一的图元形状。如果您始终使用相同组合的角度和偏移量值，生成的弧将始终具有相同形状，无论 Point1 和 Point2 距离多少。部分示例如下所示。

角度	偏移量	结果
30	1	
30	0.5	

角度	偏移量	结果
90	1	
90	0.5	
90	0.25	
165	1	
165	0.5	
270	0.5	

另请参阅

FeatureFactory.CreateEllipticalRegion 方法，第 294 页

FeatureFactory.CreateCircularRegion 方法

目的

这方法创建形似圆的区域图元，并将其返回为独立的 Feature 对象。图元可以添加到用于搜索等操作的图层。

语法

[Feature =] OBJECT.**CreateCircularRegion**(Type, Point, Distance, [Units] , [Resolution] , [Style])

部分	说明
OBJECT	**FeatureFactory** 对象。
类型	**Short**:**CircleTypeConstant**s 值：指定 miCircleTypeScreen(value:0)or miCircleTypeMap (1)。（如下所示）。
Point	**Point 对象**：表示用作圆心的位置，在当前的数字坐标系中指定。
Distance	**双精度**：圆的半径，使用由 Units 参数指定的单位。
Units	**Variant**：**MapUnitConstants** 值，例如 miUnitMile (0)，确定 Distance 参数的测量单位。可选，如果忽略，则 MapUnit 属性控制 Units。
Resolution	**Variant**：正整数，表示在创建圆时使用的节点数，最大数为 32,767，最小值为 3。可选，如果忽略，则 DefaultConversionResolution 属性控制分辨率。指定较大分辨率将生成更为平滑的区域，但是生成用时较长，并且生成的对象需要更多的存储空间。
Style	**Variant**：**Style** 对象定义图元的外观。可选；如果省略，则不设置样式。

注解

如果类型为 miCircleTypeScreen， MapX 创建了在屏幕显示为圆形的区域。但是，圆上不同的点距离圆心的地理距离可能不同。MapX 沿 x 轴计算点，采用距离圆心的距离单位。该距离转换为屏幕单位（像素），用于半径计算圆中其余的节点。注：如果在创建区域之后更改地图的显示坐标系，则区域将显示为偏向一侧，而不是圆。

如果类型是 miCircleTypeMap，则 MapX 创建的区域上的点到中心的地理距离均相同。但是，由于地球曲率，由于坐标系和地图投影的方式，类似的圆显示为偏向一侧，而不是圆。

另请参阅

FeatureFactory.BufferFeatures 方法，第 288 页

FeatureFactory.CreateEllipticalRegion 方法，第 294 页

FeatureFactory.CreateCollectionFeature 方法

目的

这将创建集合类型图元，并将其返回独立的 Feature 对象。图元可以添加到用于搜索等操作的图层。

语法

[Feature=] OBJECT.**CreateCollectionFeature** ([feature1], [feature2], [feature3])

注解

此方法创建独立的 Feature 对象。您可以将三个图元作为参数传递给此方法。您最多可以传递一个区域图元，一个多义线图元读写和一个多点图元。如果没有传递任何参数，则将创建 miFeatureTypeCollection 类型的空图元读写。您不能将相同类型的图元传递给此方法。

部分	说明
OBJECT	FeatureFactory 对象
feature1	**Variant**：表示区域、多义线或多点图元的图元。这是一个可选参数。
feature2	**Variant**：表示区域、多义线或多点图元的图元。这是一个可选参数。
feature3	**Variant**：表示区域、多义线或多点图元的图元。这是一个可选参数。

FeatureFactory.CreateEllipticalRegion 方法

目的

这将创建形似椭圆的区域图元，并将其返回独立的 Feature 对象。图元可以添加到用于搜索等操作的图层。

语法

[Feature =] OBJECT.**CreateEllipticalRegion**(Rectangle, [Angle] , [Resolution] , [Style])

部分	说明
OBJECT	表示 **FeatureFactory** 对象。
Rectangle	Rectangle 对象，表示地图的当前数字坐标系中的椭圆的最小边界矩形 (MBR)。
角度	**Variant**：椭圆的旋转角度。椭圆从其中点旋转。
Resolution	**Variant**：正整数，表示在创建椭圆时使用的节点数，最大数为 32,763。如果忽略，则 DefaultConversionResolution 属性控制分辨率。

部分	说明
Style	**Variant**：定义图元外观的 Style 对象。可选；如果省略，则不设置样式。

注解

如果当前的 DisplayCoordSys 不匹配当前的 NumericCoordSys，椭圆显示为扭曲。

另请参阅

FeatureFactory.CreateArc 方法，第 290 页

FeatureFactory.CreateCircularRegion 方法，第 293 页

FeatureFactory.CreateLine 方法

目的

此方法返回独立的 **Feature** 对象 （线对象），构建自点的**集合**。

语法

[Feature =] OBJECT.**CreateLine**([Points] , [Style])

部分	说明
OBJECT	FeatureFactory 对象
Points	**Variant**：Points 集合，表示用于定义线的点的集合。可选，如果忽略，则不向线图元添加点。
Style	**Variant**：定义图元外观的 Style 对象。可选；如果省略，则不设置样式。

FeatureFactory.CreateMultipoint 方法

目的

此方法返回独立的 Feature 对象 （多点图元），构建自点的集合。

语法

[Featurel=] OBJECT.**CreateMultipoint** ([Points], [Style])

部分	说明
OBJECT	FeatureFactory 对象
Points	**Variant**：Points 集合，表示用于定义多点的点的集合。可选，如果忽略，则不向多点图元添加点。
Style	**Variant**：定义图元外观的 Style 对象。可选；如果省略，则不设置样式。

FeatureFactory.CreateRegion 方法

目的

此方法返回独立的 Feature 对象 （区域图元），构建自点的集合。

语法

[Feature =] OBJECT.**CreateRegion**([Points] , [Style])

部分	说明
OBJECT	FeatureFactory 对象
点	**Variant**：Points 集合，表示用于定义区域或 Rectangle 对象的点的集合。传递 Rectangle 对象，使用矩形四角创建区域。可选，如果忽略，则不向图元添加点。
Style	**Variant**：定义图元外观的 Style 对象。可选；如果省略，则不设置样式。

FeatureFactory.CreateSymbol 方法

目的

此方法返回独立的 Feature 对象 （点图元），构建自 Point 对象。

语法

[Feature =] OBJECT.**CreateSymbol**([Point] , [Style])

部分	说明
OBJECT	FeatureFactory 对象
Point	**Variant**：Point 对象，表示应该放置符号的 x/y 位置。可选；如果省略，则不设置图元位置。
Style	**Variant**：定义图元外观的 Style 对象。如果省略，则不设置样式。

FeatureFactory.CreateText 方法

目的

此方法返回独立的 Feature 对象 （文本图元）。

语法

[Feature =] OBJECT.**CreateText**([Point] , [Caption] , [Position] , [Style])

部分	说明
OBJECT	**FeatureFactory** 对象。
Point	**Variant**：**Point** 对象，表示应该放置文本的位置。可选；如果省略，则不设置图元位置。
Caption	**Variant**：定义文本的字符串。可选；如果省略，则不设置图元说明。
Position	**PositionConstants** 值，控制文本相对于锚点的初始位置。可选，如果忽略则默认为 miPositionTL （即锚点位于文本左上角的顶部）。 此参数只在指定 Point 和 Caption 参数之后生效。
Style	**Variant**：定义图元外观的 Style 对象。如果省略，则不设置样式。

注解

通过指定 Point 和 Position 来控制文本图元的位置。但是要注意 Position 属性只能用于指定文本的初始位置，Feature 对象不提供允许您重设文本方向的 Position 属性。在此方法创建文本图元之后，即无法切换图元相对于锚点的方向。

FeatureFactory.EraseFeature 方法

目的

此方法通过从另一图元 “消除” 一个图元面积，返回独立的 Feature 对象。

语法

[Feature =] OBJECT.**EraseFeature**(SourceFeature, EraserFeature)

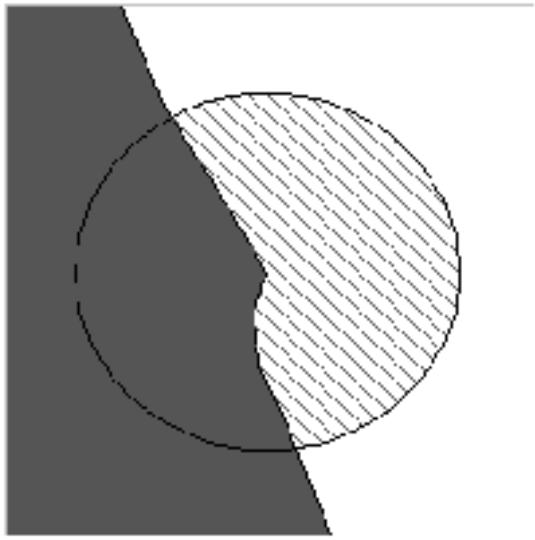
部分	说明
OBJECT	FeatureFactory 对象
SourceFeature	Feature 对象，要消除的一部分，可以是线条或区域图元。
EraserFeature	Feature 对象，表示要从 SourceFeature 删除的面积。这必须是区域图元。

注解

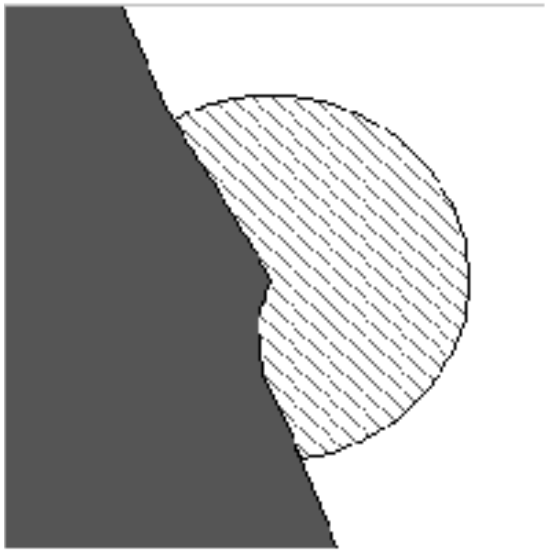
此方法通过从 SourceFeature 削减 EraserFeature 来创建 Feature 对象。如果两个图元没有相交，则没有消除任何东西，并且由此方法返回的图元和 SourceFeature 相同。

示例

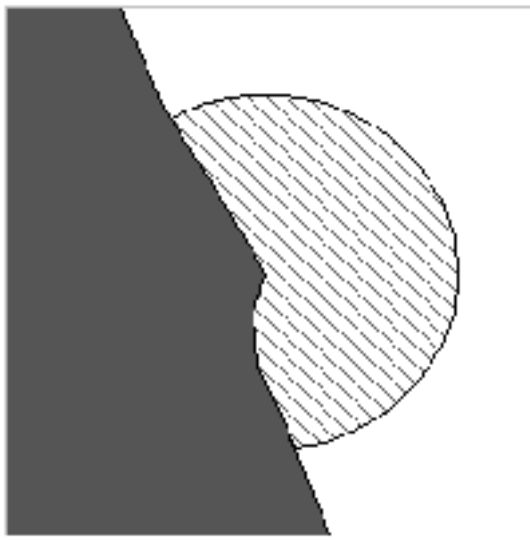
此图片显示了圆形区域，通过 BufferFeatures 方法创建。在此例中，圆形区域覆盖了海岸线，灰色区域表示陆地，白色区域表示水面。



要消除圆形区域的组成部分，可使用 EraseFeature 方法。如果令 SourceFeature 成为圆形区域，令 EraserFeature 成为灰色区域，将会消除覆盖灰色区域的组成部分。



此图片显示了在调用 EraseFeature 方法之后圆形区域上剩余的对象。 EraseFeature 方法去除了覆盖了其他图元的图元的组成部分。要执行相反的工作 （删除没有重叠图元的部分），可使用 **FeatureFactory.IntersectFeatures 方法**。



FeatureFactory.IntersectFeatures 方法

目的

此方法返回独立的图元，表示多个图元的交集。

语法

[Feature=]OBJECT.**IntersectFeatures**(feature1 [feature2])

部分	说明
OBJECT	FeatureFactory 对象
feature1	Feature 对象或 Features 集合对象，表示区域或线图元。
feature2	**Variant**：Feature 对象或 Features 集合对象，表示区域或线图元。只有在 feature1 参数是单一 Feature 对象时此参数为必需。

注解

此方法取 Feature 或 Features 对象和另一个 Feature 或 Features 对象的交集，返回的结果对象作为独立的图元。

您不能对点或文本图元使用此方法。只支持以下图元类型组合：

- 您可以取区域和其他区域的交集。如果区域重叠，则由此方法返回的图元也是区域。
- 您可以取区域和线的交集。假定区域至少覆盖了线的一部分，则交集是由该区域覆盖的线的一部分。

- 您可以取区域和其他线的交集。如果两线交叉，则 IntersectFeatures 将返回一个带有一个点的多义线图元。

请注意两个图元可能根本没有交集。如果是一个线和一个区域或两个线，则由此方法返回的值是带有 0 个点的 POLYLINE FEATURE。但是，当两个区域图元不相交，则返回的值是带有 0 个点的 REGION FEATURE。无论如何，“0 点图元”不能插入图层。

在使用多个多义线时，图元集合的交集返回所有线交叉处的点。因此，如果图元集合中的任意线都没有和其他对象相交，则返回带有 0 个点的多义线。

IntersectFeatures 方法返回重叠其他图元的图元的组成部分。要获取没有重叠的图元的组成部分，可使用 EraseFeature 方法。

FeatureFactory.IntersectionPoints 方法

目的

此方法返回 Points 集合，包含两个图元相交的点。

语法

[Points collection=]OBJECT.**IntersectionPoints**(feature1, feature2, [flag])

部分	说明
object	FeatureFactory 对象
feature1	一个 Feature 对象。
feature2	一个 Feature 对象。
flag	**Variant**：**IntersectionPointConstants** 值，如下所示。可选；如果省略，则默认为 miIntersectAll。

注解

如果两个图元不相交，则此方法返回的点集合为空。

flag 参数可以是三个 IntersectionPointConstants 值的任意一个，如下所示。

设置	值	说明
miIntersectCrossings	9	返回点集，其中一个图元的线段和另一个图元的线段交叉，但是图元在交叉点处没有节点。
miIntersectCommon	10	返回点集，其中一个图元的线段和另一个图元的线段交叉，两个图元在交叉点处都有节点。
miIntersectAll	11	返回图元交叉或交集的所有点，无论在交集点处是否有公共节点。换而言之，检索所有 miIntersectCommon 点和所有 miIntersectCrossings 点。

FeatureFactory.IntersectionTest 方法

目的

此方法返回布尔值，表示两个图元是否满足交集测试。

语法

[Boolean=]OBJECT.**IntersectionTest**(feature1, feature2, [flag])

部分	说明
OBJECT	FeatureFactory 对象
feature1	一个 Feature 对象。
feature2	一个 Feature 对象。
flag	**Variant**：**IntersectionTestConstant**s 值，如下所示。可选；如果省略，则默认为 miIntersectFeature。

注解

此方法测试两个图元，确定它们是否相交。此处有三种类型的交集测试， flag 参数控制执行哪类测试。

设置	值	说明
miIntersectCentroidWithinFeature	0	如果 feature1 质心位于 feature2 之内，则测试返回 True。
miIntersectFeature	1	如果两个图元在任意点相交，或者如果邻近的图元共享公共点，则测试返回 True。
miIntersectEntirelyWithinFeature	2	如果 feature1 完全位于 feature2 之内，则测试返回 True。

Field 对象和 Fields 集合

Fields 集合用于描述或定义数据表的列结构，并传递到 **Datasets.Add** 以通知 MapX 要将哪一源数据的字段包含在 Dataset 之内。如果并未指定 Fields 集合，则将包括源数据的所有字段。 Fields 集合还可以传递到**主题**。**添加**方法以指定要将主题基于源自现有数据集的哪些字段。如果没有指定 Fields 集合，Mapx 将使用第一个数字字段。Fields 集合还可以指定为 **LayerInfo object** 的参数之一，在添加图层类型为 miLayerInfoTypeTemp 或 miLayerInfoTypeNewTable 时，作为参数传递到 **Layers.Add 方法**。此时，Fields 集合指定正在创建的表的列结构。

对象属性

- **Field.AggregationFunction 属性**
- **Field.Decimals 属性**
- **Field.Indexed 属性**
- **Field.Name 属性**
- **Field.Type 属性**
- **Field.TypeEx 属性**

注：如果用户尝试获取无法用于该类型的 Field 对象的属性，则 MapX 将会报告错误 ERR_PROPERTY_NOT_AVAILABLE (1295)。

集合属性

- **Fields.Count 属性**
- **Fields.Item 属性**

集合方法

- **Fields.Add 方法**
- **Fields.AddDateField 方法**
- **Fields.AddFloatField**
- **Fields.AddIntegerField**
- **Fields.AddLogicalField 方法**
- **Fields.AddNumericField 方法**
- **Fields.AddSmallIntField 方法**
- **Fields.AddStringField 方法**
- **Fields.Count 属性**
- **Fields.Item 属性**
- **Fields.Remove 方法**
- **Fields.RemoveAll 方法**

另请参阅

LayerInfo 对象，第 373 页

Dataset 对象和 Datasets 集合，第 257 页

Field.AggregationFunction 属性

目的

此属性控制 MapX 如何在多个数据绑定到系统的图元时计算 Field 的值。这是一个只读属性，设置为 **AggregationFunctionConstants** 之一。它在 Fields.Add 方法中初始化。

另请参阅

Fields.Add 方法，第 305 页

Field.Decimals 属性

目的

这一只读属性用于指定字段对象中的小数个数的数值。

Field.Indexed 属性

目的

这是一个只读属性，如果底层表已索引则返回 true。

Field.Name 属性

目的

只读属性，指定字段名。这是字符串值。是 **Field** 对象的默认属性。

另请参阅

MapX 字段命名规范，第 507 页

Field.Precision 属性

目的

这一只读属性指定置于给定 Field 对象中的数字值的总个数。

Field.Width 属性

目的

只读属性，指定给定字段对象的字符串长度。

Field.Type 属性

目的

此属性返回字段类型，但是只适用于字符串、日期和数字类型的字符串类型。这是一个只读属性。

注：Field.TypeEx 属性已经添加到 MapX，以返回所有 **FieldTypeConstants**。

注解

下表说明在使用 Field.Type 和 Field.TypeEx 时返回哪些数据类型。

字段类型	MapX 返回使用 Field.Type	MapX 返回使用 Field.TypeEx
整数	数字	整数
字符	字符	字符
浮点	数字	浮点
逻辑	数字	逻辑
日期	日期	日期
小整数	数字	小整数
Decimal(4,2)	数字	数字

另请参阅

Field.TypeEx 属性，第 304 页

Field.TypeEx 属性

目的

此属性返回带有 **FieldTypeConstants** 值的字段类型。这是一个只读属性。

注：此属性是 MapX 的新属性。它实施用于纠正 Field.Type 属性的问题。 Field.Type 属性只能返回字符串、日期和数字类型。 Field.TypeEx 属性可以返回所有 FieldTypeConstants。

注解

下表说明在使用 Field.Type 和 Field.TypeEx 时返回哪些数据类型。

字段类型	MapX 返回使用 Field.Type	MapX 返回使用 Field.TypeEx
整数	数字	整数
字符	字符	字符
浮点	数字	浮点
逻辑	数字	逻辑

字段类型	MapX 返回使用 Field.Type	MapX 返回使用 Field.TypeEx
日期	日期	日期
小整数	数字	小整数
Decimal(4,2)	数字	数字

另请参阅

Field.Type 属性，第 304 页

Fields.Add 方法

目的

此方法将字段添加到 Fields 集合。

语法

[Field=]OBJECT**.Add** (DataSourceCol, [Name], [AggregateFunction], [Type])

部分	说明
OBJECT	表示 Fields 对象。
DataSourceCol	源自要使用的数据源的列。可以是列的名称或索引。
Name	要添加的字段名称。默认为数据源中的字段名。
AggregateFunction	要使用的聚集函数。值：**AggregationFunctionConstants**
Type	列中的数据类型。取值为 **FieldTypeConstants** 值。

注解

Fields 集合构建用于 Datasets.Add 方法。 Fields 参数取 Fields 集合，使用此 Add 方法构建。

Type 参数只适用于未绑定数据集。对于所有其他 Dataset 类型均忽略。

聚集函数默认为 miAggregationIndividual （对于字符串列）和 miAggregationSum （对于数字列）。

数据集创建后，不能对数据集的 Fields 集合使用 Add 方法。

集合中的所有字段名必须在给定的字段集合中唯一。

另请参阅

Datasets.Add 方法，第 263 页

Dataset.Fields 属性，第 259 页

MapX 字段命名规范，第 507 页

Fields.AddDateField 方法

目的

此方法向字段集合添加日期。

注：此方法专门在 Fields 集合被传递到 **Layers.Add** 的 **LayerInfo 对象**时使用。

语法

OBJECT.**AddDateField** (Name, [Indexed])

部分	说明
OBJECT	表示 Fields 集合。
Name	日期字段所用的名称。
Indexed	布尔值：表示是否应该在此字段创建索引。默认值为 false。

另请参阅

MapX 字段命名规范，第 507 页

Fields.AddFloatField

目的

此方法将浮点字段添加到 Fields 集合。

注：此方法专门在 Fields 集合被传递到 **Layers.Add** 的 **LayerInfo 对象**时使用。

语法

OBJECT.**AddFloatField** (Name, [Indexed])

部分	说明
OBJECT	表示 Fields 集合。
Name	浮点字段所用的名称。
Indexed	布尔值：表示是否应该在此字段创建索引。默认值为 false。

另请参阅

MapX 字段命名规范，第 507 页

Fields.AddIntegerField

目的

此方法将整数字段添加到 Fields 集合。

注：此方法专门在 Fields 集合被传递到 **Layers.Add** 的 **LayerInfo 对象**时使用。

语法

OBJECT.AddIntegerField (Name, [Indexed])

部分	说明
OBJECT	表示 Fields 集合。
Name	整数字段所用的名称。
Indexed	布尔值：表示是否应该在此字段创建索引。默认值为 false。

另请参阅

MapX 字段命名规范，第 507 页

Fields.AddLogicalField 方法

目的

此方法将逻辑字段添加到 Fields 集合。

注：此方法专门在 Fields 集合被传递到 **Layers.Add** 的 **LayerInfo 对象**时使用。

语法

OBJECT.**AddLogicalField** (Name, [Indexed])

部分	说明
OBJECT	表示 Fields 集合。
Name	逻辑字段所用的名称。
Indexed	布尔值：表示是否应该在此字段创建索引。默认值为 false。

另请参阅

MapX 字段命名规范，第 507 页

Fields.AddNumericField 方法

目的

此方法将数字字段添加到 Fields 集合。

注：此方法专门在 Fields 集合被传递到 **Layers.Add** 的 **LayerInfo 对象**时使用。

语法

OBJECT.**AddNumericField**(Name, Precision, Decimals, [Indexed])

部分	说明
OBJECT	表示 Fields 集合。
Name	数字字段所用的名称。
Precision	数字的总计位数。
Decimals	小数的总计位数。
Indexed	布尔值：表示是否应该在此字段创建索引。默认值为 false。

另请参阅

MapX 字段命名规范，第 507 页

Fields.AddSmallIntField 方法

目的

此方法向字段集合添加小整数。

注：此方法专门在 Fields 集合被传递到 **Layers.Add** 的 **LayerInfo 对象**时使用。

语法

OBJECT.**AddSmallInt**(Name, [Indexed])

部分	说明
OBJECT	表示 Fields 集合。
Name	小整数字段所用的名称。
Indexed	布尔值：表示是否应该在此字段创建索引。默认值为 false。

另请参阅

MapX 字段命名规范，第 507 页

Fields.AddStringField 方法 Purpose

此方法将字符串字段添加到 Fields 集合。

注：此方法专门在 Fields 集合被传递到 **Layers.Add** 的 **LayerInfo 对象**时使用。

语法

OBJECT.AddStringField(Name, Width, [Indexed])

部分	说明
OBJECT	表示 Fields 集合。
Name	字符串字段所用的名称。
Width	字符串的指定长度。有效值必须介于 1 和 254 之间。
Indexed	布尔值：表示是否应该在此字段创建索引。默认值为 false。

另请参阅

MapX 字段命名规范，第 507 页

Fields.Count 属性

目的

Fields 集合中的字段数。这是一个只读属性。

Fields.Item 属性

目的

从 Field 集合检索 Fields 对象。采用参数指定具体字段。参数可以指定字段名或索引编号 （基于 1 的索引）。这是 Fields 集合的默认属性。

另请参阅

MapX 字段命名规范，第 507 页

Fields.Remove 方法

目的

从 Fields 集合中删除指定的 Field 对象。

注：如果您去除项目，集合索引将再次计数，填充由已删除项目留下的空洞。

语法

OBJECT.**Remove** (index)

部分	说明
OBJECT	表示 Fields 对象。
index	Field 对象名称或基于 1 的索引编号。

Fields.RemoveAll 方法

目的

将所有 Field 对象从集合中删除。

语法

OBJECT.**RemoveAll**

Find 对象

目的

Find 对象允许您输入图元的名称， MapX 会查找它。您可以通过使用第二个多边形表来进行优化，并使用算法在街道和街道两侧插入新的位置。

对象属性

- **Find.Abbreviations 属性**
- **Find.CloseMatchMax 属性**
- **Find.ClosestAddr 属性**
- **Find.FindDataset 属性**
- **Find.FindField 属性**
- **Find.OtherBoundary 属性**
- **Find.RefineDataset 属性**
- **Find.RefineField 属性**
- **Find.RefineLayer 属性**

对象方法

- **Find.Search 方法**
- **Find.SearchEx 方法**

注：以上方法不适用于远程数据库图层。将 **Layer.Search 方法**用作在远程数据库图层上确定图元或图元集合位置的另一种方法。

Find.Abbreviations 属性

目的

这是一个读写布尔值属性，默认值为 FALSE。采用 FALSE 时， MapX 不会引用缩写文件。如果设置为 TRUE， MapX 将在它搜索表中记录时，无法发现完全匹配地址，将会使用缩写文件 (mapx.abb)。

注解

缩写文件包含众多类似 "STREET ST" 和 "AVE AV" 的成对项目。如果要搜索的地址包含 "STREET"，则 MapX 将其更改为 "ST" 并在此进行搜索。

Find.CloseMatchMax 属性

目的

此属性确定匹配可能包含的 FindResult 对象的最大数量。即在通过 **Find.SearchEx method** 执行查找时，如果未发现完全匹配项，则 SearchEx 将尽其所能返回众多近似的匹配项，最多为此属性指定的值。此属性的默认值为 8，并且必须采用 >= 0 的整数值，否则 MapX 将会报错。

Find.ClosestAddr 属性

目的

如果没有找到指定编号，则此布尔值属性设置为 TRUE 时将匹配最近似的门牌号。例如，如果用户请求 120 ELM ST，但 MapX 只知道 1-100 ELM ST，则将令 MapX 匹配 100 ELM ST。默认值为 FALSE。

Find.FindDataset 属性

目的

要匹配的字段数据集。如果未指定，则使用图层的 KeyField。

Find.FindField 属性

目的

要匹配的数据集字段。和 FindDataset 属性一起使用。

另请参阅

Find.FindDataset 属性，第 312 页

Find.SearchEx 方法

目的

SearchEx 方法扩展了搜索方法的功能。它令搜索引擎可以查找最近似的匹配项并返回集合。开发人员可以在列表框中将此展示给用户。最初搜索将尝试使用方法的给定参数返回完全匹配项。如果没有找到完全匹配项，则 MatchedFeature 将为空，并且 FindMatches 集合将包含发现的所有近似的 FindMatch 项。

注：此方法不适用于远程数据库图层。**Layer.Search 方法**可用作在远程数据库图层上确定图元或图元集合位置的另一种方法。

语法

[FindResult=]OBJECT.**SearchEx** (Address, [Boundary])

部分	说明
OBJECT	表示 Find 对象。
Address	字符串，是要查找的对象或街道地址的名称。例如 “Rensselaer, 6 Georgian Ct,”, “London”, 或 “3 Elm Street”。

部分	说明
Boundary	字符串，是优化边界对象的名称。在指定优化图层时使用。例如 “NY”, “12211” 。 如果设置优化图层，则必需使用此参数，否则为可选。

另请参阅

FindResult 对象，第 319 页

FindMatch 对象和 FindMatches 集合，第 317 页

Find.OtherBoundary 属性

目的

这一布尔值属性在使用优化边界时应用，控制 MapX 如何处理优化边界中缺少匹配项的情况。

如果为 False，则 Find.Search 在查找对象处于优化边界中时返回匹配项，否则不返回匹配项。如果为 True，则 Find.Search 在查找对象处于优化边界中时返回匹配项，否则将返回优化边界之外的匹配项。默认值为 false。

另请参阅

Find.RefineLayer 属性

Find.Search 方法

Find.RefineDataset 属性

目的

指定 RefineLayer 之后，可以使用 geocolumn 来匹配，或者可以通过指定要用作替代的 Dataset 和 Field 来进行覆盖。

Find.RefineField 属性

目的

用于和 RefineDataset 一起使用。

Find.RefineLayer 属性

目的

要优化的图层。如果可能有多个匹配项，则优化非常实用。

例如，在街道地图中可能有多个 BROADWAY － 每个邮政编码一个。您可以使用邮政编码来进行优化。在发现之后，可指定街道和邮政编码。

另请参阅

Find.RefineField 属性，第 313 页

Find.RefineDataset 属性，第 313 页

Find.Search 方法

目的

搜索图层以查找街道地址或地图上的图元的名称。它将返回 FindFeature 对象。

注：以上方法不适用于远程数据库图层。 **Layer.Search 方法**可用作在远程数据库图层上确定图元或图元集合位置的另一种方法。

语法

OBJECT.**Search** (Address, [Boundary])

部分	说明
OBJECT	Find 对象
Address	字符串，是要查找的对象或街道地址的名称。例如 “Rensselaer, 6 Georgian Ct,”, “London”, 或 “3 Elm Street”。
Boundary	字符串，是优化边界对象的名称。在指定优化图层时使用。例如 “NY”, “21135”。 如果设置优化图层，则必需使用此参数，否则为可选。

另请参阅

Find 对象，第 311 页

FindFeature 对象，第 315 页

FindFeature 对象

FindFeature 对象是带有附加结果集 (FindRC) 的 Feature 的超类。

FindFeature 通过 Find 对象的搜索方法返回，包含 Feature 对象的所有属性和 FindRC 属性。

对象属性

- **FindFeature.FindRC 属性**

另请参阅

Feature 对象和 Features 集合，第 271 页 Feature 对象

第 125 页的第 9 章：“查找地图上的图元”

FindFeature.FindRC 属性

目的

FindRC 是查找操作的结果代码。

要获取 X,Y 和 / 或对象 ID，可使用属性 CenterX、 CenterY、 FeatureID （请参阅 Feature 对象）。如果发现地址，则 FeatureID 将为 0。

数字值	含义
** 一个地点 **	
xx1	完全匹配
xx2	所用缩写文件中的替代对象
xx3 (-)	未找到完全匹配
xx4 (-)	未指定对象名，未找到匹配项
** 十个地点 **	
x1x	未确定是街道的哪一侧
x2x （+ / -)	地址编号超出限定范围
x3x （+ / -)	地址编号不在限定范围之内
x4x （+ / -)	未指定地址编号
x5x （-)	街道不相交
x6x （-)	匹配行没有地图对象
** 百个地点 **	
1xx (+ / -)	只在不是指定区域的一个区域中发现名称
2xx (-)	只在不是指定区域的多个区域中发现名称

数字值	含义
3xx (+ / -)	没有指定优化区域，发现一个匹配项
4xx (-)	没有指定区域，发现多个匹配项
5xx (+)	在指定区域多次发现名称

另请参阅

FindFeature 对象，第 315 页

FindMatch 对象和 FindMatches 集合

FindMatch 对象返回从 SearchEx 方法定义的近似匹配项或完全匹配项返回的匹配项。 FindMatches 集合按照得分次序排名，最高得分排在首位。

对象属性

- **FindMatch.FeatureID 属性**
- **FindMatch.FeatureKey 属性**
- **FindMatch.Name 属性**

集合属性

- **FindMatches.Count 属性**
- **FindMatches.Item 属性**

另请参阅

FindResult 对象，第 319 页

Find.SearchEx 方法，第 312 页

FindMatch.Name 属性

目的

此属性返回对象或街道名称。

FindMatch.FeatureID 属性

目的

此属性返回图元 （如已存在） ID。如果图元不存在，则返回 0。

FindMatch.FeatureKey 属性

目的

此属性返回图元 （如已存在） FeatureKey 。如果图元不存在，则返回空字符串。

FindMatch.Score 属性

目的

此属性返回近似匹配的匹配分值，不匹配为 0，完全匹配为 100。

FindMatches.Count 属性

目的

此只读属性表示集合中的匹配项数。

FindMatches.Item 属性

目的

此属性返回 FindMatches 集合中的项。相应的项按照其在 FindMatches 集合中的分值排序 （高分在前）。

语法

[FindMatch=]OBJECT.**Item** (index)

部分	说明
OBJECT	表示 FindMatches 集合
index	返回项的整数索引。

FindResult 对象

FindResult 对象返回有关 FindRC 的信息，采用属性的形式，以便可以轻松访问查找的结果。 FindResult 包含图元集合，后者包含多个匹配的图元或最接近的匹配图元，按照其在查找中的分值高低先后排列。如果发现完全匹配项，则图元集合将只包含 ExactMatch 图元。

对象属性

- **FindResult.AddressOutOfRange 属性**
- **FindResult.ExactMatch 属性**
- **FindResult.FindRC 属性**
- **FindResult.IntersectionNotFound 属性**
- **FindResult.MatchedFeature 属性**
- **FindResult.Matches 属性**
- **FindResult.MultipleMatches 属性**
- **FindResult.RefineRegion 属性**
- **FindResult.Substitute 属性**

另请参阅

Find.SearchEx 方法，第 312 页

FindResult.AddressOutOfRange 属性

目的

如果地址编号位于范围之外，此属性返回布尔值为 TRUE。

FindResult.ExactMatch 属性

目的

如果发现完全匹配项，此属性返回布尔值为 TRUE。

另请参阅

FindResult.MatchedFeature 属性，第 320 页

FindResult.Matches 属性，第 320 页

FindResult.MultipleMatches 属性，第 320 页

FindResult.FindRC 属性

目的

此属性搜索图层以查找街道地址或地图上的图元名称，返回 **FindFeature 对象**。如果返回负数，则表示没有发现完全匹配项。

FindResult.IntersectionNotFound 属性

目的

这一只读属性返回布尔值为 TRUE，表示街道是否相交。

FindResult.MatchedFeature 属性

目的

此属性返回从 Find.SearchEx 生成的 ExactMatch 图元。

另请参阅

Find.SearchEx 方法，第 312 页

FindResult.Matches 属性

目的

此属性返回从 Find.SearchEx 生成的 Match 集合。

另请参阅

Find.SearchEx 方法，第 312 页

FindResult.MultipleMatches 属性

目的

如果在执行 Find.SearchEx 之后发现多个匹配项，则此属性返回布尔值为 TRUE。

另请参阅

Find.SearchEx 方法，第 312 页

FindResult.RefineRegion 属性

目的

如果指定优化区域，则此属性返回布尔值为 TRUE。

FindResult.Substitute 属性

目的

如果使用缩写文件中的替代对象，则此属性返回布尔值为 TRUE。

Geoset 对象和 Geoset 集合

Geoset 对象构建自 Map 对象，允许您定义 geoset。Geoset 是图层及其设置的集合。Geosets 集合对象是 Geoset 对象的集合

对象属性

- **Geoset.Centroid 属性**
- **Geoset.PathName 属性**
- **Geoset.UserName 属性**

集合属性

- **Geosets.Count 属性**
- **Geosets.Item 属性**

另请参阅

Map 对象，第 388 页

Geoset.Centroid 属性

目的

此属性指定定义 geoset 地理中心的 Point 对象。

另请参阅

Point 对象和 Points 集合，第 429 页

Geoset.PathName 属性

目的

此属性指定定义 geoset 物理位置的 Path。

Geoset.UserName 属性

目的

指定定义 geoset 所用的 Name。

Geosets.Count 属性

目的

此属性指定集合中的 Geoset 对象数量。这是一个只读属性。

Geosets.Item 属性

目的

此属性从集合检索特定的 Geoset 对象。这是一个变量，您可以指定 Geoset 的用户名或基于 1 的索引编号。这是 Geosets 集合的默认属性。

Graphic 对象

目的

每个 Annotation 均包含 Graphic 对象 （Annotation.Graphic 属性），其中存储符号或文本和位置。

对象属性

- **Graphic.Caption 属性**
- **Graphic.Position 属性**
- **Graphic.Style 属性**
- **Graphic.X 属性**
- **Graphic.Y 属性**

另请参阅

Annotation 对象和 Annotations 集合，第 238 页

Graphic.Caption 属性

目的

如果注释类型为 miTextAnnotation，则包含文本字符串。这是字符串值。

另请参阅

Annotation.Type 属性，第 239 页

AnnotationTypeConstants，第 558 页

Graphic.Position 属性

目的

此属性包含表示文本如何根据 X,Y 坐标绘制的值。只适用于 miTextAnnotation 注释类型。取值为 **PositionConstants**，默认值为 miPositionTL。

另请参阅

Annotation.Type 属性，第 239 页

Graphic.Style 属性

目的

Style 对象包含符号或文本的样式。这是一个读写属性。

另请参阅

Style 对象，第 463 页

Annotation.Type 属性，第 239 页

Graphic.X 属性

目的

包含注释的 X 坐标。这是一个双精度值，表示地球坐标 （经度）。如果注释类型为 miTextAnnotation，则注释位置由与 Position 关联的 X,Y 坐标控制。如果注释类型为 miSymbolAnnotation，则 X,Y 坐标指符号的中心。

另请参阅

Annotation.Type 属性，第 239 页

AnnotationTypeConstants，第 558 页

Graphic.Position 属性，第 324 页

Graphic.Y 属性

目的

包含注释的 Y 坐标。这是一个双精度值，表示地球坐标 （纬度）。如果注释类型为 miTextAnnotation，则注释位置由与 Position 关联的 X,Y 坐标控制。如果注释类型为 miSymbolAnnotation，则 X,Y 坐标指符号的中心。

另请参阅

Annotation.Type 属性，第 239 页

AnnotationTypeConstants，第 558 页

Graphic.Position 属性，第 324 页

IndividualValueCategory 对象和 IndividualValueCategories 集合

个别值主题地图的设置通过 IndividualValueCategories 集合显示，这是 IndividualValueCategory 对象的集合，主题中每个唯一值一个对象。

要获取 IndividualValueCategories 集合，应引用 ThemeProperties.IndividualValueCategories 属性。

对象属性

- **IndividualValueCategory.NumItems 属性**
- **IndividualValueCategory.Style 属性**
- **IndividualValueCategory.Value 属性**

集合属性

- **IndividualValueCategories.AllOthersCategory 属性**
- **IndividualValueCategories.Count 属性**
- **IndividualValueCategories.Item 属性**

另请参阅

Legend 对象，第 379 页

ThemeProperties.IndividualValueCategories 属性，第 491 页

IndividualValueCategories.AllOthersCategory 属性

此属性从集合返回一个 IndividualValueCategory 对象，此对象说明所有不在单值主题中的值范围的类别。返回的 IndividualValueCategory 对象带有 AllOthersCategory 属性，具有未定义的 Value 属性，其默认值为空字符串 (“”)。但是，您可以设置其 Style 属性，如下例所示。在用于 AllOthersCategory 范围类别对象出现的文本可使用 **LegendTexts.AllOthersText** 属性设置。

IndividualValueCategories.Count 属性

目的

这是一个只读整数值，表示主题中的唯一单值的数量。

IndividualValueCategories.Item 属性

目的

此属性从集合返回一个 IndividualValueCategory 对象，此对象说明单值主题中的一个唯一值。

IndividualValueCategory.NumItems 属性

目的

NumItems 是在此类别之中的图元数量。这是一个整数值，是只读属性。

注：此属性只在 MapX 生成该主题时有效 （Theme.ComputeTheme = TRUE）。如果您手动计算主题 （Theme.ComputeTheme = FALSE），则 NumItems 将不会反应类别中的项数，并且不定位为结果。

IndividualValueCategory.Style 属性

目的

此属性控制类别的样式。这是一个 Style 对象，并且可以设置为现有样式对象，或者您可以分别设置样式属性。默认值设置用于提供高对比样式。

另请参阅

Style 对象，第 463 页

IndividualValueCategory.Value 属性

目的

此属性取或设置单值主题中的单值类别的值。该属性只有在 Theme.ComputeTheme 属性设置为 FALSE 时可以设置。

注：设置 Value 属性不会影响主题的 Legend.LegendTexts 集合，因为后者需要单独更新。

语法

IndividualValueCategory.Value 属性，第 327 页

Label 对象和 Labels 集合

地图中的每个图层都有一个 Labels 集合。集合中的每个项目都是一个 Label 对象。图层的 Labels 集合包含该图层中在地图窗口或缓存中可见的所有标签，以及来自已经通过手动 （使用普通选择工具选择或移动，或使用普通标签工具添加）或程序 （使用 Layer 对象和 Layers 集合的方法和属性）编辑过的图层的任意标签。

通过编辑 Label 对象的属性，用户可以更改该图层的其他标签来独立更改标签的位置或样式。

Label 对象属性

- **Label.AnchorX 属性**
- **Label.AnchorY 属性**
- **Label.Angle 属性**
- **Label.Caption 属性**
- **Label.Drawn 属性**
- **Label.Edited 属性**
- **Label.EditedAnchor 属性**
- **Label.EditedAngle 属性**
- **Label.EditedCaption 属性**
- **Label.EditedLineStyle 属性**
- **Label.EditedLineType 属性**
- **Label.EditedMultiSegment 属性**
- **Label.EditedOffset 属性**
- **Label.EditedPosition 属性**
- **Label.EditedTextStyle 属性**
- **Label.EditedVisibile 属性**
- **Label.FeatureKey**
- **Label.Layer 属性**
- **Label.LineType 属性**
- **Label.MultiSegment 属性**
- **Label.Offset 属性**
- **Label.Position 属性**
- **Label.Selected 属性**
- **Label.Style 属性**
- **Label.Visible 属性**
- **Labels.Count 属性**
- **Labels.Item 属性**

另请参阅

Layer.Labels 属性，第 354 页

LabelProperties.LabelAlong 属性，第 336 页

LabelProperties.LabelPartialObjects 属性，第 337 页

Label.AnchorX 属性

目的

此读写属性返回标记锚点位置的 X 坐标。这是一个双精度值，表示经度 （负值代表西半球）。

Label.AnchorY 属性

目的

此读写属性返回标记锚点位置的 Y 坐标。这是一个双精度值，表示纬度。

Label.Angle 属性

目的

此读写属性是标签的逆时针旋转角度。如果标签为多段，则其不适用。

Label.Caption 属性

目的

此读写属性取标签的文本。

Label.Drawn 属性

目的

这是一个只读属性，如果标签在当前视图中绘制，则设置为 true。

Label.Edited 属性

目的

这是一个只读属性，如果标签已使用某种方式定制 （通过程序或使用 SelectPoint 工具），则设置为 true。否则设置为 False。

Label.EditedAnchor 属性

目的

这是一个只读属性，如果标签锚点已使用某种方式定制（通过程序或使用 SelectPoint 工具），则设置为 True。否则设置为 False。

另请参阅

Label.AnchorX 属性，第 329 页

Label.AnchorY 属性，第 329 页

Label.EditedAngle 属性

目的

这是一个只读属性，如果标签的角度已通过编程定制，则设置为 True。否则设置为 False。

另请参阅

Label.Angle 属性，第 329 页

Label.EditedCaption 属性

目的

这是一个只读属性，如果标签的说明已通过编程定制，则设置为 True。否则设置为 False。

另请参阅

Label.Caption 属性，第 329 页

Label.EditedLineStyle 属性

目的

这是一个只读属性，如果标签的标注线样式已通过编程定制，则设置为 True。否则设置为 False。

另请参阅

Label.Style 属性，第 333 页

Label.EditedLineType 属性

目的

这是一个只读属性，如果标签的标注线类型已通过编程定制，则设置为 True。否则设置为 False。

另请参阅

Label.LineType 属性，第 332 页

Label.EditedMultiSegment 属性

目的

这是一个只读属性，如果标签的多段状态已经更改 （通过程序或使用 SelectPoint 工具），则设置为 True。否则设置为 False。

另请参阅

Label.MultiSegment 属性，第 332 页

Label.EditedOffset 属性

目的

这是一个只读属性，如果标签补偿量已使用某种方式定制 （通过程序或使用 SelectPoint 工具），则设置为 True。否则设置为 False。

另请参阅

Label.Offset 属性，第 333 页

Label.EditedPosition 属性

目的

这是一个只读属性，如果标签位置已使用某种方式定制（通过程序或使用 SelectPoint 工具），则设置为 True。否则设置为 False。

另请参阅

Label.Position 属性，第 333 页

Label.EditedTextStyle 属性

目的

这是一个只读属性，如果标签的文本样式已通过编程定制，则设置为 True。否则设置为 False。

Label.EditedVisibile 属性

目的

这是一个只读属性，如果标签的可见性已使用某种方式定制 （通过程序或使用 SelectPoint 工具），则设置为 True。否则设置为 False。

另请参阅

Label.Visible 属性目的，第 333 页

Label.FeatureKey

目的

此只读属性返回和标签关联的图元的关键字。使用此关键字访问带有 Layer.GetFeatureByKey 的图元方法。

另请参阅

Layer.GetFeatureByKey 方法，第 352 页

Label.Layer 属性

目的

此只读属性将图层设置为标签所属的图层。

另请参阅

Layer 对象和 Layers 集合，第 340 页

Label.LineType 属性

目的

此读写属性取标签的标准线类型 （miLineTypeArrow、 miLineTypeNone 或 miLineTypeSimple）。

另请参阅

LineTypeConstants，第 569 页

Label.MultiSegment 属性

目的

此读写属性取标签的多段状态。如果标签为多段标签，则其为 True，否则为 False。

Label.Offset 属性

目的

此读写属性取距离 （像素），标签自其锚点位置补偿。

另请参阅

LabelProperties.Offset 属性，第 338 页

Label.Position 属性

目的

此读写属性取标签相对于其锚点位置的位置。取值为 PositionConstants 值。

另请参阅

PositionConstants，第 571 页

Label.Selected 属性

目的

这一读写属性表示是否选择相应标签。（读写属性）。

Label.Style 属性

目的

此读写属性取标签文本的样式。

另请参阅

Style 对象，第 463 页

Label.Visible 属性目的

这一读写属性表示是否隐藏相应标签。隐藏标签表示标签是否符合自动绘制的要求，如果标记为隐藏，则不会绘制。

Labels.Count 属性

目的

此只读属性统计标签集合中的标签数。

Labels.Item 属性

目的

此读写属性从标签集合取特定标签。该索引是从 1 开始的整数索引。这是用于 Labels 集合的默认属性。

LabelProperties 对象

LabelProperties 对象包含控制如何为图层绘制标签的属性。

对象属性

- **LabelProperties.DataField 属性**
- **LabelProperties.Dataset 属性**
- **LabelProperties.Duplicate 属性**
- **LabelProperties.LabelAlong 属性**
- **LabelProperties.LabelMax 属性**
- **LabelProperties.LabelPartialObjects 属性**
- **LabelProperties.LabelZoom 属性**
- **LabelProperties.LabelZoomMax 属性**
- **LabelProperties.LabelZoomMin 属性**
- **LabelProperties.LineType 属性**
- **LabelProperties.Offset 属性**
- **LabelProperties.Overlap 属性**
- **LabelProperties.Parallel 属性**
- **LabelProperties.PartialSegments 属性**
- **LabelProperties.Position 属性**
- **LabelProperties.Style 属性**
- **LabelProperties.Visible 属性**

另请参阅

Layer 对象和 Layers 集合，第 340 页

LabelProperties.DataField 属性

目的

此属性设置用于标签的位置 （例如 DataField）。该字段用于标记，由此属性和 Dataset 属性指定，表示要使用的 Dataset。取值为 Field 对象。

另请参阅

LabelProperties.Dataset 属性，第 336 页

Field 对象和 Fields 集合，第 302 页

LabelProperties.Dataset 属性

目的

此属性将 Dataset 设置为用于标签。用于标记的字段由此属性和 DataField 属性指定，表示要使用 Dataset 中的哪一字段。不指定 Dataset 将导致图元使用图元名称标记。这是默认行为。取 Datasets 对象。

另请参阅

LabelProperties.DataField 属性，第 335 页

Dataset 对象和 Datasets 集合，第 257 页

LabelProperties.Duplicate 属性

目的

控制同名图元是否可以在地图上同时具有标签。该属性是一个布尔值，默认值为 False。

LabelProperties.LabelAlong 属性

目的

现在，如果所要标注的对象为多义线，则使用 LabelAlong 属性就可表明该标注将沿被标注对象的轮廓绘制。 LabelAlong 有三种状态，使用 LabelAlongConstants 表示。

注解

LabelAlong 属性取代了原有的 Parallel 属性。虽然我们仍对 Parallel 属性提供支持，但已将其降格为隐藏属性。以前，由 Parallel 属性来决定是否让线条或多义线上的标注进行旋转，或者绘制这些标注时所使用的角度是否与其所锚定的线段或多义线段的绘制角度相同。

对于除多义线之外的其他对象 （例如，点对象和区域对象），如果将属性设置为 miLabelAlongMultiSegment，则该属性将被忽略。对于线条对象，如果将属性设置为 miLabelAlongMultiSegment，就将绘制自动标注，其效果与将属性设置为 miLabelAlongParallel 相同。

另请参阅

LabelAlongConstants

LabelProperties.LabelMax 属性

目的

控制自动为图层生成的最大标签数。该属性是一个整数值，默认为 100。

LabelProperties.LabelPartialObjects 属性

目的

将此值设置为真将可令 MapX 标记地图对象，即使地图对象当前只有一部分可见。但是此属性不适用于点对象。该属性是一个布尔值，默认值为 False。

注解

即使对象的质心不在地图视图之内，此属性允许标签自动显示。

LabelProperties.LabelZoom 属性

目的

控制标签是否为缩放图层。缩放图层控制标签显示的缩放级别范围 （地图上的距离）。如果 LabelZooming 为启用，则值存储在 LabelZoomMax 之中，并且将使用 LabelZoomMin 属性。该属性是一个布尔值，默认值为 False。

另请参阅

LabelProperties.LabelZoomMax 属性，第 337 页

LabelProperties.LabelZoomMin 属性，第 337 页

LabelProperties.LabelZoomMax 属性

目的

如果标签 ZoomLayering 为 （LabelProperties.LabelZoom 属性）启用，则此属性指定显示标签的最大缩放值。取双精度值，以地图单位指定距离 (Map.MapUnit)。如果缩放为 >= 最小值并且 < 最大值，则显示标签。

另请参阅

LabelProperties.LabelZoom 属性，第 337 页

Map.MapUnit 属性，第 405 页

LabelProperties.LabelZoomMin 属性

目的

如果标签 ZoomLayering 为 （LabelProperties.LabelZoom 属性）启用，则此属性指定显示标签的最小缩放值。取双精度值，以地图单位指定距离 (Map.MapUnit)。如果缩放为 >= 最小值并且 < 最大值，则显示标签。

另请参阅

LabelProperties.LabelZoom 属性，第 337 页

Map.MapUnit 属性，第 405 页

LabelProperties.LineType 属性

目的

控制将标签连接到要标记的图元所使用的线形。在偏移量较大时使用。取 **LineTypeConstants** 值，默认为 miLineTypeArrow。

另请参阅

LabelProperties.Offset 属性，第 338 页

LabelProperties.Offset 属性

目的

此属性指定了标签和要标记的图元之间的距离。在希望标签远离符号特定距离时，对于符号图元尤为实用。该属性是一个整数值，以像素指定距离。默认值为 2。

注：如果 LableProperties.Position 属性设置为 miPositionCC （中心对中心），然后 LableProperties.Offset 属性将被忽略。

另请参阅

LabelProperties.Position 属性，第 339 页

LabelProperties.LineType 属性，第 338 页

LabelProperties.Overlap 属性

目的

此属性允许标签在绘制到图层上时相互重叠。该属性是一个布尔值，默认值为 False。

注解

LabelProperties.Overlap 只用于自动标签。对于手动标记的图元，重叠将被忽略。

LabelProperties.Parallel 属性

注：这是一个 “不赞成使用”的属性，目前仍然可以工作，确保向后的兼容性。请使用 LabelProperties.LabelAlong 属性，第 336 页 取而代之

LabelProperties.PartialSegments 属性

注：这是一个 “不赞成使用”的属性，目前仍然可以工作，确保向后的兼容性。请使用 LabelProperties.LabelPartialObjects 属性，第 337 页 取而代之。

LabelProperties.Position 属性

目的

此属性设置标签应该相对于地图图元如何放置。取值为 **PositionConstants** 值。

默认值取决于 PredominantObjectType：

- miFeatureTypeSymbol is miPositionCR
- miFeatureTypeLine is miPositionBC
- miFeatureTypeRegion is miPositionCC

LabelProperties.Style 属性

目的

此属性设置用于显示标签的样式。默认值为 10 磅 Helvetica。

LabelProperties.Visible 属性

目的

控制标签是否对图层可见。标签可使用 Layer.Autolabel 属性创建，或者使用 Layer.LabelAtPoint 方法或使用标签工具手动创建。 Visible 属性控制所有标签是否对图层可见。该属性取布尔值，默认值为 True。

Layer 对象和 Layers 集合

每个地图都具有一个图层的集合。Layer 集合由 Layer 对象构成。Layer 集合具有用于从集合添加或移除 Layer 对象的方法和属性。

计算机地图被组织成多个图层。将图层视作在顶部彼此堆叠的透明层。每个图层包含整个地图的不同方面。每一地图图层都包含不同的地图对象，例如区域、点、线条和文本。

例如，一个图层可以包含州边界，第二个图层可以具有表示首都的符号，第三个图层可以包含文本标签。通过将这些图层堆叠到其它图层的顶部，就可以开始构建完整的地图。一次可以显示多个图层。地图图层构成 MapX 中的地图构建模块。创建了自己的图层之后，就可以使用多种方式定制图层，添加、删除图层或对它们进行重新排序。

您可以通过使用 Layers 集合、 Layer 对象和 LabelProperties 对象，设置以下图层选项。

图层重新排序

图层使用特定顺序显示。正确设定图层顺序非常重要。例如，假定有一个客户点的图层和一个人口地域图层。如果图层在地图窗口上的顺序不正确， MapX 将先绘制客户点，然后再显示人口地域图层。您的点将由于人口地域图层而变得模糊。

缩放图层

有时需要只显示特定缩放级别的图层。只有地图的缩放级别在预设距离之中时，缩放图层才控制图层的显示。

例如，您有一个街道图层和一个邮政编码边界图层。当您缩小超过 10 英里时，窗口中的街道看起来象黑色脏块。这是由于缩放 （窗口宽度）太宽，无法显示详细的街道地图。只有缩放设置为允许您正确看到街道细节的水平，例如，小于 5 英里，使用缩放图层才提示 MapX 显示街道图层。

第一个地图没有为其街道图层设置缩放图层。在 15 英里跨距的缩放级别上，难于看到细节。第二张地图的缩放图层在缩放小于 5 英里时显示街道。因此，在窗口设置为 15 英里时，不显示街道图层。

您可以为每个图层设置不同的缩放图层级别。

对象属性

- **Layer.AutoLabel 属性**
- **Layer.Bounds 属性**
- **Layer.ClippedBounds 属性**
- **Layer.CoordSys 属性**
- **Layer.DrawLabelsAfter 属性**
- **Layer.Editable 属性**
- **Layer.FileSpec 属性**
- **Layer.Find 属性**
- **Layer.KeyField 属性**
- **Layer.LabelProperties 属性**
- **Layer.Name 属性**

- **Layer.OverrideStyle 属性**
- **Layer.PredominantFeatureType 属性**
- **Layer.Selectable 属性**
- **Layer.Selection 属性**
- **Layer.ShowCentroids 属性**
- **Layer.ShowLineDirection 属性**
- **Layer.ShowNodes 属性**
- **Layer.Style 属性**
- **Layer.Type 属性**
- **Layer.Visible 属性**
- **Layer.ZoomLayer 属性**
- **Layer.ZoomMax 属性**
- **Layer.ZoomMin 属性**

对象方法

- **Layer.AddFeature 方法**
- **Layer.AllFeatures 方法**
- **Layer.BeginAccess 方法**
- **Layer.ClearCustomLabels 方法**
- **Layer.DeleteFeature 方法**
- **Layer.DrilldownAddFeatures 方法**
- **Layer.DrilldownRemoveFeatures 方法**
- **Layer.DrilldownReset 方法**
- **Layer.FeatureIDFromFeatureName 方法**
- **Layer.FeatureKeyFromFeatureName 方法**
- **Layer.GetDrilldownFeaturesByID 方法**
- **Layer.GetFeatureByID 方法**
- **Layer.GetFeatureByKey 方法**
- **Layer.Invalidate 方法**
- **Layer.LabelAtPoint 方法**
- **Layer.NoFeatures 方法**
- **Layer.Refresh 方法**
- **Layer.Search 方法**
- **Layer.SearchAtPoint 方法**
- **Layer.SearchWithinDistance 方法**
- **Layer.SearchWithinFeature 方法**
- **Layer.SearchWithinRectangle 方法**
- **Layer.UpdateFeature 方法**

集合属性

- **Layers.AnimationLayer 属性**

- **Layers.Bounds 方法**
- **Layers.ClippedBounds 属性**
- **Layers.Count 属性**
- **Layers.Item 属性**

集合方法

- **Layers.Remove 方法**
- **Layers.Add 方法**
- **Layers.AddGeosetLayers 方法**
- **Layers.AddServerLayer 方法**
- **Layers.AddUserDrawLayer 方法**
- **Layers.ClearSelection 方法**
- **Layers.CreateLayer 方法**
- **Layer.InsertionLayer**
- **Layers.LayersDlg 方法**
- **Layers.Move 方法**
- **Layers.RemoveAll 方法**

另请参阅

Map 对象，第 388 页

Layer.AddFeature 方法

目的

此方法在图层中创建并返回的新图元，带有 Source 对象图元的属性。此方法对于 Object 编辑非常实用。

语法

[feature=]OBJECT.**AddFeature** (Source, [RowValues])

部分	说明
OBJECT	表示 Layer 对象。
Source	Source 是 Feature 对象。
RowValues	RowValues 表示一行数据的属性数据新值。 **RowValues 集合**中的每个值均相应于属性数据的一列。此参数只适用于 miDatasetLayer 类型的 Dataset。

另请参阅

Layer.AllFeatures 方法，第 343 页

Layer.DeleteFeature 方法，第 347 页

Layer.Invalidate 方法，第 353 页

Layer.NoFeatures 方法，第 355 页

Layer.SearchAtPoint 方法，第 358 页

Layer.SearchWithinDistance 方法，第 359 页

Layer.SearchWithinFeature 方法，第 359 页

Layer.SearchWithinRectangle 方法，第 360 页

Layer.UpdateFeature 方法，第 363 页

Feature.FeatureKey 属性，第 274 页

Layer.AllFeatures 方法

目的

返回图层中所有图元的 Features 对象。

语法

[Features=]OBJECT.**AllFeatures**

另请参阅

Layer.AddFeature 方法，第 342 页

Layer.DeleteFeature 方法，第 347 页

Layer.Invalidate 方法，第 353 页

Layer.NoFeatures 方法，第 355 页

Layer.SearchAtPoint 方法，第 358 页

Layer.SearchWithinDistance 方法，第 359 页

Layer.SearchWithinFeature 方法，第 359 页

Layer.SearchWithinRectangle 方法，第 360 页

Layer.UpdateFeature 方法，第 363 页

Layer.AutoLabel 属性

目的

此属性控制图层是否自动标记。为了标签可以自动显示，其执行必须位于可查看的地图区域之内。该属性是一个布尔值，默认值为 False。

Layer.BeginAccess 方法

目的

此方法将打开并锁定用于读写访问的表。这将改善重复图层和 Dataset 操作的性能。对于 Layer.BeginAccess 的每次调用，必须调用 Layer.EndAccess 一次。

语法

OBJECT.BeginAccess (BeginAccessType)

部分	说明
OBJECT	表示 Layer 对象。
BeginAccessType	**LayerBeginAccessConstants** 之一，指定访问类型 （例如只读或读写）。

注解

MapX 编写程序不使用 BeginAccess， EndAccess 的工作如前，但是不具有多个操作的性能改善。

您可以嵌套调用 BeginAccess，但是只有第一个生效。

另请参阅

Layer.EndAccess 方法目的，第 350 页

Layer.Bounds 属性

目的

返回 Rectangle 对象，表示图层中所有对象的地理范围 （如其最小边界矩形），不适用于用户绘制图层。

注解

如果要缩小地图以显示一个图层上的全部对象，此属性非常实用：

另请参阅

Layers.Bounds 方法，第 368 页

Layer.CreateAllFeaturesConstraint 方法

目的

此方法创建新的 AllFeaturesConstraint 对象，用于控制缓存中的数据。

语法

[*objAllFeaturesConstraint* =]OBJECT.**CreateAllFeaturesConstraint**

注解

只要该约束 （或者其他约束）存在，由此对象在缓存中保存的数据就将保存。为此指定以下代码：

```
lyr.CreateAllFeaturesObject.Enable = True
```

此代码将创建约束，确保缓存所有记录，如上所示）。但是，新建约束对象在从缓存释放记录时将销毁。在不再需要相应记录时，为了效率更高，约束对象必须存储在由客户端发布的变量之中 （或设置 Nothing 或 Enabled = False）。

与 BoundsConstraint 和 FeaturesConstraint 不同， AllFeatures 约束最初禁用，在启用之前无法确保缓存所有记录。

另请参阅

AllFeaturesConstraint 对象，第 236 页

Feature 对象和 Features 集合，第 271 页

Layer.CreateBoundsConstraint 方法

目的

此方法用于创建新 BoundsConstraint 对象，在控制图层中的缓存数据时使用。

语法

```
[objBoundsConstraint = ]OBJECT.CreateBoundsConstraint
```

注解

只要该约束 （或者其他约束）存在，由此对象在缓存中保存的数据就将保存。为此指定以下代码：

```
Set lyr.CreateBoundsObject.Bounds = rect
```

此代码创建约束并设置约束的界限。这将确保缓存记录。但是，新建 BoundsConstraint 对象在从缓存释放记录时将销毁。在不再需要相应记录时，为了效率更高， BoundsConstraint 对象必须存储在由客户端发布的变量之中 （或设置 Nothing）。

另请参阅

BoundsConstraint 对象，第 249 页

Rectangle 对象，第 437 页

Layer.CreateFeaturesConstraint 方法

目的

此方法创建新 FeaturesConstraint，用于控制缓存中的数据。

语法

```
[objFeaturesConstraint = ]OBJECT.CreateFeaturesConstraint
```

注解

只要该约束 （或者其他约束）存在，由此对象在缓存中保存的数据就将保存。为此指定以下代码：

```
Set lyr.CreateFeaturesObject.Features = ftrs
```

此代码创建约束并设置约束的图元集合，确保缓存这些记录。但是，新建 FeaturesConstraint 对象在从缓存释放记录时将销毁。在不再需要相应记录时，为了效率更高， FeaturesConstraaint 对象必须存储在由客户端发布的变量之中 （或设置 Nothing）。

另请参阅

FeaturesConstraint 对象，第 285 页

Feature 对象和 Features 集合，第 271 页

Layer.ClippedBounds 属性

目的

这一只读属性返回 Rectangle 对象中的所有图层剪贴边界的并集。当地图投影具有无穷极点，使用此属性来设置 Map.Bounds 属性，例如 Mercator 投影，但图层投影则没有。否则，纬度需要手动调整，以便没有产生接近无穷的数字 （否则将导致地图显示为较细的竖线）。

注解

由于这是一个只读属性，因此不可以将其设置为另一个矩形对象。

示例

```
Set Map.Bounds = Map.Layer.ClippedBounds
```

另请参阅

Map.Bounds 属性，第 391 页

Layer.Bounds 属性，第 344 页

Layers.ClippedBounds 属性，第 368 页

Layer.CoordSys 属性

目的

此属性返回只读 CoordSys 对象，表示保存图层的坐标系。

另请参阅

CoordSys 对象，第 252 页

Map.DisplayCoordSys 属性，第 399 页

Map.NumericCoordSys 属性，第 411 页

Layer.ClearCustomLabels 方法

目的

移除放置在当前图层上的任意标记。

语法

OBJECT.ClearCustomLabels

Layer.Datasets 属性 目的

这一只读属性是 Datasets 集合，绑定到其关联的图层。此集合是 Datasets 全集 (Map.Datasets) 的一个子集。 Layer.Datasets 中的对象和 Map.Datasets 中的对象相同。

注解

没有 Datasets 的图层 （例如 Raster 和 UserDraw）将返回空集合。

Layer.Datasets 不支持 Add 或 Restore 方法。如果两种方法均未使用则报错。

通过 Datasets.Add 向 Map.Datasets 添加 Dataset 的同时，还将向其绑定到的图层的 Dataset 集合添加 Dataset。从 Layer.Dataset 集合移除 Dataset 的同时，还将其从 Map.Datasets 中删除，反之亦然。

另请参阅

Datasets.Add 方法，第 263 页

Map.Datasets 属性，第 397 页

Layer.DeleteFeature 方法

目的

从图层删除图元及其数据库行。

语法

OBJECT.**DeleteFeature** (FeatureKey)

部分	说明
OBJECT	表示 Layer 对象。
FeatureKey	一个要删除的 Feature 对象。是 FeatureID 参数的替代参数。 FeatureID 的工作和以前相同，但建议使用新的 FeatureKey 参数。 FeatureKey 用于确定表中的唯一记录。在此前的版本中， FeatureID 用于此目的，但是对于无缝和远程图层无法正常工作。 FeatureKey 属性适用于所有图层类型。

另请参阅

Layer.AddFeature 方法，第 342 页

Layer.DeleteFeature 方法，第 347 页

Layer.Invalidate 方法，第 353 页

Layer.NoFeatures 方法，第 355 页

Layer.SearchAtPoint 方法，第 358 页

Layer.SearchWithinDistance 方法，第 359 页

Layer.SearchWithinFeature 方法，第 359 页

Layer.SearchWithinRectangle 方法，第 360 页

Layer.UpdateFeature 方法，第 363 页

Layer.DrawLabelsAfter 属性

目的

布尔值。设置为 True 时，源自底层图层的所有图层的标签 （或使用此属性集的最后一个图层）取决于，并且包括图层本身在该图层绘制之后绘制。这样可以在绘制顺序中更好地控制标签的位置。

默认情况下，图层将此属性设置为 False，并且标签在最顶层图层之后，在任何动画图层之前绘制。

Layer.DrilldownAddFeatures 方法

目的

向逐层细化图层添加图元。

语法

OBJECT.**DrilldownAddFeatures** (Level, FeatureKeys)

部分	说明
OBJECT	Layer 对象相应于 Drilldown 图层。
Level	字符串，表示 Drilldown 图层中的哪一层次包含要添加的图元。
FeatureKeys	变量：字符串或字符串数组，表示应该添加的图元的逐层细化 ID。注意这些是逐层细化 ID （例如值源自逐层细化组件表中的列），不是 MapX 图元的内部 FeatureID 值。

注解

此方法并不实际更改任意表的内容。您正在添加的图元已在表中，通过将其 “添加”到图层，可令其可见。

Layer.DrilldownRemoveFeatures 方法

目的

从逐层细化图层移除图元。

语法

OBJECT.**DrilldownRemoveFeatures** (Level, FeatureKeys)

部分	说明
OBJECT	Layer 对象相应于 Drilldown 图层。
Level	字符串表示包含要移除的图元的层次。
FeatureKeys	变量：字符串或字符串数组，表示应该移除的图元的逐层细化 ID。注意这些是在逐层细化表的第一列指定的 ID，不是 MapX 图元的内部 FeatureID 值。

注解

此方法并不实际更改任意表的内容。您正在移除的图元已在表中，通过将其从图层 “移除”，可令其消失。

Layer.DrilldownReset 方法

目的

清除 Drilldown 图层的所有图元，使用源自单级的图元重新初始化图层。

语法

OBJECT.**DrilldownReset** (Level)

部分	说明
OBJECT	Layer 对象相应于 Drilldown 图层。
Level	字符串表示在 Drilldown 图层应该显示的细节层次。如果字符串为空，则图层将会清空 （所有对象都将从 Drilldown 图层移除）。

注解

此方法重设 Drilldown 图层，以便它只显示一个特定级别的图元。如果您要提供 “重置” 按钮，您可以使用此方法，将 Drilldown 图层恢复到其原始状态。

Layer.Editable 属性

这一属性允许编辑图层中的选择 （移动、调整大小或删除）。 ArrowTool 可用于移动所选对象或调整大小。 Delete 键将删除所选对象。

Layer.EndAccess 方法目的

EndAccess 方法解除地图表的锁定。对于 Layer.BeginAccess 的每次调用，必须调用 Layer.EndAccess 一次。

语法

OBJECT.**EndAccess** (endAccessType)

部分	说明
OBJECT	表示 Layer 对象。
endAccessType	**LayerEndAccessConstants** 之一，指定结束访问类型。

注解

如果调用 Layer.BeginAccess，带有 miAccessRead 或 miAccessReadWrite 的 beginAccessType，则 EndAccessType 参数必须为 miAccessEnd。

MapX 编写程序不使用 BeginAccess， EndAccess 的工作如前，但是不具有多个操作的性能改善。

另请参阅

Layer.BeginAccess 方法，第 344 页

Layer.FeatureIDFromFeatureName 方法

给定名称，这一方法将返回图元的 ID，使用该名称 （如果名称在该图层中不唯一，则返回第一个图元的 ID，带有指定名称）。当前定义的名称是 MapInfo 表的第一列中的值。如果没有指定名称的图元，则将会报错。

语法

[int=]OBJECT.**FeatureIDFromFeatureName** (Name)

部分	说明
OBJECT	表示 Layer 对象。
Name	字符串，图元的名称，其 ID 为要取的 ID。

Layer.FeatureKeyFromFeatureName 方法

目的

此方法返回源自 Layer 对象中的 FeatureName 的 FeatureKey。MapX 搜索 Layer.KeyField 查找第一个包含 KeyValue 参数中的名称的行。如果没有找到 KeyValue，则 MapX 将报错 #1205。否则，将返回它所发现的新行的 FeatureKey。

语法

[BSTR=]OBJECT.**FeatureKeyFromFeatureName** (KeyValue)

部分	说明
OBJECT	表示 Layer 对象。
KeyValue	图元的名称和要搜索的值。

注解

此方法不适用于远程数据库图层 （ODBC 或 OCI）。**Layer.Search 方法**可用作在远程数据库图层上确定图元或图元集合位置的另一种方法。

另请参阅

Layer.KeyField 属性，第 353 页

Layer.FileSpec 属性

目的

此属性包含文件的完全规格，包括图层的物理存储位置。这是一个字符串值，是只读的。

Layer.Find 属性

目的

此属性定义图层的 Find 对象。

Layer.GetDrilldownFeaturesByID 方法

目的

此方法返回 Feature 对象，给出图元的 ID。

语法

[Feature=]OBJECT.**GetDrilldownFeaturesByID**(strLevel, FeatureID)

部分	说明
OBJECT	表示 Layer 对象。
strLevel	Drilldown 图层中的层次名称。
FeatureID	字符串或字符串数组，确定要查询的图元。

Layer.GetFeatureByID 方法

目的

此方法返回 Feature 对象，给出图元的 ID。

语法

[Feature=]OBJECT.**GetFeatureByID**(FeatureKey)

部分	说明
OBJECT	Layer 对象。
FeatureKey	定义图元的字符串。由 **Feature.FeatureKey** 属性返回的值。这是 **Feature.FeatureID** 的替代，此属性的工作和以前相同，但是建议您使用 FeatureKey 属性。FeatureKey 属性用于确定表中的唯一记录。在此前的版本中， FeatureID 属性用于此目的，但是对于无缝和远程图层无法正常工作。 FeatureKey 属性适用于所有图层类型。

另请参阅

Feature.FeatureKey 属性，第 274 页

Feature.FeatureID 属性，第 274 页

Layer.GetFeatureByKey 方法

目的

此方法返回 Feature 对象，带有 Layer 对象中的给定关键字。

语法

[Feature=]OBJECT.**GetFeatureByKey** (FeatureKey)

部分	说明
OBJECT	表示 Layer 对象。

部分	说明
FeatureKey	定义图元的字符串。 This is the value returned by Feature.FeatureKey property.FeatureKey 用于确定表中的唯一记录。在此前的版本中， FeatureID 属性用于此目的，但是对于无缝和远程图层无法正常工作。 FeatureKey 属性适用于所有图层类型。

另请参阅

Feature.FeatureKey 属性，第 274 页

Layer.Invalidate 方法

目的

此方法导致重新绘制矩形 （在容器屏幕坐标中）。

语法

OBJECT.**Invalidate** ([InvalidRect])

部分	说明
OBJECT	表示 Layers 对象。
InvalidRect	表示相应矩形。

另请参阅

Layer.AddFeature 方法，第 342 页

Layer.DeleteFeature 方法，第 347 页

Layer.NoFeatures 方法，第 355 页

Layer.SearchAtPoint 方法，第 358 页

Layer.SearchWithinDistance 方法，第 359 页

Layer.SearchWithinFeature 方法，第 359 页

Layer.SearchWithinRectangle 方法，第 360 页

Layer.UpdateFeature 方法，第 363 页

Layer.KeyField 属性

目的

此字符串属性确定图层的 MapInfo 表中的列 （字段）名，将由图元的 KeyValue 属性设置或检索。它当前默认为图层表中的第一列。此属性不应用于光栅或用户绘制图层。如果尝试将 KeyField 设置为无效字段名，则将会报错。

Layer.Labels 属性

目的

这一只读属性返回用于地图中的图层的 Labels 集合。图层的 Labels 集合包含该图层中在地图窗口或缓存中可见的所有标签，以及来自已经通过手动 （使用普通选择工具）或程序 （使用 Layer 对象和 Layers 集合的方法和属性）编辑过的图层的任意标签。

语法

[Labels=]OBJECT.**Labels**

另请参阅

Label 对象和 Labels 集合，第 328 页

Map.EditableLabels 属性，第 400 页

Layer.LabelAtPoint 方法

目的

标记某点处的特定对象。

语法

OBJECT**.LabelAtPoint** (x, y)

部分	说明
OBJECT	表示 Layer 对象。
x	点的 X 坐标 （双精度）。
Y	点的 Y 地图坐标 （双精度）。

Layer.LabelProperties 属性

目的

只读属性，返回图层的 LabelProperties 对象。 LabelProperties 对象用于控制图层标签的外观。

另请参阅

LabelProperties 对象，第 335 页

Layer.AutoLabel 属性，第 343 页

Layer.Name 属性

目的

此属性包含图层的名称。图层在添加时给定名称。

Layer.NoFeatures 方法

目的

返回图层的空 Features 对象。

语法

OBJECT.**NoFeatures**()

另请参阅

Layer.AddFeature 方法，第 342 页

Layer.DeleteFeature 方法，第 347 页

Layer.Invalidate 方法，第 353 页

Layer.SearchAtPoint 方法，第 358 页

Layer.SearchWithinDistance 方法，第 359 页

Layer.SearchWithinFeature 方法，第 359 页

Layer.SearchWithinRectangle 方法，第 360 页

Layer.UpdateFeature 方法，第 363 页

Layer.OverrideStyle 属性

目的

控制地图图元的样式是否应该使用 Style 属性覆盖。该属性是一个布尔值，默认值为 False。

另请参阅

Layer.Style 属性

Layer.Pack 方法

目的

此方法令您可以压缩表 （以便使用较小磁盘空间），消除已经标记为删除的记录。打包所有数据将重建索引和图形文件。

注：执行 Pack Table 操作将可能导致 MapInfo Professional Workspaces 中的定制标签崩溃。有关详细

信息，请参阅 MapInfo 专业版联机帮助或用户指南。

语法

OBJECT.Pack (PackType)

部分	说明
OBJECT	表示 Layer 对象。
PackType	一个或多个 LayerPackConstants，指定要执行的打包类型。 • 去除已删除的记录 • 重建图形 • 打包图形 • 重建索引 • 打包索引 • 压缩数据库 （只对 Access 表有效）

注解

尽管打包文件的图形信息可以保存磁盘空间，但是将会导致该文件中的图形更新操作较慢，获取磁盘控件将会牺牲处理速度。此外，如果您通过执行导入操作创建了一个 TAB 文件，可能需要打包图形数据。

在打包表之前，务必确保有足够的磁盘空间来存储作为初步文件的数据库复本。您不能打包只读文件。如果表的文件无法锁定用于写入访问，则将无法对表执行打包。不调用 Layer.EndAccess 方法，将无法打包为之调用 Layer.BeginAccess 方法的表。

要重建索引和 / 或图形文件，不去除已删除的记录，可分别指定重建选项 （例如重建图形和 / 或重建索引）。要使用更高的压缩比来打包索引和 / 或图形文件，可指定用于索引和 / 或图形的打包选项。如果表为 Access (DAO) 表，则您还可以在其他选项之外或通过其本身指定 “压缩”。

在调用 Pack 之前调用 Layer.SupportsPack 方法，查看是否 Layer 的表支持预期的打包操作。

另请参阅

LayerPackConstants，第 569 页

Layer.SupportsPack 方法，第 362 页

Layer.PredominantFeatureType 属性

目的

只读属性，返回表示 **FeatureTypeConstants** 之一的整数。用于确定图层中哪类图元最为普遍。

注解

如果图层中多种类型的图元的数量相同 （例如 2 个符号和 2 个区域），则 PredominantFeatureType 将返回 miFeatureTypeMixed，除非类型之一是区域，此时 PredominantFeatureType 将返回 miFeatureTypeRegion。

Layer.Refresh 方法

目的

此方法将从图层刷新缓存。这对于若干启用缓存的图层非常实用。

语法

OBJECT.**Refresh**

另请参阅

LayerInfo 对象，第 373 页

使用缓存，第 166 页

Layer.Search 方法

目的

Layer.Search 方法展示了 SQL 查询的函数。表达式 （语句的 where 子句）将为 Layers 表中的每行求解，并返回 **Features 集合**，构成每个图元，表达式为真。

语法

[Features collection=]OBJECT.**Search** (strWhere, [Variables])

部分	说明
OBJECT	表示 Layer 对象。
strWhere	字符串：对每个图元求解的表达式。
Variables	**Variant**：这是一个可选参数。如果指定，则必须为 Variables 对象。当表达式和无法解析的标识符交互时，它将尝试在 Variables 集合中进行解析。

注解

Layer.Search 不适用于光栅、无缝或用户绘制表。如果表达式使用的不是 Dataset 字段名的标识符，或者在 Variables 集合中发现 Variable，则 Layer.Search 将会报错。

如果使用 AND、 OR 或 NOT 表达式中的字符串值，如果其中至少有一个字符，则字符串将视为 True ；如果其长度为 0，则为 False。

在表达式中， MapX 将执行从布尔值到用于预期为字符串参数的函数参数字符串的类型转换。

将日期指定为字符串常数时，以下项目适用：

- 由月份、日期和可选年份组成的日期。
- 年份必须由 4 位数字指定。
- 整个日期字符串应该使用双引号引起来。
- 日期的组成部分应该通过连字符或短横线间隔开。

另请参阅

创建表达式，第 575 页

Idispatch 表，第 633 页

地理运算符，第 576 页

Variable 对象和 Variables 集合，第 502 页

MapX 字段命名规范，第 507 页

Layer.SearchAtPoint 方法

目的

返回 Features 集合对象。

- 区域 – 如果点在区域之内，则区域在结果集合之内。
- 线和符号 – 在给定的点周围的 3 个像素区域之内的最接近的点或先 （如果许多点的坐标相同则可有多个）将添加到结果集合。
- 组合 – 通过组合 （使用 OR）一个或多个 **SearchResultTypeConstants**，您可以定义您是否需要所有图元、最顶层的图元、所有区域、所有点、最顶端的点等。默认为返回所有图元。

语法

[Features collection=]OBJECT.**SearchAtPoint** (Point, [SearchResultFlag])

部分	说明
OBJECT	表示 Layer 对象。
Point	给定点位置的 X 和 Y 地图坐标。
SearchResultFlag	此可选参数控制返回的内容。它取 **SearchResultTypeConstants** 的一个值或组合值。

另请参阅

Layer.AddFeature 方法，第 342 页

Layer.DeleteFeature 方法，第 347 页

Layer.NoFeatures 方法，第 355 页

Layer.Invalidate 方法，第 353 页

Layer.SearchWithinDistance 方法，第 359 页

Layer.SearchWithinFeature 方法，第 359 页

Layer.SearchWithinRectangle 方法，第 360 页

Layer.UpdateFeature 方法，第 363 页

Layer.SearchWithinDistance 方法

目的

搜索指定距离内的地图图元，将搜索结果作为 Features 集合返回。

语法

OBJECT.**SearchWithinDistance**(Source, double Distance, short Units, short SearchType)

部分	说明
OBJECT	表示 Layer 对象。
Source	Point 对象或 Feature 对象，表示搜索源。
Distance	要搜索的距离，采用由 Units 参数指定的地图单位。
Units	**MapUnitConstants** 值，例如 miUnitMile (0)，确定 Distance 参数的测量单位。
SearchType	**SearchTypeConstants** 值，控制用于确定图元是否处于搜索区域之内的搜索标准。

注解

此方法在源对象周围创建一个缓冲区，然后在缓冲区之内搜索。在执行搜索之后，缓冲将被忽略，如果需要保存缓冲，则使用 BufferFeatures 方法来取代。

缓冲区的分辨率 （节点数）由 Map.DefaultConversionResolution 属性控制。

另请参阅

Layer.AddFeature 方法，第 342 页

Layer.DeleteFeature 方法，第 347 页

Layer.Invalidate 方法，第 353 页

Layer.NoFeatures 方法，第 355 页

Layer.SearchAtPoint 方法，第 358 页

Layer.SearchWithinFeature 方法，第 359 页

Layer.SearchWithinRectangle 方法，第 360 页

Layer.UpdateFeature 方法，第 363 页

Layer.SearchWithinFeature 方法

目的

此方法返回其搜索的图元所在的 Features 集合。

语法

OBJECT.**SearchWithinFeature** (Feature, SearchType)

部分	说明
OBJECT	表示 Layer 对象。
Feature	用作搜索基础的 Feature 对象。
SearchType	SearchType 是 miSearchTypeCentroidWithin、 miSearchTypePartiallyWithin 或 miSearchTypeEntirelyWithin。

另请参阅

Selection.SelectByRegion 方法，第 452 页

Layer.AddFeature 方法，第 342 页

Layer.DeleteFeature 方法，第 347 页

Layer.Invalidate 方法，第 353 页

Layer.NoFeatures 方法，第 355 页

Layer.SearchAtPoint 方法，第 358 页

Layer.SearchWithinDistance 方法，第 359 页

Layer.SearchWithinRectangle 方法，第 360 页

Layer.UpdateFeature 方法，第 363 页

Layer.SearchWithinRectangle 方法

目的

此方法返回指定 Rectangle 边界范围内的 Features 集合。

语法

[features collection=]OBJECT.**SearchWithinRectangle** (Rectangle, SearchType)

部分	说明
OBJECT	表示 Layer 对象。
Rectangle	用作搜索基础的 **Rectangle** 对象。
SearchType	SearchType 是 miSearchTypeCentroidWithin、 miSearchTypePartiallyWithin、 miSearchTypeEntirelyWithin。

另请参阅

Layer.AddFeature 方法，第 342 页

Layer.DeleteFeature 方法，第 347 页

Layer.Invalidate 方法，第 353 页

Layer.NoFeatures 方法，第 355 页

Layer.SearchAtPoint 方法，第 358 页

Layer.SearchWithinFeature 方法，第 359 页

Layer.SearchWithinDistance 方法，第 359 页

Layer.UpdateFeature 方法，第 363 页

Layer.Selectable 属性

目的

控制图层是否可使用选择进行选择。此属性不影响使用 **Selection 选择** 中的自动方法的选择进程。该属性取布尔值，默认值为 On/True。

Layer.Selection 属性

目的

该图层的所选图元集合。

另请参阅

Selection 集合，第 446 页

Layer.ShowNodes 属性

目的

这一读写属性为 true 时显示图层中所有图元的所有节点。默认值为 false。

Layer.ShowCentroids 属性

目的

这一读写属性设置为 true 时绘制对象质心 （仅限区域）。默认值为 false。

Layer.ShowLineDirection 属性

目的

这一读写属性设置为 true 时绘制表现线的方向的箭头 （仅限线和多义线）。默认值为 false。

Layer.Style 属性

目的

Style 对象用于覆盖地图上的图层中的所有图元的外观。 Style 对象将检索并填充新请求的覆盖值。如果 Override 属性设置 True，则使用此样式。

另请参阅

Layer.OverrideStyle 属性，第 355 页

Layer.SupportsPack 方法

目的

此方法用于确定图层是否可以打包。

语法

[bool=]OBJECT.SupportsPack (PackType)

部分	说明
OBJECT	表示 Layer 对象。
PackType	一个或多个 LayerPackConstants，指定要执行的打包类型。
bool=	表示图层是否可以打包的布尔值。

另请参阅

Layer.Pack 方法，第 355 页

LayerPackConstants，第 569 页

Layer.Type 属性

目的

这是只读属性，返回相应于 **LayerTypeConstants** 之一的整数值。

- miLayerTypeNormal = 0
- miLayerTypeRaster = 2
- miLayerTypeSeamless = 4
- miLayerTypeUnknown = 5
- miLayerTypeUserDraw = 6

Layer.UpdateFeature 方法

目的

此方法更新图层中的目标图元，具有的属性为 Source 图元。此方法对于对象编辑非常实用。

语法

OBJECT.**UpdateFeature** (Target, [Source], [RowValues])

部分	说明
OBJECT	表示 Layer 对象。
Target	要更新的 Feature 对象，可以为 FeatureKey、 FeatureID 或 Feature 对象。但是，不建议您使用 FeatureKey，由于它是 FeatureID 参数的替代。（**注释**：FeatureID 的工作和以前相同。） FeatureKey 用于确定表中的唯一记录。在此前的版本中，FeatureID 属性用于此目的，但是对于无缝和远程图层无法正常工作。 FeatureKey 属性适用于所有图层类型。
Source	这可选参数是取属性的 Feature 对象。默认值为 Target。
RowValues	此可选参数是 RowValues，表示用于一行数据的属性数据新值。 **RowValues 集合**中的每个值均相应于属性数据的一列。此参数只适用于 miDatasetLayer 类型的 Dataset。

注解

此方法不应该用于更新计算列，但是如果这样做也不会报错。其原因在于这是一个 RowValues 集合，可能包括计算列和非计算机列。

在 Computetheme 属性设置为 false 时， legend.showemptyranges 属性必须设置为 true，以便令主题的图例文本可见。在 computetheme 设置为 false 时，这是由于 MapX 不计算主题类别计数。

名称或别名为 MI_PRINX 或 PRINX 的列不可更新。

注：Layer.UpdateFeature 在用户尝试更新不可更新列时将不会报错。以下数据类型的 Oracle 图层列不可更新，并且在尝试更新时不会报错：

- Long
- Raw
- LOB,BFILE

另请参阅

RowValue 对象和 RowValues 集合，第 442 页

Feature 对象和 Features 集合，第 271 页

Layer.Visible 属性

目的

控制图层是否可见。该属性是一个布尔值，默认值为 True。

Layer.ZoomLayer 属性

目的

控制图层是否是缩放图层。缩放图层控制图层显示的缩放级别范围 （地图上的距离）。如果缩放图层为启用，则值存储在 ZoomMax 之中，并且将使用 ZoomMin 属性。该属性是一个布尔值，默认值为 False。

另请参阅

Layer.ZoomMax 属性，第 364 页

Layer.ZoomMin 属性，第 364 页

Layer.ZoomMax 属性

目的

如果标签 ZoomLayering 为 （Layer.ZoomLayer 属性）启用，则此属性指定在地图上绘制的图层的最大缩放值。取双精度值，以地图单位指定距离 (Map.MapUnit)。

另请参阅

Layer.ZoomMax 属性，第 364 页

Layer.ZoomMin 属性，第 364 页

Layer.ZoomMin 属性

目的

如果标签 ZoomLayering 为 （Layer.ZoomLayer 属性）启用，则此属性指定在地图上绘制的图层的最小缩放值。取双精度值，以地图单位指定距离 (Map.MapUnit)。

另请参阅

Layer.ZoomLayer 属性，第 364 页

Map.MapUnit 属性，第 405 页

Layers.Add 方法

目的

此方法创建指定文件的图层对象，将其添加到集合，并在地图上显示它。

语法

[Layer=]OBJECT.**Add** (LayerInfo, [Position])

部分	说明
OBJECT	表示 Layers 对象。
LayerInfo	**Variant**：可以是 **LayerInfo 对象**或要添加为图层的 MapInfo 表 (.TAB) 文件的路径名。
Position	**Variant**：图层列表中的初始位置。如果忽略或为 0，则将使用自动图层自定算法。

注解

该图层相对于地图中的其他图层自动定位。例如，带有点的图层将置于带有区域的图层之上。

如果字符串传递用于 LayerInfo 参数，则该字符串将被视为文件规格，其说明如下：

- .tab 扩展名视为 .tab 文件。
- 任何其他扩展名将视为自注册的光栅文件。
- 无扩展名视为 geodictionary 用户名。

MapX 不再允许图层名重复。当开发人员显式指定图层名 （正如 **LayerInfo 对象**、**Layers.AddUserDrawLayer** 和 **Layers.AddServerLayer** 的名称参数所示，如果名称不唯一，则将会报错。

当 LayerInfo 的类型属性为 NewTab，而 Fields 参数的字符串字段长度为 255 或更高时，将会报错。字段长度是以 MapInfo 专业版的 254 个字符为基础。

另请参阅

MapX 字段命名规范，第 507 页

Layers.AddGeosetLayers 方法

目的

此方法向当前地图添加指定的 GeoSet，为不在 Layers 集合中的 GeoSet 指定的每个图层创建 Layer 对象。

语法

OBJECT.**AddGeosetLayers** (GeosetName)

部分	说明
OBJECT	表示 Layers 对象。
GeosetName	Geoset 友好名称或文件完全规格。

另请参阅

Map.GeoSet 属性，第 403 页

Layers.AddServerLayer 方法

目的

包含此方法用于向下兼容性。建议您不使用此方法。执行相同函数的首选方法只是 Layers.Add 方法，并带有 LayerInfo 对象，传递为参数。这允许您指定该图层是服务器图层，用于控制缓存和 MBR 搜索，将远程空间数据添加到地图 （即从远程数据库检索的图元图层，例如 Oracle 或 Informix）。

注：此方法在 MapX Mobile 中不可用。

语法

[Layer=]OBJECT.**AddServerLayer** (Name, ConnectString, Query, [Position], [ServerLayerOptions])

部分	说明
OBJECT	Layer 集合对象。
Name	表示要创建的图层名称的字符串 （例如 “高速公路”）
ConnectString	ODBC 连接字符串，指定连接信息，例如主机名、口令等。如果字符串没有提供所有必要信息，则 ODBC 数据源将显示连接对话框。您可以复用连接，避免重复连接对话。如果传递完整的连接字符串匹配当前连接池中的现有字符串，则连接将由两个表共享。要匹配两个字符串，连接字符串中不需要口令。
Query	指定 GISSQL/ 针对服务器的 SQL 查询的字符串。
Position	可选：图层列表中的位置。要令此图层成为地图中的顶层，可指定为 1。
ServerLayerOptions	可选：接受 LayerSrvLayerOptions 值的位逻辑运算符 OR，控制服务器图层的行为。默认值为 (miLayerCacheOn OR miMBRSearchOn)。分发其的原因在于在 VB 中， OR 运算符用于逻辑和位逻辑运算 OR，但 C++ 采用了两个不同的运算符。在 C++ 应用中嵌入 MapX 应该使用正确的 OR。以便控制服务器图层的特定行为 （当前为缓存和 MBRsearch 选项）。它不取布尔值。取值为一组标记 （位掩码），如下所示。使用提供的枚举值，然后将其求 OR，在调用 AddServerLayer 之前定义 SrvLayerOptions 值。

下表概要列出了 Options 参数的效用。常数可以是逐位，一起用于合并选项。

miLayerCacheOn	读取的属性和对象将保存在内存中。这将实现更快的地图交互 （例如缩放、标记、主题和移动）。但是，对于数据的最新更新将在 Layer.Refresh 调用之后显示。
miLayerCacheOff	所有数据将在每次绘制图层时取自数据库。这样做效率较低，但是可以提供最及时的数据。
miLayerMBRSearchOn	这将导致在检索数据时，将 MBRSearch 谓词添加到空间查询。
miLayerMBRSearchOff	禁用 MBRSearch 谓词将在绘制地图时将其附加到查询。这对于已经包含空间谓词的查询和附加空间谓词应该避免时非常实用。

注解

MapX 不再允许图层名重复。当开发人员显式指定图层名 （正如 **LayerInfo 对象**、**Layers.AddUserDrawLayer** 和 **Layers.Add** 的名称参数所示），如果名称不唯一，则将会报错。

这将只添加 ODBC 图层。 Oracle 空间图层必须通过 Layers.Add 方法使用 LayerInfo 对象添加。

另请参阅

Layers.Add 方法，第 365 页

LayerInfo 对象，第 373 页

Layers.AddUserDrawLayer 方法

目的

包含此方法用于向下兼容性。建议您不使用此方法。执行相同函数的首选方法只是 Layers.Add 方法，并带有 LayerInfo 对象，传递为参数。但是，此方法添加容器应用程序负责绘制的图层。它用于和 **DrawUserLayer 事件** 一起使用，该事件在图层需要绘制时触发。返回新建的 Layer 对象。

语法

OBJECT.**AddUserDrawLayer** (Name, Position)

部分	说明
OBJECT	表示 Layers 对象。
Name	要创建的图层的名称。
Position	要在图层列表中添加图层的位置。

注解

MapX 不再允许图层名重复。当开发人员显式指定图层名 （正如 **LayerInfo 对象**、**Layers.AddServerLayer** 和 **Layers.Add** 的名称参数所示），如果名称不唯一，则将会报错。

Layers.AnimationLayer 属性

动画图层在地图图元需要频繁更新的地方尤为实用，例如实时的应用程序。例如，您可以开发车队管理应用程序，在其中将每个车辆表示为一个点对象。您可以使用全球定位卫星 (Global Positioning Satellite, GPS) 技术接收当前的车辆坐标，然后更新点对象，在地图上显示当前的车辆位置。在此类型的应用程序中，地图对象不断变化，如果正在更新的对象存储在动画图层中而不是常规图层中，地图的重新绘制将会更加迅速。

AnimationLayer 最初设置为空。您可以将 Layer 对象分配给属性，令该图层称为动画图层 （可以是常规图层或用户绘图图层）。将图层分配给 AnimationLayer 属性时，它将在所有图层顶部绘制，包括注释图层和选择。该图层还处于 Layers 集合的相同位置上。浮动对象，类似图例，将仍然显示在动画图层的顶部，尽管它们由于已经剪切而无需每次重新绘制。如果正常的图层用作动画图层，选择和标记仍然有效。

要关闭动画图层，可以将该图层指定为空：

```
Set Map.Layers.AnimationLayer = nothing
```

这将图层转换回正常图层，仍然置于图层列表中的相同位置。

Layers.Bounds 方法

目的

返回 Rectangle 对象，表示集合中所有图元的地理范围 （如其最小边界矩形），不适用于用户绘制图层。

注解

如果要缩小地图以显示所有图层上的全部对象，此属性非常实用。

另请参阅

Layer.Bounds 属性，第 344 页

Layers.ClippedBounds 属性

目的

这一只读属性返回 Rectangle 对象中的所有图层剪贴边界的并集。当地图投影具有无穷极点，使用此属性来设置 Map.Bounds 属性，例如 Mercator 投影，但图层投影则没有。否则，纬度需要手动调整，以便没有产生接近无穷的数字 （否则将导致地图显示为较细的竖线）。

注解

由于这是一个只读属性，因此不可以将其设置为另一个矩形对象。

示例

```
Set Map.Bounds = Map.Layers.ClippedBounds
```

另请参阅

Map.Bounds 属性，第 391 页

Layer.Bounds 属性，第 344 页

Layer.ClippedBounds 属性，第 346 页

Layers.ClearSelection 方法

目的

取消选择 **Layers** 集合的所有图层中的图元。

另请参阅

Selection.ClearSelection 方法，第 447 页

Layers.Count 属性

目的

包含集合中的 **Layer** 对象数。该属性是一个整数值，是只读的。

Layers.CreateLayer 方法

目的

CreateLayer 方法允许您创建新的临时或永久 MapInfo 表图层。创建的表具有一个用于图元名的列，由标记和数据绑定使用。在添加或更新图元时， Name 属性将置于图元名称列中。

注：这是一个 “不赞成使用”的方法，目前仍然可以工作，确保向后的兼容性。使用 **Layers.Add**，通过 **Layerinfo 对象**是创建图层的新途径。

语法

[Layer=]OBJECT.**CreateLayer** (Name , [FileSpec] , [Position] , [KeyLength] , [CoordSys])

部分	说明
OBJECT	Layer 集合对象。
Name	引用图层所用的名称 （图层的用户名）。
FileSpec	Variant：创建图层位置的路径名。文件名应该包括 .tab 扩展名。构成 MapInfo 表（.map、 .dat 等）的其他文件在 .tab 文件所在的同一目录中创建。如果没有给定文件名，则将创建临时图层，该图层在地图销毁时删除。

部分	说明
Position	Variant：图层列表中的初始位置。如果忽略，则字段图层定位算法将基于图层的类型指定图层顺序。
KeyLength	Variant：添加到保存图元名称的表的列的长度。如果忽略，则默认为 32。
CoordSys	Variant：CoordSys 对象，指定存储新图层的坐标系。可选；如果省略，则使用 Map.NumericCoordSys 属性。

注解

该方法返回 Layer 对象，添加到集合的 Layer 对象。新图层使用名为 GEONAME 的关键字列创建。

Layer.InsertionLayer

目的

此属性指定要通过内置对象编辑工具插入的新图层对象。InsertionLayer 最初设置为空。您可以将 Layer 对象指定为属性，以令该图层成为 InsertionLayer。

注解

您不能将逐层细分、光栅、无缝或用户绘制图层设置为插入图层。

在可以将该图层设置为 InsertionLayer 之前，图层的可编辑属性必须设置为 ture。

无论何时，只能将图层指定为插入图层。

AddPoint、 AddLine、 AddRegion 和 AddPolyLine 普通工具 （对象创建工具）将向指定为插入图层的图层添加图元。

在指定插入图层之前，您不能将 **Map.CurrentTool** 设置为任意对象创建工具。

Layers.Item 属性

目的

从集合取特定的 Layer 对象。项目或者取索引 （自 1 开始的整数值）或图层名，指定要取的 Layer 对象。

Layers.LayersDlg 方法

目的

提供了一个对话框，用户在其中可以添加图层、移除图层、更改图层顺序和更改图层属性。如果用户单击 “确定”，则在对话框中所做的更改将立即应用于地图。

语法

[Boolean=]OBJECT.**LayersDlg** ([HelpFile], [HelpID])

部分	说明
OBJECT	Layers 集合。
HelpFile	Variant：Windows 帮助文件的文件名。（可选）
HelpID	Variant：帮助文件中的主题 ID 号。（可选）

注解

如果您指定 HelpFile 和 HelpID，则对话框将包括 “帮助”按钮，如果用户单击 “帮助”按钮，则 MapX 将显示指定的帮助主题。

如果用户单击 “确定”，则 LayersDlg 的返回值为 True ；如果用户单击 “取消”则为 False。

示例

以下语句显示 “图层控制”对话框。

```
Map1.Layers.LayersDlg
```

Layers.Move 方法

目的

移动 Layer 集合中的图层，更改图层绘制的顺序。

语法

OBJECT.**Move** (From, To)

部分	说明
OBJECT	表示 Layer 对象。
From	要移动的图层的索引号。最顶层为 1。
To	用于图层的新位置。例如，如果令其成为第 2 个图层，则使用 2。

注解

Layers 集合中的图层顺序，控制绘制顺序。当图层为使用 Layers.Add 或 Layers.AddGeosetLayers 添加时，图层将只能插入到图层集合中。例如，带有点的图层将置于带有线的图层之上，带有线的图层将置于带有区域的图层之上，带有区域的图层将置于光栅图层之上。

Layers.Remove 方法

目的

此方法从集合去除 Layer 对象。这将影响到从地图去除图层

语法

OBJECT**.Remove** (Index)

部分	说明
OBJECT	表示 Layer 对象。
Index	或者是从 1 开始的整数索引，或者是要去除的图层的名称。

注解

如果通过 Layers.Remove 方法从 Layers 集合去除图层，则和图层关联的数据集也将从 Datasets 集合中被删除。

如果您去除项目，集合索引将再次计数，填充由已删除项目留下的空洞。

Layers.RemoveAll 方法

目的

从集合去除所有图层对象。

语法

OBJECT.**RemoveAll**

LayerInfo 对象

LayerInfo 对象是第一个传递给 Layers.Add 的参数。传递给 Layers.Add 的 LayerInfo 对象说明 / 定义要添加的图层。 LayerInfo 对象的 Type 属性确定了对象必须具备的参数，即任意给定的 LayerInfo 对象必须具有的参数将因要添加的层次类型的不同而异。 LayerInfo.AddParameter 方法用于设置这些参数。 LayerInfo.Type 设置为 **LayerInfoTypeConstants** 之一。

下表显示了需要用于各种图层类型的参数。

LayerInfoType	参数	必需	类型
miLayerInfoTypeTab	FileSpec	是	**String**：要打开的 .tab 文件的文件规格
	Name	否	**String**：添加的图层的名称
miLayerInfoTypeUserDraw	Name	是	**String**：添加的图层的名称
miLayerInfoTypeRaster	FileSpec	是	**String**：要打开的自注册光栅文件的文件规格
	Name	否	**String**：添加的图层的名称
miLayerInfoTypeShape *MapX 支持的形状有： • 空 • 点 • 多义线 • 多边形	FileSpec	是	**String**：.shp 文件的文件规格 （形状文件）。
	Name	否	**String**：添加的图层的名称
	CoordSys	是	**CoordSys Object**：对象数据存储在的 .shp 文件所在的坐标系。
	Style	否	**Style Object:** 呈现对象所使用的样式。
miLayerInfoTypeServer	Name	是	**String**：添加的图层的名称
	ConnectString	是	**String**：ODBC 连接字符串 （请参阅 AddServerLayer）
	Query	是	**String**：SQL 查询 （请参阅 AddServerLayer）
	ToolKit	是	**String**：ToolKit 选项， ODBC 或者 ORAINET

LayerInfoType	参数	必需	类型
	Cache	否	**String** – 可能的值有： • OFF – 缓存不用于此图层。 • ON （默认） – 启用缓存，基于地图视图自动执行缓存 （对中 / 缩放）此外，用户可以通过缓存约束对象控制缓存。 • USER – 启用用于此图层的缓存，但是不在 MapX 内部自动执行缓存。只有缓存用户请求的数据 （通过缓存约束对象）。 • ALL – 整个图层在图层的生存期内均缓存。 默认值为 ON。 **注：**指定 Cache=ON 或 USER，将 MBRSearch 设置为 OFF，在内部处理时和 Cache=ALL 相同，缓存所有记录。
	MBRSearch	否	**String：**ON 或 OFF。 默认值为 ON。
	CacheStorageType	否	**String:**MemTable 或 Native。默认值为 MemTable。
	CacheMaxDisk	否	**Numeric：**表示由缓存使用的磁盘空间上限 (KB)。值为 -1 表示不加限制，是默认设置。请注意即使 Memtable 的 CacheStorageType 也使用某些磁盘空间存储地理对象。此选项只有在缓存参数设置为 ON 时应用。
	CacheMaxMemory	否	**Numeric：**表示由缓存使用的内存空间上限 (KB)。值为 -1 表示不加限制，是默认设置。此选项只有在 CacheStorageType 参数设置为 Memtable 时可以应用。此选项只有在缓存参数设置为 ON 时应用。
	CacheMaxAge	否	**Numeric：**表示未使用约束仍然保存在缓存的历史记录中的时间上限 （秒）。值为 -1 表示不加限制，是默认设置。此选项只有在缓存参数设置为 ON 时应用。
	CacheMaxRecords	否	**Numeric：**表示由保存在缓存中的记录数量上限。值为 -1 表示不加限制，是默认设置。此选项只有在缓存参数设置为 ON 时应用。

LayerInfoType	参数	必需	类型
	CacheMaxViews	否	**Numeric**：表示由保存在缓存历史记录中的视图数量上限。值为 -1 表示不加限制。默认值为 0。此选项只有在缓存参数设置为 ON 时应用。
miLayerInfoTypeGeodictUserName	Name	是	**String**：要添加的图层的名称 （存储在 Geodictionary 中）。
miLayerInfoTypeTemp	Name	否	**String**：添加的图层的名称。 默认值为 LayerX。
	Fields	否	**Fields 集合**：指定表将拥有的列。
	图元	如果没有字段则为是。	**Features 集合**：指定用于填充表的列 默认为 none。
	TableStorageType	否	**String**：默认为 Native。 有效选项：Native、 MemTable
miLayerInfoTypeNewTable	FileSpec	是	**String**：用于新表的 .tab 文件的名称。
	Name	否	**String**：添加的图层的名称 默认 （如未指定）为 LayerX，其中 X 等于当前在地图中使用的图层数 + 1。
	Fields	否	**Fields 集合**：指定表将拥有的列。
	图元	如果没有字段则为是。	**Features 集合**：指定用于填充表的列 默认为 none。
	OverwriteFile	否	**String**：如果为 1，则由 FileSpec 参数指定的文件将被覆盖 （如果当前存在）。 如果为 0，则将不覆盖该文件。 **注**：所有非 0 的值均将强制覆盖该文件。
	TableStorageType	否	**String**：默认为 Native。有效选项： Native、 Access **注**：对于 Access 的表存储类型， DECIMAL 类型的字段将更改为 FLOAT。
	AccessVersion	否	**String**：默认为 V2000。有效选项： V9597、 V2000。 **注**：此参数只在表存储类型为 Access 时应用。

对于 LayerInfoType，不是 miLayerInfoTypeUserDraw 或 miLayerInfoTypeRaster，支持以下选项：

LayerInfoType	参数	必需	类型
除 miLayerInfoTypeUserDraw 或 miLayerInfoTypeRaster 之外的任意其他对象	AutoCreateDataset	否	**Numeric**：如果为 1，则 miDatasetLayer 的数据集将在图层添加之后添加。如果为 0 （默认），则不添加任何数据集。
除 miLayerInfoTypeUserDraw 或 miLayerInfoTypeRaster 之外的任意其他对象	DatasetName	否	**String:** 如果 AutoCreateDataset 参数为 1，则将是新的 Dataset 的名称。如果未指定，则使用默认名称。

此参数可以采用以下任意 LayerInfoType 类型：

LayerInfoType	参数	必需	类型
任意	可见	否	Bool：指定图层在添加后是否可见。 默认为 true。

注解

要改善性能，可以将 TableStorageType 指定为 memtable。

MapX 不再允许图层名重复。当开发人员显式指定图层名 （正如 **Layers.AddServerLayers**、**Layers.AddUserDrawLayer** 和 **Layers.Add** 的名称参数所示），如果名称不唯一，则将会报错。

miLayerInfoTypeTemp 是临时图层，在 MapX 控件关闭之前保存，然后图层将被废弃。该图层不存储到磁盘。用户可以指定图层名，但不能指定 filespec。

miLayerInfoTypeNewTable 是一个图层，可从外部数据生成 （例如查询结果）。

对于 miLayerInfoTypeTemp 和 miLayerInfoTypeNewTable：

- 必须使用 Fields 集合或 Features 集合，两者均可使用。
- 如果只传递 Features 集合，则 MapX 将尝试从 Features 所源自的表来获取列信息。
- 如果传递 Features 和 Fields 集合，则 MapX 将尝试从 Features 所源自的表来获取列信息。
- 如果只传递 Fields 集合，则检查用于 miDatasetLayer 类型的 Dataset。在发现之后，则对照 miDatasetLayer 中的列来检查传入的 Fields。如果 miDatasetLayer Dataset 没有出现，则只在传递入的 Fields 信息上创建表。
- Fields 默认应该为 Features 集合的源表中发现的列。

使用 LayerInfo 对象添加引用光栅、远程表等的 .tab 文件时，使用 LayerInfo 类型 miLayerInfoTypeTab。LayerInfo 类型 LayerInfoTypeRaster 专用于添加没有关联的 .tab 文件的自注册光栅文件。 LayerInfo 类型 LayerInfoTypeServerLayer 专用于添加没有关联的 .tab 文件的远程表文件。

对象属性

- **LayerInfo.Type 属性**

- **LayerInfo.Parameter 属性**

对象方法

- **LayerInfo.AddParameter 属性**

另请参阅

Layers.Add 方法，第 365 页

使用缓存，第 166 页

MapX 字段命名规范，第 507 页

LayerInfo.Type 属性

目的

这是读写属性，设置为 **LayerTypeConstants** 之一。

另请参阅

LayerInfo.AddParameter 属性，第 377 页

Layers.Add 方法，第 365 页

LayerInfo.AddParameter 属性

目的

Addparameter 方法添加 Layers.Add 方法的给定调用必需的参数集。必要参数由 LayerInfo 对象的类型属性确定。

语法

OBJECT.**AddParameter** (name, value)

部分	说明
OBJECT	表示 LayerInfo 对象。
Name	参数的名称。
value	参数的值。

注解

MapX 不再允许图层名重复。当开发人员显式指定图层名 （在 layerinfo 的名称参数 addserverlayer 或 adduserdrawlayer 中）时，如果名称不唯一，则将会报错。

另请参阅

Layers.Add 方法，第 365 页

LayerInfo.Parameter 属性

目的

此读写属性取参数，是表示参数名称的字符串。在检索之后， Parameter 属性返回一个变量，其类型是参数插入时的任意类型。您还可以设置、添加、插入或更改此属性的参数。

示例

要取现有参数的值 （例如 CoordSys），请执行以下操作：

```
Set oCoordSys = oLayerInfo.Parameter("CoordSys")
```

要更改现有或添加新参数 （例如 Style），可使用 AddParameter 方法或使用 Parameter 属性，如下所示：

```
Set oLayerInfo.Parameter("Style") = Map1.Layers("MyLayer").Style
```

注：设置只在设置对象时使用。其他属性，例如字符串或数字，不需要使用 Set 语句。

Legend 对象

目的

每个 Theme 对象均有 Legend 对象 (Theme.Legend)。 Legend 对象包含控制主题图例显示的属性。每个 ThemeCategory 对象 （RangeCategory、 IndividualValueCategory 或 MultiVarCategory）在包含在 LegendText 对象的图例中都有一个条目。

对象属性

- **Legend.BodyTextStyle 属性**
- **Legend.Compact 属性**
- **Legend.CompactTitle 属性**
- **Legend.CompactTitleStyle 属性**
- **Legend.CurrencyFormat 属性**
- **Legend.Height 属性**
- **Legend.Left 属性**
- **Legend.LegendTexts 属性**
- **Legend.PaperHeight 属性**
- **Legend.PaperWidth 属性**
- **Legend.ShowCount 属性**
- **Legend.ShowEmptyRanges 属性**
- **Legend.SubTitle 属性**
- **Legend.SubTitleStyle 属性**
- **Legend.Title 属性**
- **Legend.TitleStyle 属性**
- **Legend.Top 属性**
- **Legend.Visible 属性**
- **Legend.Width 属性**

对象方法

- **Legend.ExportLegend 方法**
- **Legend.LegendDlg 方法**
- **Legend.PrintLegend 方法**

Legend.BodyTextStyle 属性

目的

这是一个 Style 对象，控制用于图例主体的文本样式，可以设置为现有 Style 对象，或者您可以分别设置 style 属性。

另请参阅

Style 对象，第 463 页

Legend.Compact 属性

目的

控制图例是压缩 （1 行）还是完全大小。该属性是一个布尔值，默认值为 Map.PreferCompactLegends 的设置。

另请参阅

Map.PreferCompactLegends 属性，第 412 页

Legend.CompactTitle 属性

目的

包含压缩图例标题文本字符串。当图例为压缩大小时显示。是 String 值，默认标题是在主题地图创建时创建的。

Legend.CompactTitleStyle 属性

目的

这是一个 Style 对象，控制压缩标题的文本样式。这是一个 Style 对象，并且可以设置为现有样式对象，或者您可以分别设置样式属性。

另请参阅

Style 对象，第 463 页

Legend.CurrencyFormat 属性

目的

控制数字是否使用货币格式或标准数字格式。该属性是一个布尔值，默认值为 False。

Legend.ExportLegend 方法

目的

将 Legend 的图像导出到文件或剪贴板。

语法

OBJECT.**ExportLegend** (Destination, Format)

部分	说明
OBJECT	表示 Legend 对象。
Destination	放置输出结果的文件规格，例如 C:\Temp\Legend.bmp。如果指定 CLIPBOARD 而不是文件路径，则图像将置于剪贴板。
Format	输出格式。这将取 **ExportFormatConstants** 的值。注：如果指定 miFormatGIF（生成 GIF 文件），请读取用于 GIF 文件用户的许可要求。如果您将输出指定到 CLIPBOARD，则格式必须为元文件或位图。

另请参阅

Map.ExportMap 方法，第 401 页

Legend.Height 属性

目的

包含使用屏幕单位的图例高度。这是一个单精度值，是只读属性。

Legend.Left 属性

目的

包含图例的左边缘的位置的屏幕单位，相对于地图控制的左上角。这是单精度值。

注：您可以将图例以某种形式拖放到 MapX 控件之外，并且不能将其拖回 （如果在图例离开屏幕时释放鼠标按钮）。

但是如果可能，可以通过编程来重设在地图范围之内的对象的 Left 和 Top 属性。这同样适用于地图标题和注释。

Legend.LegendDlg 方法

目的

显示对话框，用户可以在其中调整图例设置 （通过用户界面来更改图例对象的属性值）。如果用户单击“确定”，则在对话框中所做的更改将立即应用于地图。

语法

[Boolean=]OBJECT.**LegendDlg**([HelpFile] [HelpID])

部分	说明
OBJECT	表示 Legend 对象。
HelpFile	HelpFile 是可选参数，是 .hlp 文件的路径名，包含用于对话框的帮助主体。
HelpID	HelpID 是可选参数，引用给定 .hlp 文件中的特定帮助主题 ID。

注解

如果没有给定任何给定参数，则帮助按钮将不会出现在对话框上。

如果用户单击 “确定”，则 LegendDlg 的返回值为 True ；如果用户单击 “取消”则为 False。

Legend.LegendTexts 属性

目的

用于图例的 Legendtext 对象的集合。范围、单值、饼图、条形图和点密度主题对于每个类别均有一个 LegendText 项目。分级符号有三种。

另请参阅

LegendText 对象和 LegendTexts 收集，第 386 页

Legend.PaperHeight 属性

目的

包含使用 Map.PaperUnit 单位的图例纸上高度。这是一个单精度值，是只读属性。

另请参阅

Map.PaperUnit 属性，第 412 页

Legend.PaperWidth 属性

目的

包含使用 Map.PaperUnit 单位的图例纸上宽度。这是一个单精度值，是只读属性。

另请参阅

Map.PaperUnit 属性，第 412 页

Legend.PrintLegend 方法

目的

将指定矩形中的图例打印到指定设备上下文。

语法

OBJECT.**PrintLegend** (hDC, x, y, w, h)

部分	说明
OBJECT	表示 Legend 对象。
hDC	打印机设备上下文。Can be any device context.
x	采用 HIMETRIC 单位的左上角 X。
Y	采用 HIMETRIC 单位的左上角 Y。
w	采用 HIMETRIC 单位的宽度。
h	采用 HIMETRIC 单位的高度。

注解

当前图例的绘制符合给定的矩形。在维持宽高比时可以获取最佳结果。

Legend.ShowCount 属性

目的

控制是否在图例上显示范围的计数部分。该属性取布尔值，默认值为 True。

Legend.ShowEmptyRanges 属性

目的

这是读写布尔值属性，确定是否在图例中显示空范围 （范围不包含任何图元）。默认值为 false。

Legend.SubTitle 属性

目的

这是读写字符串值，包含图例小标题文本字符串。当图例为正常大小时显示。默认的小标题在主题地图创建时创建。

Legend.SubTitleStyle 属性

目的

这是一个读写样式对象，控制图例小标题的文本样式。这是一个 Style 对象，并且可以设置为现有样式对象，或者您可以分别设置样式属性。

另请参阅

Style 对象，第 463 页

Legend.Title 属性

目的

包含图例标题文本字符串。当图例为正常大小时显示。这是字符串值。默认标题在主题地图创建时创建。

Legend.TitleStyle 属性

目的

这是一个 Style 对象，控制标题的文本样式。这是一个 Style 对象，并且可以设置为现有样式对象，或者您可以分别设置样式属性。

另请参阅

Style 对象，第 463 页

Legend.Top 属性

目的

此属性包含图例的顶边缘的位置的屏幕单位，相对于地图控制的左上角。这是单精度值。

注：您可以将图例以某种形式拖放到 MapX 控件之外，并且不能将其拖回 （如果在图例离开屏幕时释放鼠标按钮）。

但是如果可能，可以通过编程来重设在地图范围之内的对象的 Left 和 Top 属性。这同样适用于地图标题和注释。

Legend.Visible 属性

目的

控制图例是否可见。该属性是一个布尔值，默认值为 True。

Legend.Width 属性

目的

包含使用屏幕单位的图例宽度。这是一个单精度值，是只读属性。

LegendText 对象和 LegendTexts 收集

目的

每个主题都有一个称为 LegendTexts 的 LegendText 对象集合。图例的每个对象 （在标题下的每个行）都在此集合中具有 LegendText 对象，可以分别控制。

- **miThemeRanged** 对于每个范围 （每个 RangeCategory）都具有一个 LegendText。
- **miThemeIndividualValue**、 **miThemePieChart** 和 **miThemeBarChart** 对于每个类别 （每个 IndividualValueCategory 或每个 MultiVarCategory）都具有 oneLegendText。
- **miGradSymbol** 具有 3 个 LegendTexts － 一个示例大小一个。
- **miDotDensity** 具有一个 LegendText。

对象属性

- **LegendText.Text 属性**
- **LegendText.Visible 属性**

集合属性

- **LegendTexts.AllOthersText 属性**
- **LegendText.AutoGenerate 属性**
- **LegendTexts.Count 属性**
- **LegendTexts.Item 属性**

另请参阅

Theme 对象和 Themes 集合，第 481 页

LegendTexts.AllOthersText 属性

目的

此属性从集合返回 LegendText 对象。此 LegendText 对象和 RangeCategories 集合或 IndividualValueCategories 集合的 AllOthersCategory 属性关联。您可以设置 AllOthersText 属性的属性，类似于任何其他 LegendText 对象。

LegendText.Text 属性

目的

包含用于图例中的范围或类别的文本。是字符串值，默认值是在主题地图创建时创建的。图例文本是在 Theme 对象重新计算时自动重新计算的（例如，范围的数量更改为 6），除非 LegendTexts.AutoGenerate 属性设置为 False。

另请参阅

Theme 对象和 Themes 集合，第 481 页

Theme.AutoRecompute 属性，第 481 页

LegendText.Visible 属性

目的

控制范围是否对于图例可见。该属性取布尔值，默认值为 True。

LegendText.AutoGenerate 属性

目的

控制 LegendText 值在 Theme 对象重新计算时是否应该由 MapX 重新计算。该属性是一个布尔值，默认值为 True。如果要修改 LegendText 的值，应该将此设置为 False，以便 MapX 不覆盖所作更改。

另请参阅

Theme.AutoRecompute 属性，第 481 页

LegendTexts.Count 属性

目的

集合中的 LegendText 对象数。这是一个只读属性。

LegendTexts.Item 属性

目的

此属性从集合检索特定的 LegendTexts 对象。必须指定文本字符串。

Map 对象

地图是 MapInfo MapX 中的基本构建模块。每个地图由 Dataset、 Layer 和 Annotations 对象和集合定义。

对象属性

- **Map.Annotations 属性**
- **Map.AreaUnit 属性**
- **Map.AutoRedraw 属性**
- **Map.BackColor 属性**
- **Map.Bounds 属性**
- **Map.CenterX 属性**
- **Map.CenterY 属性**
- **Map.CurrentTool 属性**
- **Map.Dataset 属性**
- **Map.DatasetGeoField 属性**
- **Map.Datasets 属性**
- **Map.DatasetTheme 属性**
- **Map.DefaultConversionResolution 属性**
- **Map.DefaultStyle 属性**
- **Map.DisplayCoordSys 属性**
- **Map.ExportSelection 属性**
- **Map.FeatureEditMode 属性**
- **Map.FeatureFactory 属性**
- **Map.GeoDictionary 属性**
- **Map.GeoSet 属性**
- **Map.GeoSets 属性**
- **Map.GeoSetWidth 属性**
- **Map.hWnd 属性**
- **Map.InfotipSupport 属性**
- **Map.InfotipPopupDelay 属性**
- **Map.IsPointVisible 方法**
- **Map.Layers 属性**
- **Map.MapPaperHeight 属性**
- **Map.MapPaperWidth 属性**
- **Map.MapScreenHeight 属性**
- **Map.MapScreenWidth 属性**
- **Map.MapUnit 属性**
- **Map.MatchNumericFields 属性**
- **Map.MatchThreshold 属性**
- **Map.MaxSearchTime 属性**

- **Map.MouseIcon 属性**
- **Map.MousePointer 属性**
- **Map.MouseWheelSupport 属性**
- **Map.NumericCoordSys 属性**
- **Map.PanAnimationLayer 属性**
- **Map.PaperUnit 属性**
- **Map.PreferCompactLegends 属性**
- **Map.RedrawInterval 属性**
- **Map.ReuseEquivalentOnRestore 属性**
- **Map.Rotation 属性**
- **Map.SearchPath 属性**
- **Map.SelectionStyle 属性**
- **Map.SnapTolerance 属性**
- **Map.SnapToNodeSupport 属性**
- **Map.Title 属性**
- **Map.TitleText 属性**
- **Map.Version 属性**
- **Map.WaitCursorEnabled 属性**
- **Map.Zoom 属性**

对象方法

- **Map.AboutBox 方法**
- **Map.ClipLine 方法**
- **Map.CliplineV 方法**
- **Map.ConvertCoord 方法**
- **Map.ConvertCoordP 方法**
- **Map.ConvertCoordV 方法**
- **Map.CreateCustomTool 方法**
- **Map.Distance 方法**
- **Map.ExportMap 方法**
- **Map.IsPointVisible 方法**
- **Map.Pan 方法**
- **Map.PrintMap 方法**
- **Map.PropertyPage 方法**
- **Map.Refresh 方法**
- **Map.SaveMapAsGeoset 方法**
- **Map.SetSize 方法**
- **Map.ZoomTo 方法**

另请参阅

Dataset 对象和 Datasets 集合，第 257 页

Layer 对象和 Layers 集合，第 340 页

Annotation 对象和 Annotations 集合，第 238 页

Map.AboutBox 方法

目的

此方法用于显示 “关于”对话框，包含例如 MapX 版本和适当开发人员的信息。

语法

OBJECT.**AboutBox**

Map.Annotations 属性

目的

用于地图的注释集合。

另请参阅

Annotation 对象和 Annotations 集合，第 238 页

Map.AreaUnit 属性

目的

这是在 Area 方法 / 属性中使用的单位。取值为 **AreaUnitConstants** 值。默认值为 miUnitSquareMile。

Map.AutoRedraw 属性

目的

这一临时属性用于关闭屏幕更新。取布尔值，是读写属性。

注解

将 AutoRedraw 设置为 FALSE 以禁用屏幕更新，或设置为 TRUE 来重新开始正常的屏幕更新。

在属性或方法导致地图做出更改时 （例如 Map.Zoom），地图将会重新绘制。如果要对地图做出若干更改，此属性将允许您禁用重新绘图，做出若干更改，然后再次启用重新绘制。这样可以避免不必要的多次重新绘制。设置为 TRUE 时，将会绘制整个地图。

Map.BackColor 属性

目的

此属性控制在绘制地图之前消除地图的哪些颜色和背景。必须指定 OLE_COLOR 值。可以是特定的纯色，或 Windows 系统颜色之一，例如 “Window 背景色”。（大部分容器和 Visual Basic 一样，在其属性页中显示特殊的颜色采集器对话框，用于 OLE_COLOR 类型的属性。）

注解

由于 MapX 只有一个顶层颜色属性，该颜色默认为选中，并且禁用属性名组合框。您可以选择相应颜色之一，所选颜色将会突出显示，系统颜色组合为空。此外，您还可以选择系统色。

另请参阅

OLE_COLOR 值，第 631 页

Map.Bounds 属性

目的

Map.Bounds 属性 （早期版本中称为 Map.MBR）可用于设置地图坐标中的地图窗口的界限。取 Rectangle 对象。

另请参阅

Rectangle 对象，第 437 页

Map.CenterX 属性

目的

此属性设置地图中心的当前 X 坐标。此属性是一个双精度值，表示经度 （负值代表西半球）。

注解

CenterX 属性在加载新的 Geoset 时设置。

另请参阅

Map.CenterY 属性，第 391 页

Map.ZoomTo 方法，第 419 页

Map.CenterY 属性

目的

此属性设置地图中心的当前 Y 坐标。此属性是一个双精度值，表示纬度。

注解

CenterY 属性在加载新的 Geoset 时设置。

另请参阅

Map.CenterY 属性，第 391 页

Map.ZoomTo 方法，第 419 页

Map.ClipLine 方法

目的

此方法接受表示地图上的假想线端点的参数。如果线的任何部分位于视图矩形之内， ClipLine 将调整参数以便生成的线不会位于视图矩形之外。

语法

[Boolean=]OBJECT**.ClipLine** (X1, Y1, X2, Y2)

部分	说明
OBJECT	表示 Map 对象。
X1	双精度 （通过引用）。对于输入：未截短的线的第一个端点的经度坐标。对于输出：未截短的线的第一个端点的经度坐标。
Y1	双精度 （通过引用）。对于输入：未截短的线的第一个端点的纬度坐标。对于输出：未截短的线的第一个端点的纬度坐标。
X2	双精度 （通过引用）。对于输入：未截短的线的第二个端点的经度坐标。对于输出：未截短的线的第二个端点的经度坐标。
Y2	双精度 （通过引用）。对于输入：未截短的线的第二个端点的纬度坐标。对于输出：未截短的线的第二个端点的纬度坐标。

注解

所有参数均通过引用传递，并且应该位于 Map.NumericCoordSys 坐标系中。

如果线的任意部分位置视图矩形之内，则 ClipLine 的返回值为 True，否则为 False。

如果线完全位于 “地图”窗口之内，则将不会影响到任何参数，并且 ClipLine 将会返回 True。如果只是线的部分位于地图窗口之内，则坐标将会做出调整，以便它们构成视图窗口中的线的部分，此时 ClipLine 将返回 True。如果线完全位于 “地图”窗口之外，则将不会影响到坐标，并且 ClipLine 将会返回 False。

另请参阅

Map.CliplineV 方法，第 393 页

Layers.AddUserDrawLayer 方法，第 367 页

Map.NumericCoordSys 属性，第 411 页

Map.CliplineV 方法

目的

此方法接受表示地图上的假想线端点的参数。如果线的任何部分位于视图矩形之内， ClipLineV 将调整参数以便生成的线不会位于视图矩形之外。它和 Map.ClipLine 方法相同，除了其具有 Variant 参数，而不是双精度值。

语法

[Boolean=] OBJECT.**ClipLineV**(X1, Y1, X2, Y2)

部分	说明
OBJECT	表示 Map 对象。
X1	变量 （通过引用）。对于输入：未截短的线的第一个端点的经度坐标。对于输出：未截短的线的第一个端点的经度坐标。
Y1	变量 （通过引用）。对于输入：未截短的线的第一个端点的纬度坐标。对于输出：未截短的线的第一个端点的纬度坐标。
X2	变量 （通过引用）。对于输入：未截短的线的第二个端点的经度坐标。对于输出：未截短的线的第二个端点的经度坐标。
Y2	变量 （通过引用）。对于输入：未截短的线的第二个端点的纬度坐标。对于输出：未截短的线的第二个端点的纬度坐标。

所有参数均为由引用传递的变量。它们预期包含数据值，并且应该位于 Map.NumericCoordSys 坐标系之中。如果任意参数不能转换为数字，则可能会报告异常错误代码 1111。

如果线的任意部分位置视图矩形之内，则 ClipLineV 的返回值为 True，否则为 False。

如果线完全位于 “地图”窗口之内，则将不会影响到任何参数，并且 ClipLineV 将会返回 True。如果只是线的部分位于地图窗口之内，则坐标将会做出调整，以便它们构成视图窗口中的线的部分，此时 ClipLineV 将返回 True。如果线完全位于 “地图”窗口之外，则将不会影响到坐标，并且 ClipLineV 将会返回 False。

另请参阅

Map.ClipLine 方法，第 392 页

Map.NumericCoordSys 属性，第 411 页

Map.ConvertCoord 方法

目的

将地图坐标转换为屏幕显示坐标，或将屏幕坐标转换为地图坐标，具体取决于在此方法中传递的 Dir 参数。屏幕坐标以像素指定。

语法

OBJECT.**ConvertCoord** (ScreenX, ScreenY, MapX, MapY, Dir)

部分	说明
OBJECT	表示 Map 对象。
ScreenX	以像素指定的屏幕 X 坐标。Single value.
ScreenY	以像素指定的屏幕 Y 坐标。单精度值。
MapX	Map x 坐标 （经度）。双精度值。
MapY	Map Y 坐标 （纬度）。双精度值。
Dir	转换坐标的方向，是从地图到屏幕，还是屏幕到地图。取值为 **ConversionConstants** 值。

是提供的屏幕坐标或地图坐标，以及其他坐标，根据指定的 Dir 填充。

Map.ConvertCoordP 方法

目的

此方法去**屏幕 / 地图** X 和 Y 坐标和方向 （miScreenToMap 或 miMapToScreen），返回 X 和 Y 属性相应于转换的**地图 / 屏幕**坐标的点。

语法

Object.ConvertCoordP(X , Y , Dir)

部分	说明
X	变量：是屏幕 / 地图坐标 X。
Y	变量：是屏幕 / 地图坐标 Y。
Dir	转换坐标的方向，是从地图到屏幕，还是屏幕到地图。取值为 **ConversionConstants** 值。

Map.ConvertCoordV 方法

目的

将地图坐标转换为屏幕显示坐标，或将屏幕坐标转换为地图坐标，具体取决于在此方法中传递的 Dir 参数。 ConvertCoord 方法的变量版本。屏幕坐标以像素指定。

语法

OBJECT.**ConvertCoordV** (ScreenX, ScreenY, MapX, MapY, Dir)

部分	说明
OBJECT	表示 Map 对象。
ScreenX	以像素指定的屏幕 X 坐标。变量值。
ScreenY	以像素指定的屏幕 Y 坐标。变量值。
MapX	Map x 坐标 （经度）。变量值。
MapY	Map Y 坐标 （纬度）。变量值。
Dir	转换坐标的方向，是从地图到屏幕，还是屏幕到地图。取值为 **ConversionConstants** 值。

是提供的屏幕坐标或地图坐标，以及其他坐标，根据指定的 Dir 填充。

Map.CreateCustomTool 方法

目的

此方法创建了一个定制工具，在使用时发送 ToolUsed 事件。

注：如果您创建了圆类型定制工具，并且在 MapX 控件的 theTool_Used 事件中使用传递给事件的值进行 SelectByRadius，则结果可能和使用半径选择工具选择对象得到的结果不同。在使用控制工具的情况下， **SelectByRadius** 方法将不会完全选择位于控件上绘制的圆下的内容，其原因在于此圆是一个近似的圆，不考虑地图的投影。使用实际的半径选择工具做出的选择和该圆中内容完全相同。

语法

OBJECT.**CreateCustomTool** (ToolNumber, Type, Cursor, [ShiftCursor] , [CtrlCursor], [InfoTips])

部分	说明
OBJECT	表示 Map 对象。
ToolNumber	用于在此后引用它的工具的数量。此值必须是介于 1 和 999 之间的整数。
Type	描述工具行为的类型。取值为 **ToolTypeConstants** 的值。
Cursor	在工具创建时的光标形状是 CurrentTool，并且光标位于地图之上。取值为 **CursorConstants** 的值或定制光标文件名。
ShiftCursor	变量：**CursorConstants** 值，表示应该在按住 SHIFT 键时出现的光标。可选，如果省略，则 SHIFT 键对于光标没有任何影响。
CtrlCursor	变量：**CursorConstants** 值，表示应该在按住 CTRL 键时出现的光标。可选，如果省略，则 CTRL 键对于光标没有任何影响。

部分	说明
InfoTips	布尔值：如果要显示 InfoTips，则设置为 true。默认值为 false。

注解

可以创建执行特定行为的定制工具。例如，可以创建工具，设置显示有关单击对象信息的行为。在此，我们将定义 miToolTypePoint 类型的工具，并在 ToolUsed 事件处理程序之中，编写代码以显示有关点击对象点下的对象的信息 （使用 Layer.SelectByPoint）。

另请参阅

Map.CurrentTool 属性，第 396 页

Map.CurrentTool 属性

目的

此属性设置为当前工具。取值为 **ToolConstants** 值。默认值为 miArrowTool。

注解

不同的工具将使鼠标可以执行多种不同任务。例如，如果当前工具被设置为 miLabelTool，则在您单击鼠标时，它会给该特定地图对象加标签。鼠标光标将基于您所使用的工具进行更改。

如果已经定义定制工具，可通过指定其 ToolNumber 来将其设置为 **CurrentTool**。

另请参阅

Map.CreateCustomTool 方法，第 395 页

Map.Dataset 属性

目的

此属性指定要添加的 Dataset。只在设计阶段可用，在运行时不可用。此属性仅限 Microsoft Visual Basic。

注：此方法在 MapX Mobile 中不可用。

注解

Visual Basic 将会显示在用于 MapX 的 Visual Basic 属性页中的当前表单上的所有数据控件的列表。选择此列表中用于 Dataset 属性的 Data 控件将导致 MapX 在运行时使用由 DataControl 定义的数据创建 Dataset。

这一点和 Datasets.Add 方法类似。但是，由于不使用任何参数，它将假定用于其余参数的默认行为。GeoField 参数可以使用 Map.DatasetGeoField 属性指定。您还可以选择对于 MapX 数据属性页上的每个字段，要使用的字段 （列）和聚集方法。

另请参阅

Datasets.Add 方法，第 263 页

Map.DatasetGeoField 属性，第 397 页

Map.DatasetGeoField 属性

目的

此属性指定在 Map.Dataset 属性中指定的哪一 Dataset 字段包含地理信息。此字符串指定字段的长度。

注：此方法在 MapX Mobile 中不可用。

注解

这和 Datasets.Add 方法的 GeoField 参数类似，但是指定 GeoField 用于设计阶段指定的 Dataset。

另请参阅

Map.Datasets 属性，第 397 页

Datasets.Add 方法，第 263 页

Map.Datasets 属性

目的

用于地图的 Datasets 集合

另请参阅

Dataset 对象和 Datasets 集合，第 257 页

Map.DatasetTheme 属性

目的

此属性指定用于由 Map.Dataset 属性指定的 Dataset 的主题类型。取值为 **ThemeTypeConstants** 值。默认为 miAutoThemeType。

注：此方法在 MapX Mobile 中不可用。

注解

如果在设计阶段 （Map.Dataset 属性）指定 Dataset，则您可以绘制数据的主题地图。此方法控制主题样式。此方法和 Themes.Add 方法类似，但是在设计阶段可用。您还可以选择主题在属性页的 MapX 主题选项卡上要基于的一个或多个字段。

如果没有设置 Dataset 属性，则将会忽略 DatasetTheme 属性。

另请参阅

Themes.Add 方法，第 485 页

Map.Datasets 属性，第 397 页

Map.DefaultConversionResolution 属性

目的

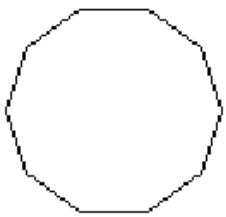
读写系统设置，控制在将特殊 MapInfo 图元类型 （例如弧）转换为区域或多义线，或生成圆、椭圆、弧或缓存时的分辨率 （例如节点数）。是长整形整数，可以取 3 和 32,763 之间的任意值。

注解

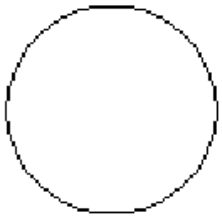
允许您显示指定可选分辨率参数的各种方法 （如下所示）。如果不指定分辨率值， MapX 将使用由 DefaultConversionResolution 属性指定的分辨率设置。

此属性将影响缓存、圆、椭圆和弧的生成。在生成这些类型的图元时，您可以控制图元的分辨率（有时称为 “平滑度”）。例如，下表显示了两个不同的区域，一个包含 10 个节点，另一个包含 30 个节点。

带有 10 个节点的圆



带有 30 个节点的圆



增加分辨率将生成更为平滑的区域，但是生成用时较长，并且生成的对象需要更多的存储空间。

此属性还控制在 MapInfo 表包含弧、圆、矩形或圆整矩形图元时的分辨率，并且 MapX 将图元转换为区域或多义线。

另请参阅

FeatureFactory.BufferFeatures 方法，第 288 页

FeatureFactory.CreateArc 方法，第 290 页

FeatureFactory.CreateCircularRegion 方法，第 293 页

FeatureFactory.CreateEllipticalRegion 方法，第 294 页

Layer.SearchWithinDistance 方法，第 359 页

Map.DefaultStyle 属性

目的

这是一个 Style 对象，在使用 Annotation.AddSymbol 或 Annotation.AddText 方法创建符号和文本对象时具有其样式

另请参阅

Style 对象，第 463 页

Map.DisplayCoordSys 属性

目的

返回一个读写 CoordSys 对象 （坐标系），控制地图显示方式。

注：如果一个地图包含一个或多个光栅图图层，则 MapX 自动在最可见的光栅图指定的投影中显示该地图。如果具有不同投影的其它光栅图成为最可见的光栅图，则坐标系统可以随着地图视图的更改 （由于缩放或平移）而进行更改。在此情况下，您无法更改地图的显示坐标系统。

注解

Map.DisplayCoordSys 表示在其中显示地图的坐标系 （或 “投影”）。默认情况下，地图的坐标系由 geoset 设置。通过修改地图的 DisplayCoordSys （通过 CoordSys.Set 方法），您可以在另一坐标系中显示地图。

Map.DisplayCoordSys 属性只影响地图的外观。要访问用于解释或返回地图坐标的坐标系，可使用 Map.NumericCoordSys 属性。

另请参阅

第 177 页的第 13 章：“使用坐标系统”

CoordSys 对象，第 252 页

Map.NumericCoordSys 属性，第 411 页

Map.Distance 方法

目的

此方法计算地图上指定两点之间的距离。该距离将返回为双精度值，采用由 MapUnit 属性指定的单位。

语法

[distance =]OBJECT.**Distance**(x1,y1,x2,y2)

部分	说明
OBJECT	表示 Map 对象。
X1	第一个点的 X 坐标 （经度）。双精度值。
Y1	第一个点的 Y 坐标 （纬度）。双精度值。
X2	第二个点的 X 坐标 （经度）。双精度值。
Y2	第二个点的 Y 坐标 （纬度）。双精度值。

注解

距离使用大圆计算法计算。

另请参阅

CoordSys 对象，第 252 页

Map.NumericCoordSys 属性，第 411 页

Map.MapUnit 属性，第 405 页

Map.DynamicSelectionSupport 属性

目的

此布尔值属性控制是否启用动态选择。设置为 True，用户将看到添加到地图的选择突出显示，此时将使用半径选择和矩形选择工具。设置为 False 时，选择突出显示将会在用户使用半径选择和矩形选择工具时释放鼠标按钮时绘制。

注解

此属性默认为 false。该模式仅适用于矩形和圆形选择工具。工具本身的行为是相同的。

Map.EditableLabels 属性

目的

这是有关 Map 对象的布尔值属性。将其设置为 true 允许通过用户界面，使用 MapX 选择工具 (miSelectTool)，选择和移动标签。将其设置为 false 将会禁用通过用户界面进行的编辑。

注解

Labels 集合始终可用于通过程序编辑标签，无论 Map.EditableLabels 取什么值。 Map.EditableLabels 控制用户是否可以使用选择工具来编辑标签 （选择和移动）。

在此属性设置为 True 时，将会触发 LabelChanged，并且用户使用选择工具来选择或移动标签。

另请参阅

Label 对象和 Labels 集合，第 328 页

Map.ExportMap 方法

目的

此方法用于导入地图图像。用于台式机的 MapX 可以导出到以下格式 （如果使用 MapX Mobile 则参阅注释）：

JPG	**TIF**	**GIF**
PNG	**WMF**	**PSD**
BMP		

注：如果使用 MapX Mobile，可能只需要导出为 BMP。尝试导出为其他格式将会出错。

语法

OBJECT.**ExportMap** (Location, Format, [Width], [Height])

部分	说明
OBJECT	表示 Map 对象。
Location	放置输出结果的文件规格。如果使用关键字 CLIPBOARD，则图像将置于剪贴板。**注：**如果使用 MapX Mobile，则只能将 BMP 格式的地图粘贴到给定位置。
Format	输出格式。这将取 **ExportFormatConstants** 的值。
Width	输出宽度取双精度值，以纸张单位指定宽度 (Map.PaperUnit)。此参数可选，如未指定，则将使用 Map.MapPaperWidth。 **注：**如果使用 MapX Mobile，则参数将被忽略，以便导出的地图和手持设备屏幕的大小相同。
Height	输出高度取双精度值，以纸张单位指定高度 (Map.PaperUnit)。此参数可选，如未指定，则将使用 Map.MapPaperHeight。 **注：**如果使用 MapX Mobile，则参数将被忽略，以便导出的地图和手持设备屏幕的大小相同。

注解

如果选择了地图上的图元，可能不需要输出选择突出显示，请参阅 Map.ExportSelection 属性。

另请参阅

Map.ExportSelection 属性，第 402 页

Map.PaperUnit 属性，第 412 页

Map.ExportSelection 属性

目的

此属性返回或设置布尔值属性，控制是否在输出中包含选择的突出显示。

如果设置为 True，则由 **Map.ExportMap 方法** 或 **Map.PrintMap 方法**生成的输出将包含任意当前选择突出显示。如果设置为 False，则输出不包含突出显示。

要控制绘制选择突出显示的样式，可设置 **Map.SelectionStyle** 属性。

另请参阅

Map.ExportSelection 属性，第 402 页

Map.PrintMap 方法，第 412 页

Map.SelectionStyle 属性，第 416 页

Map.FeatureEditMode 属性

目的

此属性控制地图的编辑模式 （例如图元编辑 (miEditModeFeature) 或节点编辑 (miEditModeNode)）。节点编辑可以在设置为 **FeatureEditModeConstants** 之一时使用其他节点操作此属性默认为 miEditModeFeature。

Map.FeatureFactory 属性

目的

此属性返回 FeatureFactory 对象，支持各种用于创建和修改地图上的图元的方法。

注解

此属性返回 **FeatureFactory** 对象，该对象具有支持各种编辑任务创建的功能、生成缓冲区域、合并图元并测试图元之间的交集。

另请参阅

FeatureFactory 对象，第 288 页

Map.GeoDictionary 属性

目的

此只读属性指定了注册表条目的位置，用于确定 GeoDictionary 文件和默认的数据目录。

注解

在创建地图控件时， MapX 将查询注册表键
HKEY_LOCAL_MACHINE\Software\MapInfo\MapX\5.0\<GeoDictonary 的属性值 >

此注册表键的值指定两个项目：

- GeoDictionary 文件保存有关图层匹配信息的有关信息。（MapX 可以不使用 GeoDictionary。）
- 默认的数据目录。将搜索此目录，查找 GeoSet 文件 (*.gst)，该文件显示在属性页和 Map.GeoSets 选择中。

对于该注册表值有三种有效选项：

- **Geodictionary 文件的完整文件规格** （例如 “C:\Program Files\MapInfo MapX\Maps\geodict.dct”）。数据目录设置为由此键命名的 Geodictionary 文件之前的路径。内存中 geodictionary 是从 Geodictionary 文件初始化的，并且 Geodictionary 中包含的图层也可以自动与其进行匹配。添加到地图中的图层被添加到内存中的 geodictionary，这样就可以自动与其相匹配。
- **路径规格** （例如 C:\Program Files\MapInfo MapX\Maps\）。数据目录设置为此目录。内存中 geodictionary 被初始化为空。添加到地图中的图层仍被添加到内存中的 geodictionary，这样就可以自动与其相匹配。
- **空**。内存中 geodictionary 被初始化为空。添加到地图中的图层仍被添加到内存中的 geodictionary，这样就可以自动匹配。

Map.GeoDictionary 的默认值是 “GeoDictionary”，表示 MapX 将查询注册表条目
HKEY_LOCAL_MACHINE\Software\MapInfo\MapX\5.0\GeoDictionary。

另请参阅

Map.GeoSet 属性，第 403 页

Map.GeoSet 属性

目的

此读写属性设置 GeoSet。它将返回字符串值。要设置此属性，将使用 filespec （完整路径）或 GeoSet 的友好名称。

注解

GeoSet 是图层及其设置的集合。 GeoSet 可在设计阶段指定。如果在运行时设定，则它将先移除所有加载的图层和 Dataset，然后加载新的 GeoSet。要只去除所有加载的图层和 Datasets，不不加载新的 geoset，可指定空字符串。默认加载的 GeoSet 是 US.GST。

Map.GeoSets 属性

目的

只读属性，返回 Geoset 对象的集合。此集合包含 Maps 目录中的所有 geoset。

另请参阅

Geoset 对象和 Geoset 集合，第 322 页

Map.GeoSetWidth 属性

目的

这一只读属性表示 GeoSet 的宽度 （地图中所有图层的最大宽度）。这是一个双精度值，采用由 MapUnit 属性指定的单位。

此属性在需要将缩放设置为地图范围时非常实用。

Map.hWnd 属性

目的

这是 MapX 控件窗口的句柄。Only valid at Run time. 这是一个只读属性。

Map.InfotipSupport 属性

目的

此属性允许程序员控制 MapX 是否显示弹出式 InfoTips。该属性是一个布尔值，默认值为 True。

注解

当 InfoTipSupport 为真时，将弹出 InfoTips 用于以下工具：箭头工具、选择工具、标签工具，以及 Map.CreateCustomTool 参数 ShowInfoTips 设置为 True 的任意定制工具。

Infotips 只涉及到可选择图层。 It also searches the layer list from top to bottom. Therefore, if objects from one layer obscure another, you will only see infotips for uppermost objects.

另请参阅

Map.InfotipPopupDelay 属性，第 405 页

Map.CreateCustomTool 方法，第 395 页

Map.InfotipPopupDelay 属性

目的

此属性控制在 InfoTip 弹出之前，用户需要将鼠标指针置于对象之上的时间长短。此属性只在 Map.InfotipSupport 设置为 True 时适用。这是一个整数值，以毫秒指定。默认值为 500。

另请参阅

Map.InfotipSupport 属性，第 404 页

Map.IsPointVisible 方法

目的

此方法返回 true 或 false，即点 （地图坐标）是否在地图窗口的可见部分中。

语法

[bool=] OBJECT.**IsPointVisible** (X, Y)

部分	说明
OBJECT	表示 Map 对象。
X	地图的东西位置坐标 （纬度）。
Y	地图的南北位置坐标 （经度）。

Map.Layers 属性

目的

用于地图的 Layers 集合

另请参阅

Layer 对象和 Layers 集合，第 340 页

Map.MapUnit 属性

目的

此属性设置用于距离计算的单位。取值为 **MapUnitConstants** 值。默认单位为 miUnitMile。

Map.MapPaperHeight 属性

目的

地图控件的高度。取双精度值，以纸张单位 (**Map.PaperUnit**) 指定。这是一个只读属性。

Map.MapPaperWidth 属性

目的

地图控件的宽度。取双精度值，以纸张单位 (**Map.PaperUnit**) 指定。这是一个只读属性。

Map.MapScreenHeight 属性

目的

MapScreenHeight 和 **MapScreenWidth** 属性可以令图例定位更加轻松。这些属性包含地图控件的宽度和高度，以像素为单位。由于图例也包括其宽度和高度 （单位为像素），因此有助于轻松定位图例。

Map.MapScreenWidth 属性

目的

MapScreenHeight 和 **MapScreenWidth** 属性可以令图例定位更加轻松。这些属性包含地图控件的宽度和高度，以像素为单位。由于图例也包括其宽度和高度 （单位为像素），因此有助于轻松定位图例。

Map.MatchNumericFields 属性

目的

此属性控制数字和日期字段的匹配。此属性只在数据 （设计阶段）属性页上显示。如果 **MatchNumericFields** 为 true，则在 MapX 自动匹配时，数字字段将视为备选关键字 （和字母 / 数字或日期字段一起）。如果为 false，则在执行自动匹配时将不考虑数字列。此属性默认为 false。

如果指定日期或数字 GeoField in **Datasets.Add**，则无论 **MatchNumericFields** 是否为 TRUE，MapX 都将使用它。

Map.MatchThreshold 属性

目的

此属性是使用数据源匹配土册所必需的最低阈值百分比。

默认值为 80。

Map.MaxSearchTime 属性

目的

此属性设置在使用 **Datasets.Add** 方法引入数据时，用于自动 GeoColumn 检测和自动 GeoSet 检测的搜索时间。这是一个整数值，以秒指定时间。默认值为 30 秒。

Map.MilitraryGridReferenceFromPoint 方法

目的

此方法从地图的数字坐标系取 XY 点，并将其转换为军用网格参考系统 (Military Grid Reference System, MGRS) 点。

语法

[*strPoint*=]OBJECT.**MilitaryGridReferenceFromPoint** (*dblX, dblY*)

部分	说明
strPoint=	返回的值是转换点。是一个字符串。
OBJECT	表示地图对象。
dblX	双精度值，表示要转换的 X 坐标点。
dblY	双精度值，表示要转换的 Y 坐标点。

示例

```
Private Sub Map1_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    Dim MapX As Double
    Dim MapY As Double

    Map1.ConvertCoord X, Y, MapX, MapY, miScreenToMap
    Text2.Text = MapX
    Text3.Text = MapY

    Dim s As String
    s = Map1.MilitaryGridReferenceFromPoint(MapX, MapY)

    Dim b As Boolean

    Dim dx As Double
    Dim dy As Double
    b = Map1.MilitaryGridReferenceToPoint(s, dx, dy)

    Dim vx As Variant
```

```
    Dim vy As Variant
    b = Map1.MilitaryGridReferenceToPointV(s, vx, vy)
```

Map.MilitaryGridReferenceToPointV 方法

目的

此方法将军用网格参考系统 (MGRS) 点转换为地图的数字坐标系中的 XY 点。此方法返回布尔值，并以变量返回 x,y 值

语法

[bool=] OBJECT.**MilitaryGridReferenceToPointV** (strMGRS, vX, vY)

部分	说明
bool=	This return value is the converted point. 是一个字符串。
OBJECT	表示地图对象。
strMGRS	字符串值，表示要转换的 MGRS 坐标点。
vX	变量：表示转换的 X 坐标点。
vY	变量：表示转换的 Y 坐标点。

示例

```
Private Sub Map1_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    Dim MapX As Double
    Dim MapY As Double

    Map1.ConvertCoord X, Y, MapX, MapY, miScreenToMap
    Text2.Text = MapX
    Text3.Text = MapY

    Dim s As String
    s = Map1.MilitaryGridReferenceFromPoint(MapX, MapY)

    Dim b As Boolean

    Dim dx As Double
    Dim dy As Double
    b = Map1.MilitaryGridReferenceToPoint(s, dx, dy)

    Dim vx As Variant
    Dim vy As Variant
    b = Map1.MilitaryGridReferenceToPointV(s, vx, vy)
```

Map.MilitaryGridReferenceToPoint 方法

目的

此方法将军用网格参考系统 (MGRS) 点转换为地图的数字坐标系中的 XY 点。

语法

[bool=] OBJECT.**MilitaryGridReferenceToPointV** (strMGRS, dblX, dblY)

部分	说明
bool=	返回的值是转换点。是一个字符串。
OBJECT	表示地图对象。
strMGRS	字符串值，表示要转换的 MGRS 坐标点。
dblX	双精度值，表示转换的 X 坐标点。
dblY	双精度值，表示转换的 Y 坐标点。

示例

```
Private Sub Map1_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    Dim MapX As Double
    Dim MapY As Double

    Map1.ConvertCoord X, Y, MapX, MapY, miScreenToMap
    Text2.Text = MapX
    Text3.Text = MapY

    Dim s As String
    s = Map1.MilitaryGridReferenceFromPoint(MapX, MapY)

    Dim b As Boolean

    Dim dx As Double
    Dim dy As Double
    b = Map1.MilitaryGridReferenceToPoint(s, dx, dy)

    Dim vx As Variant
    Dim vy As Variant
    b = Map1.MilitaryGridReferenceToPointV(s, vx, vy)
```

Map.MouseIcon 属性

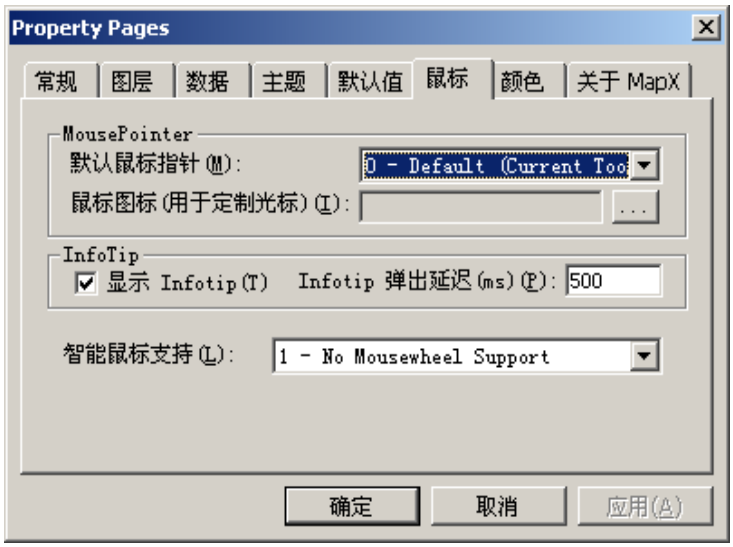
目的

如果需要指定用于 Map.MousePointer 属性鼠标图标的定制光标，可使用 Map.MouseIcon 属性。Map.MouseIcon 属性允许程序员使用字符串指定定制光标用作 MapX 中的默认鼠标光标，该字符串或者指定全路径 （包括文件名称），或者指定定制鼠标光标。该名称必须指定 .cur、 .ico 或 .ani 文件类型。如果无法加载指定光标，则将会报错。

注解

如果 Map.MousePointer 设置为 miCustomCursor 并且 Map.MouseIcon 尚未设置，则光标必须使用 miDefaultCursor 的形状绘制。 MousePointer 属性将仍然具有 miCustomCursor (40) 的值。在正确设置 Map.MouseIcon 之后，将会使用新的光标形状。

此外， MouseIcon 和 MousePointer 属性可以通过 MapX 属性对话框设置。为此只需单击地图，然后选择“鼠标”选项卡即可。



另请参阅

Map.MousePointer 属性

CursorConstants

Map.MousePointer 属性

目的

此属性设置鼠标光标的形状。取值为 **CursorConstants** 值。默认为 **miDefaultCursor**，表明将使用当前工件的光标。

注解

如果 Map.MousePointer 设置为 miCustomCursor (40) 并且 Map.MouseIcon 尚未设置，则光标必须使用 miDefaultCursor 的形状绘制。 MousePointer 属性仍然具有 miCustomCursor 的值。只要正确设置 Map.MouseIcon，即会使用新的光标形状。

Map.MouseWheelSupport 属性

目的

控制智能鼠标支持的水平：无、缩放 / 仅限滚动 （无 Autscroll）或全部。此属性使用 **MouseWheelSupportConstants** 设置。

Map.NumericCoordSys 属性

目的

返回一个读写 **CoordSys** 对象 （坐标系），控制 MapX 如何解释数字地图坐标。

注：如果一个地图包含一个或多个光栅图图层，则 MapX 自动在最可见的光栅图指定的投影中显示该地图。如果具有不同投影的其它光栅图成为最可见的光栅图，则坐标系统可以随着地图视图的更改 （由于缩放或平移）而进行更改。在此情况下，您无法更改地图的显示坐标系统。

注解

NumericCoordSys 表示用于处理数字地图坐标的坐标系。默认的坐标系为经度 / 纬度 WGS-84。通过修改地图的 NumericCoordSys （通过其 **CoordSys.Set** 方法），您可以使用另一坐标系中的地图坐标。

NumericCoordSys 属性不影响地图的外观。要设置显示地图的坐标系 （或 “投影”），可使用 **Map.DisplayCoordSys** 。

另请参阅

第 177 页的第 13 章：“使用坐标系统”

Map.Pan 方法

目的

此方法将按照采用屏幕坐标系 （像素）的给定偏移量来平移地图。

语法

OBJECT.**Pan** (ScreenX, ScreenY)

部分	说明
OBJECT	表示 Map 对象
ScreenX	该偏移量用于补充水平屏幕坐标。
ScreenY	该偏移量用于补充垂直屏幕坐标。

Map.PanAnimationLayer 属性

目的

此属性用于控制在使用平移工具时，动画图层 （如果存在）是否绘制到后台，以便图层内容可以在地图的其余部分上拖动。默认值为 FALSE。

Map.PaperUnit 属性

目的

此属性设置 MapX 将用于纸张地图的测量单位。通过使用 **PaperUnitsConstants 来指定测量单位。**默认值为 miPaperUnitInch。

Map.PreferCompactLegends 属性

目的

此属性设置是否显示压缩图例。取值为布尔值。默认值为 false。

注解

图例在创建时可以采用压缩或完全两种大小。创建主题地图之后，将会始终显示图例。此设置控制如何显示图例。图例始终可以使用 Theme.Legend.Compact 或 Theme.Legend.Visible 设置来更改，但是其位置由初始设置控制。

Map.PrintMap 方法

目的

将指定矩形中的地图打印到指定设备上下文。

语法

OBJECT.**PrintMap** (hDC x, y, w, h)

部分	说明
OBJECT	表示 Map 对象。
hDC	打印机设备上下文。Can be any device context.
x	采用 HIMETRIC 单位的左上角 X。
y	采用 HIMETRIC 单位的左上角 Y。
w	采用 HIMETRIC 单位的宽度。
h	采用 HIMETRIC 单位的高度。

注解

当前绘制的地图匹配给定矩形的大小。在维持宽高比时可以获取最佳结果。

如果选择了地图上的图元，可能不需要输出选择突出显示，请参阅 Map.ExportSelection 属性。

Map.PropertyPage 方法

目的

PropertyPage 方法允许用户在运行时控制或设置地图对象的各种属性。这是在设计模式中可用的相同属性页，尽管某些属性无法使用此方法在运行时设置。

注：此方法在 MapX Mobile 中不可用。

语法

OBJECT.PropertyPage

Map.RedrawInterval 属性

目的

此属性可用于增量绘图。这是一个以 1/100 秒为单位的时间，在显示使用非屏幕位图更新期间。最小值为 10。过高的值 （例如 3000）将有效关闭增量绘图 – 屏幕在地图完全充屏幕消失之后更新。

注解

当前，图例绘制到非屏幕位图，然后在位图完成之后传输到屏幕。目前，可以更加频繁地传输到屏幕，向用户显示确实发生的事情。（在 25 英里的缩放层次上这尤为实用。）

Map.Refresh 方法

目的

强制地图立即重新绘制。

语法

OBJECT.Refresh

注解

通常在采取操作之后，将会导致重新绘制地图。事实上， MapX 在接收到绘图消息之前不会重新绘制地图，以便可将多次绘制合而为一。此方法导致 MapX 立即重新绘制，无需等待下一绘制消息。此方法在重新绘制完成之前不会返回结果。

另请参阅

Map.AutoRedraw 属性，第 390 页

Map.ReuseEquivalentOnRestore 属性

目的

该属性是一个布尔值，默认值为 True。

在此属性设置为 False 并且 Datasets 集合或 Map 对象从 State 对象恢复之后， MapX 将废弃现有数据集，并在从 State 恢复时恢复新的数据集。要完全恢复每一数据集，还必须调用 Datasets.Restore 来提供正在恢复的每一数据集外部资源 （例如 ADO 记录集）。

在此属性值设置为 Ture 并且 Datasets 集合或 Map 对象从 State 对象恢复之后， MapX 将尝试复用等同于正在恢复的数据集的现有数据集。您必须调用 Datasets.Item 来确认是否找到了等同的数据集。如果没有发现等同的数据集，则必须调用 Datasets.Restore 来完全恢复数据集。

示例

以下示例保存并恢复带有名为 Dataset1 数据集的 Map 对象：

```
' 保存地图：
objState.Save objMap
' 执行其他操作
...
' 如果等同，则在恢复状态时允许复用现有图层和 / 或数据集
objMap.ReuseEquivalentOnRestore = True
' 恢复地图对象：
objState.Restore objMap
' 如果数据集仍未出现：
if not objMap.Datasets.Contains("Dataset1") then
    ' 设置数据集来源 （例如 ADO 记录集）：
    ...
```

```
        ' 完成数据集恢复：
        objMap.Datasets.Restore "Dataset1", source
    end if
```

注解

此属性在 MapXtreme 应用程序对每一会话请求加速恢复 Map 对象状态时非常实用。恢复数据集操作占用资源较多，其原因在于 MapX 必须绑定并从外部来源读取所有数据 （例如 ADO）。允许 MapX 复用匹配正在恢复的数据集的数据集，即无需绑定并从外部数据集读取。

设置此属性的值不能确保 MapX 可以发现匹配的数据集。例如，如果具有在会话之间共享 MapX 对象的 ASP 应用程序，并且应用程序允许每个会话添加或删除数据集，当每个会话恢复其 Map 状态时，地图可能具有或不具有数据集。如果 MapX 没有发现匹配的数据集，则必须调用 Datasets.Restore 来完全恢复数据集。您可以发现 MapX 是否通过调用 Datasets.Contains 发现了匹配的数据集。

切记如果外部来源更改频繁 （例如在会话活动时），您可能不希望复用现有数据集。如果数据没有频繁更新，则您可以在第一个会话请求时开始新的数据集，然后允许 MapX 来复用现有数据集，如果在后续会话请求期间出现的话。

另请参阅

State 对象，第 456 页

Datasets.Contains 方法，第 265 页

Map.Rotation 属性

目的

此属性设置地图的旋转。旋转值以度为单位指定，顺时针增长， 0 表示地图不旋转。向地图应用旋转不会更改其缩放级别或中点。

注：包含一个或多个的光栅图层不能旋转，其原因在于 MapX 无法旋转光栅图像。

Map.SaveMapAsGeoset 方法

目的

此方法创建 geoset 文件，表示地图的当前状态 （图层信息，标记设置等）。

语法

OBJECT.**SaveMapAsGeoset** (name, filespec)

部分	说明
OBJECT	一个 Map 对象。
name	表示 geoset 友好名称的字符串 （出现在 Geodictionary 中的描述）。如果指定空字符串，则将会使用地图的标题。

部分	说明
filespec	表示用于 geoset 文件的完全文件路径的字符串。如果没有指定路径，则新文件将置于数据目录中。如果没有指定扩展名，则将使用标准的 .gst 扩展名。如果指定文件已存在，则将会被覆盖。

注解

如果地图包含任意临时图层、用户绘制图层，或通过 miLayerInfoTypeServer 添加的远程图层，则相应图层将置于新的 geoset 文件之外，并且将会保存 （在保存 geoset 之后）。通过 .tab 文件添加的远程图层将保存在结果 geoset 之内。如果地图中没有图层，则此方法还将报错。此外，形状文件 (*.shp) 不得保存到 geoset。

Map.SearchPath 属性

目的

此属性允许用户动态设置搜索路径。搜索路径本质上是信息所在的驱动器和目录的 “电子绘图过程”。在程序员设置此属性之前，搜索路径最初将从注册表读取。 SearchPath 属性只在运行时可用。

Map.SelectionStyle 属性

目的

此属性返回或设置 **Style** 对象，表示绘制所选图元的样式。

注解

此属性返回一个读写 Style 对象，控制所有图层中的所选图元的外观。

要控制所选区域的外观，设置此 Style 对象的 RegionXXX 属性，例如 RegionColor。 RegionColor 和 RegionBorderXXX 属性用于在所选符号调整对象周围绘制图框。 LineXXX 属性用于绘制所选的线图元。注意 SymbolXXX 和 TextXXX 属性不应用并且无效。

注：其中的 XXX 是用于 **Style 对象**的各种属性占位符，定义区域、线等对象。

Remarks

要从输出忽略选择突出显示，可将 **Map.ExportSelection** 属性设置为 False。

Map.SetSize 方法

目的

此属性设置控件的大小，单位为像素。

语法

OBJECT.**SetSize** (W, H)

部分	说明
OBJECT	表示 Map 对象。
W	预期的控件宽度，单位为像素。
H	预期的控件高度，单位为像素。

注解

在大部分 Visual Basic 应用程序中，您可以重新调整控件大小，不使用 SetSize 方法 － 只使用 Visual Basic 支持的标准 Width 和 Height 属性。SetSize 方法无法和表单上的地图对象一起使用。SetSize 用于通过 CreateObject() 调用创建地图对象的用户。

Map.SnapTolerance 属性

目的

此属性取 1 和 50 之间的整数。使用 SnapTolerance 来指定匹配节点操作的允差 （例如为工具匹配到节点，该节点距离鼠标光标的必要像素距离）。默认值为 5 个节点。

注解

如果使用的地图窗口中包含众多位置接近的对象，则可能需要相当紧凑的匹配容差 （1、 2 或 3 个像素）。这将允许您在非常靠近节点的位置上绘图，同时无需匹配它。在对象的位置较为松散时，您可以使用宽松的容差。这将允许您在距离节点相当远的地方还可以匹配到节点。

另请参阅

节点选择和编辑，第 209 页

Map.SnapToNodeSupport 属性，第 417 页

Map.SnapToNodeSupport 属性

目的

当此布尔值属性为 True 时，将启用 SnapToNode 功能。匹配到该节点将启用一个十字型的光标，在鼠标光标接近可选图层中的节点时出现。您使用的工具将匹配的该节点，允许您在该节点上更加轻松地使用当前工具 （例如使用选择工具选择节点）。您可以通过设置 Map.SnapTolerance 属性，确定在十字型光标出现之前，光标距离节点的距离。

注解

您可以通过按下 s 或 S 键来演示。（此字符在 MapX 资源中定义，因此可在外语平台上更改）。

此功能和任意 MapX 普通工具以及任意由应用程序创建的定制工具一起协同工作。此功能在通过使用选择工具选择和拖放节点来进行编辑操作时尤为实用。

另请参阅

Map.SnapTolerance 属性，第 417 页

节点选择和编辑，第 209 页

Map.Title 属性

目的

Title 对象指定地图的标题。如果没有标题可将其设置为 NULL。

使用 Title 对象可以控制标题文本、显示样式和位置。提示：如果要设置标题文本，并且您不考虑其样式或位置，则可以在设计阶段设置 Map.TitleText 属性。

Map.TitleText 属性

目的

此属性返回或设置作为地图标题的字符串。如果您指定空字符串，则将隐藏标题。

注解

该标题默认为 GeoSet 所用的名称。如果 TitleText 等同于 GeoSet 名称，并且 Geoset 已更改，则 TitleText 也将更改。

另请参阅

Map.Title 属性，第 418 页

Map.Version 属性

目的

此属性指定您正在使用的 MapX 版本。这是一个字符串值，是只读的。

Map.WaitCursorEnabled 属性

目的

此属性允许您手动启用或禁用 Wait Cursor。默认值为 True。

Map.Zoom 属性

目的

包含或设置地图的当前缩放值。缩放值定义为地图的宽度。这是一个双精度值，采用由 MapUnit 属性指定的单位。

Map.ZoomTo 方法

目的

缩放并对中地图。

语法

OBJECT.**ZoomTo** (Zoom, x, y)

部分	说明
OBJECT	表示 Map 对象。
缩放	缩放值。 MapX 将缩放值定义为控件的宽度。取双精度值，以地图单位 (MapUnit) 指定。
X	将地图对中至的 X 坐标。双精度值，表示经度。
Y	将地图对中至的 Y 坐标。双精度值，表示纬度。

注解

将此方法用作缩放和重新对中地图的简单途径。另外一种方法时设置 Map 对象的 Zoom、 CenterX 和 CenterY 属性，但是这样做将导致 3 个屏幕重新绘制，除非禁用屏幕更新，然后在使用 Map.AutoRedraw 属性时启用。

MultivarCategory 对象和 MultivarCategories 集合

饼图或柱状图主题地图通过 MultivarCategories 集合展现给 OLE， MultivarCategories 集合是 MultivarCategory 对象的集合。柱状图中的每个柱体在该集合中均有一个对象，或者饼图中每个楔块一个对象。

要获取 **MultivarCategories** 集合，应引用 ThemeProperties.MultivarCategories 属性。

对象属性

- **MultivarCategory.Style 属性**

集合属性

- **MultivarCategories.Count 属性**
- **MultivarCategories.Item 属性**

另请参阅

ThemeProperties.MultivarCategories 属性，第 491 页

LegendText 对象和 LegendTexts 收集，第 386 页

MultivarCategories.Count 属性

目的

这是一个只读整数值，表示由主题使用的变量数、主题条形图中的条数或主题饼图中的楔块数。

MultivarCategories.Item 属性

目的

此属性从集合返回一个 MultivarCategory 对象。此对象说明一个主题变量的设置 （例如条形图主题中的一个条）。

MultivarCategory.Style 属性

目的

此属性控制位于此 MultivarCategory 中的对象的样式。这是一个 Style 对象，并且可以设置为现有样式对象，或者您可以分别设置样式属性。默认值设置用于提供高对比样式。

另请参阅

Style 对象，第 463 页

NotesQueryInfo 对象

通过创建 NotesQueryInfo 对象和使用 Dataset 类型 8 (miDatasetNotesQuery) 针对 Notes Dataset 执行查询。这一 Dataset 类型允许 MapX 用户请求 Notes 在运行时针对数据库执行查询。要向 MapX 添加此类 Datasets，需要创建 NotesQueryInfo 对象，并将其传递到 Datasets.Add。

注：此对象及其属性在 MAPX MOBILE 不可用，其原因在于 MAPX MOBILE 不支持远程连接。

对象属性

- **NotesQueryInfo.BeginDate 属性**
- **NotesQueryInfo.Database 属性**
- **NotesQueryInfo.DefaultNumericValue 属性**
- **NotesQueryInfo.DefaultStringValue 属性**
- **NotesQueryInfo.EndDate 属性**
- **NotesQueryInfo.FullTextSearch 属性**
- **NotesQueryInfo.MaxNumDocs 属性**
- **NotesQueryInfo.Query 属性**
- **NotesQueryInfo.Server 属性**

NotesQueryInfo.BeginDate 属性 （NotesQueryInfo 对象）

目的

定义开始截止日期 / 时间的字符串或表示其的字符串 （可选）。请注意在此日期 / 时间之前修改的 Notes 在搜索期间将被忽略。

NotesQueryInfo.Database 属性 （NotesQueryInfo 对象）

目的

是服务器上 Notes 数据库的名称，或指向本地 Notes 数据库 (.nsf) 的路径。

NotesQueryInfo.DefaultNumericValue 属性 （NotesQueryInfo 对象）

目的

由于 Notes Query 可以返回不包含所有指定字段的文档，因此如果请求的数字字段未找到 （可选），则 MapX 将返回默认的数字 （双精度）。默认值为 0。

另请参阅

Datasets.Add 方法，第 263 页

NotesQueryInfo.DefaultStringValue 属性 （NotesQueryInfo 对象）

目的

由于 Notes Query 可以返回不包含所有指定字段的文档，因此如果请求的字符串字段未找到 （可选），则 MapX 将返回默认的字符串。默认值为 ""。

NotesQueryInfo.EndDate 属性 （NotesQueryInfo 对象）

目的

定义结束截止日期 / 时间的字符串或表示其的字符串 （可选）。请注意在此日期 / 时间之前的所做修改在搜索期间将被忽略。

NotesQueryInfo.FullTextSearch 属性 （NotesQueryInfo 对象）

目的

布尔值 （TRUE 或 FALSE），表示数据库将具有全文索引并且 Query 属性为全文搜索查询。

另请参阅

NotesQueryInfo.Query 方法

NotesQueryInfo.MaxNumDocs 属性 （NotesQueryInfo 对象）

目的

指定结果中允许的 Notes 数量上限的整数 （可选）。如果输入为 0，则将收集所有满足搜索标准的注释。

NotesQueryInfo.Query 属性 （NotesQueryInfo 对象）

目的

如果 FullTextSearch 属性设置为 TRUE，则此属性为全文搜索查询字符串。如果 FullTextSearch 属性为 TRUE，则此属性将成为 Notes 搜索规则。请参阅 Notes 数据库，查阅有关如何构成全文搜索查询或规则字符串的有关信息。

另请参阅

NotesQueryInfo.FullTextSearch 方法

NotesQueryInfo.Server 属性 （NotesQueryInfo 对象）

目的

定义运行上述数据库的 Notes 服务器的字符串。如果 Database 属性引用本地数据库，则此字符串留为空。

另请参阅

NotesQueryInfo.Database 方法

NotesViewInfo 对象

NotesViewInfo 对象是用于 miDatasetNotesView 类型 Datasets 的 Datasets.Add 的第二个参数所必需的。Dataset 类型将访问指定的 Dataset 视图。

对象属性

- **NotesViewInfo.Database 属性**
- **NotesViewInfo.Server**
- **NotesViewInfo.View**

另请参阅

Datasets.Add 方法，第 263 页

NotesViewInfo.Database 属性 （NotesViewInfo 对象）

目的

此读写属性是服务器上 Notes 数据库的名称，或指向本地 Notes 数据库 (.nsf) 的路径。

NotesViewInfo.Server （NotesViewInfo 对象）

目的

此读写属性是定义运行上述数据库的 Notes 服务器的字符串。如果 Database 属性引用本地数据库，则此字符串留为空。

NotesViewInfo.View （NotesViewInfo 对象）

目的

此读写属性是引用将要从中提取数据的数据库中的特定视图。

OCIQueryInfo 对象

如果 Dataset type 是 **miDatasetOCI**，则 OCIQueryInfo 对象传递到 Datasets.Add 方法中作为源参数。此对象包含有关如何连接到 OCI 数据源和将哪些记录作为记录集处理的有关信息。

在使用 OCIQueryInfo 对象时，必须确保安装正确的 OCI dll。在 Visual Basic 中，选择 Project > References。单击 Browse 按钮以向适当的 dll 添加引用。这些 dll 安装在 MapX 程序目录中。

注：此对象及其属性在 MAPX MOBILE 不可用，其原因在于 MAPX MOBILE 不支持远程连接。

对象属性

- **OCIQueryInfo.ConnectString 属性**
- **OCIQueryInfo.Query 属性**

另请参阅

Datasets.Add 方法，第 263 页

OCIQueryInfo.ConnectString 属性 （OCIQueryInfo 对象）

目的

此属性设置字符串值，指定要连接到的数据库、连接的用户身份和必要的有效口令。

示例

```
"SRVR=rock;UID=mapx;PWD=mapx"
```

OCIQueryInfo.Query 属性 （OCIQueryInfo 对象）

目的

此属性设置指定有效 SQL Select 语句的字符串值。

ODBCQueryInfo 对象

如果 Dataset type 是 **miDatasetODBC**，则 ODBCQueryInfo 对象传递到 Datasets.Add 方法中作为源参数。此对象包含有关如何连接到 ODBC 数据源和将哪些记录作为记录集处理的有关信息。

注：如果使用 ODBCQueryInfo 对象，则项目必须引用注册 MODBCDataset.DLL （MapInfo ODBC Dataset 引擎库）。

在 Visual Basic 中，选择 Tools > References （用于 VB 4）或 Project > References （用于 VB 4）。单击 Browse 按钮以向适当的 dll 添加引用。这些 dll 安装在 MapX 程序目录中。

注：此对象及其属性在 MAPX MOBILE 不可用，其原因在于 MAPX MOBILE 不支持远程连接。

对象属性

- **ODBCQueryInfo.ConnectString 属性**
- **ODBCQueryInfo.DataSource 属性**
- **ODBCQueryInfo.SqlQuery 属性**

另请参阅

Datasets.Add 方法，第 263 页

ODBCQueryInfo.ConnectString 属性 （ODBCQueryInfo 对象）

目的

此属性设置连接字符串，以便与 ODBC 数据源连接。例如，指定 “ODBC;”。还可以加入 “uid=”，“pwd=”或 DLG=。如果缺少必要的信息，数据源的 ODBC 驱动程序将会提示用户。

“DLG=”控制连接对话框的显示：DLG=0 表示从不打开对话框， DLG=1 表示总是打开一个对话框，DLG=2 显示连接对话框，但是仅在需要时显示 （即如果未提供所有必要的信息时）。

ODBCQueryInfo.DataSource 属性 （ODBCQueryInfo 对象）

目的

此属性设置 ODBC 数据源的名称。如果属性留为空白，则 ODBC 将提示用户。ODBC DataSources 使用 ODBC 管理器设置，后者可见于 Windows “控制面板”。

ODBCQueryInfo.SqlQuery 属性 （ODBCQueryInfo 对象）

目的

此属性指定用于检索构建 Dataset 的行和列的 SQL 查询字符串。

例如，指定 "select * from USA"。

Parts 集合

Parts 集合是一组 Points 集合对象即 Parts 集合的每个对象均由一个 Points 集合构成。Parts 集合具有正常的 Count、 Item 和 Remove 方法。 Add 方法取 Source Points 对象，返回添加到 Parts 集合的 Points 对象（例如，新的 Points 对象添加到 Parts 集合，点从 Source Points 集合复制）。

集合属性

- **Parts.Count 属性**
- **Parts.Item 属性**

集合方法

- **Parts.Add 方法**
- **Parts.Remove 方法**
- **Parts.RemoveAll 方法**

Parts.Add 方法

目的

Add 方法取 Source Points 对象，返回添加到 Parts 集合的 Points 对象 （例如，新的 Points 对象添加到 Parts 集合，点从 Source Points 集合复制）。

语法

OBJECT.**Add**(Points)

Points 是您要添加的 Points 集合对象

Parts.Count 属性

目的

此属性统计集合中的项数。

Parts.Item 属性

目的

此属性从 Parts 集合返回 Points 集合对象。这是一个变量，您必须指定基于 1 的索引编号。这是 Parts 集合的默认属性。

Parts.Remove 方法

目的

此方法从这一集合去除 Points 集合对象。

语法

OBJECT**.Remove** (index)

Parts.RemoveAll 方法

目的

此方法从集合去除所有 Parts 对象。

语法

OBJECT**.RemoveAll**

Point 对象和 Points 集合

Point 对象表示 XY 坐标对。当 Point 属于 Feature 时， X 坐标表示经度， Y 坐标表示纬度。

对象属性

- **Point.X 属性**
- **Point.Y 属性**

对象方法

- **Point.Offset 方法**
- **Point.Set 方法**

集合属性

- **Points.Count 属性**
- **Points.Item 属性**

集合方法

- **Points.Add 方法**
- **Points.AddXY 方法**
- **Points.GetXY 方法**
- **Points.GetXYV 方法**
- **Points.X 方法**
- **Points.Y 方法**
- **Points.Remove 方法**
- **Points.RemoveAll 方法**

Point.Offset 方法

目的

此方法通过更改点的坐标来更改点的位置。

语法

OBJECT.**Offset**(double deltaX, double deltaY)

部分	说明
OBJECT	表示 Point 对象。
deltaX	用于添加到 X 坐标的量。
deltaY	用于添加到 Y 坐标的量。
double	DeltaX 和 deltaY 是 Map 的数字坐标系中的值。

另请参阅

Map.NumericCoordSys 属性，第 411 页

Point.Set 方法

目的

此方法设置点的坐标。

语法

OBJECT.**Set**(double X, double Y)

部分	说明
OBJECT	表示 Point 对象。
X	X 是点的 X 坐标。
Y	Y 是点的 Y 坐标。

Point.X 属性

目的

是点的 X 坐标。 X 和 Y 是 Map 的数字坐标系中的值。

另请参阅

Map.NumericCoordSys 属性，第 411 页

Point.Y 属性

目的

是点的 Y 坐标。 X 和 Y 是 Map 的数字坐标系中的值。

另请参阅

Map.NumericCoordSys 属性，第 411 页

Points.Add 方法

目的

此方法向 Points 集合添加点。

语法

[Point=]OBJECT.**Add**(Point, [Position])

部分	说明
OBJECT	表示 Points 对象。
Point	要添加的 Point 对象。
Position	介于 1 和 Points.Count 之间的整数，指定点添加到集合中的位置。如未指定或无效，则点应该追加到集合的末尾。

Points.AddXY 方法 目的

此方法使用给定的 X 和 Y 值创建新点，并将其添加到 Points 集合。

语法

[Point=]OBJECT**.AddXY** (x, y, [Position])

部分	说明
OBJECT	表示 Points 对象。
X	双精度数字 CoordSys。
Y	双精度数字 CoordSys。
Position	介于 1 和 Points.Count 之间的整数，指定点添加到集合中的位置。如未指定或无效，则点应该追加到集合的末尾。

Points.Count 属性

目的

集合中的项数。

Points.Item 属性

目的

此属性从 Parts 集合返回 Points 集合对象。这是一个变量，您必须指定基于 1 的索引编号。这是 Parts 集合的默认属性。

Points.GetXY 方法

目的

此方法返回 Points 集合中指定点的 XY 坐标。

注：此方法在 MapX Mobile 中不可用。要在 MapX Mobile 中获取 XY 坐标，必须使用 GetXYV 方法。

语法

OBJECT.**GetXY**(*lIndex, dX, dY*)

部分	说明
OBJECT	Points 集合。
lIndex	长精度：此引用值是点对象的索引。
dX	双精度：这一引用值是 X 坐标。
dY	双精度：这一引用值是 Y 坐标。

Points.GetXYV 方法

目的

此方法返回作为变量的 Points 集合中指定点的 XY 坐标。

语法

OBJECT.**GetY**(*lIndex*, *vX*, *vY*)

部分	说明
OBJECT	这是一个 Points 对象。
lIndex	长精度：此引用值是点对象的索引。
vX	变量：这一引用值是 X 坐标。
vY	变量：这一引用值是 Y 坐标。

Points.X 方法

目的

此方法返回 Points 集合中指定点的 X 坐标。

语法

OBJECT.**X**(*lIndex*)

部分	说明
OBJECT	这是一个 Points 对象。
lIndex	长精度：此引用值是点对象的索引。

Points.Y 方法

目的

此方法返回 Points 集合中指定点的 Y 坐标。

语法

OBJECT.**Y**(*lIndex*)

部分	说明
OBJECT	这是一个 Points 对象。
lIndex	长精度：此引用值是点对象的索引。

Points.Remove 方法

目的

此方法从集合去除 Point 对象。

语法

OBJECT.**Remove** (index)

Points.RemoveAll 方法

目的

此方法从集合去除所有 Point 对象。

语法

OBJECT.**RemoveAll**

RangeCategory 对象和 RangeCategories 集合

范围主题地图的设置通过 RangeCategories 集合显示，这是 RangeCategory 对象的集合，每个范围一个对象，按升序排序。集合中的每个对象说明一个范围 （显示样式、最小值和最大值等）。

对象属性

- **RangeCategory.Max 属性**
- **RangeCategory.Min 属性**
- **RangeCategory.NumItems 属性**
- **RangeCategory.Style 属性**

集合属性

- **RangeCategories.AllOthersCategory 属性**
- **RangeCategories.Count 属性**
- **RangeCategories.Item 属性**

RangeCategories.AllOthersCategory 属性

目的

此属性从集合返回一个 RangeCategory 对象，此对象说明不在范围主题中的所有范围的类别。返回的 RangeCategory 对象带有 AllOthersCategory 属性，具有未定义的 Min 和 Max 属性，其默认值为 0。但是，您可以设置其 Style 属性，如下例所示。在用于 AllOthersCategory 范围类别对象出现的文本可使用 **LegendTexts.AllOthersText** 属性设置。

RangeCategories.Count 属性

目的

这是一个只读整数值，表示范围主题中的范围数量。

RangeCategories.Item 属性

目的

此属性从集合返回一个 RangeCategory 对象，此对象说明范围主题中的所有范围。

RangeCategory.Max 属性

目的

包含或设置范围主题中的范围的最大值。这是双精度值。初始值在主题创建时设置 (**Themes.Add**)，取决于分发方法 (**ThemeProperties.DistMethod**)：

- 如果 ThemeProperties.DistMethod 是 miEqualCountPerRange、 miEqualRangeSize、miNaturalBreak 或 miStandardDeviation，则将在每次重新计算 Theme 对象时设置新值。
- 如果 ThemeProperties.DistMethod 为 miCustomRanges， MapX 将假定您自己设置该值，并将在对数据值分组时使用您定义的范围。如果重新计算主题时范围出现重叠，则会出错。

另请参阅

ThemeProperties.DistMethod 属性，第 489 页

RangeCategory.Min 属性

目的

包含或设置范围主题中的范围的最小值。这是双精度值。初始值在主题创建时设置 (Themes.Add)，取决于分发方法 (ThemeProperties.DistMethod)：

- 如果 ThemeProperties.DistMethod 是 miEqualCountPerRange、 miEqualRangeSize、miNaturalBreak 或 miStandardDeviation，则将在每次重新计算 Theme 对象时设置新值。
- 如果 ThemeProperties.DistMethod 为 miCustomRanges， MapX 将假定您自己设置该值，并将在对数据值分组时使用您定义的范围。如果重新计算主题时范围出现重叠，则会出错。

另请参阅

ThemeProperties.DistMethod 属性，第 489 页

Themes.Add 方法，第 485 页

Theme.AutoRecompute 属性，第 481 页

RangeCategory.NumItems 属性

目的

NumItems 是在由此范围类别的 Min 和 Max 属性指定的范围中的图元数量。这是一个整数值，是只读属性。

注：此属性只在 MapX 生成该主题时有效 （Theme.ComputeTheme = TRUE）。如果您手动计算主题 （Theme.ComputeTheme = FALSE），则 NumItems 将不会反应类别中的项数，并且不定位为结果。

RangeCategory.Style 属性

目的

包含用于给定 RangeCategory 对象的 Min 和 Max 属性范围中的所有对象的样式。可以设置为现有样式对象，也可以分别设置 Style 属性。

注：如果您正在使用主题范围，并且分别指定范围颜色 （例如每个范围），则 ThemeProperties.SpreadBy 必须设置为 miSpreadByNone。

Rectangle 对象

Rectangle 对象用于指定对象的范围。有若干属性，例如 Map.Bounds、 Feature.Bounds 和 Layer.Bounds 均为矩形对象。矩形对象由其最小和最大 X 坐标以及其最小和最大 Y 坐标定义。对于地理属性 （例如 Map.Bounds），这些坐标将就 Map.NumericCoordSys 指定。对于和 Windows 相关的方法 （例如 Style.DrawLineSample）， Rectangle 对象的属性将以像素值指定。

对象属性

- **Rectangle.Height 属性**
- **Rectangle.Width 属性**
- **Rectangle.XMax 属性**
- **Rectangle.XMin 属性**
- **Rectangle.YMax 属性**
- **Rectangle.YMin 属性**

对象方法

- **Rectangle.Offset 方法**
- **Rectangle.Set 方法**

Rectangle.Height 属性

目的

矩形的高度。高度定义为 Rectangle.YMax - Rectangle.Ymin。这是一个只读双精度属性。要更改矩形的高度，可使用 Rectangle.Set 方法。

Rectangle.Offset 方法

目的

通过更改其坐标来更改矩形的位置。此方法通过 X 方向的 deltaX 和 Y 方向的 deltaY 移动矩形。

语法

Object.**Offset**(deltaX, deltaY)

部分	说明
OBJECT	表示 Rectangle 对象。
deltaX	双精度值，表示矩形在 X 方向的移动量。
deltaY	双精度值，表示矩形在 Y 方向的移动量。

Rectangle.Set 方法

目的

此属性设置矩形的坐标。 Rectangle.Set 将对参数分类，确保 Rectangle.XMin <= Rectangle.XMax 和 Rectangle.YMin <= Rectangle.Ymax。如果 x1 > x2 或 y1 > y2，则可以传递参数。

语法

OBJECT.**Set** (x1, y1, x2, y2)

部分	说明
OBJECT	表示 Rectangle 对象。
x1, y1	双精度值，表示矩形一角的坐标。
x2, y2	双精度值，表示矩形对角的坐标。

Rectangle.Width 属性

目的

矩形的宽度。宽度定义为 Rectangle.XMax - Rectangle.XMin。这是一个只读双精度属性。要更改矩形的宽度，可使用 Rectangle.Set 方法。

Rectangle.XMax 属性

目的

矩形左边的坐标。这是一个只读双精度属性。要更改矩形的位置，可使用 Rectangle.Set 或 Rectangle.Offset 方法。

Rectangle.XMin 属性

目的

矩形右边的坐标。这是一个只读双精度属性。要更改矩形的位置，可使用 Rectangle.Set 或 Rectangle.Offset 方法。

Rectangle.YMax 属性

目的

矩形顶边的坐标。这是一个只读双精度属性。要更改矩形的位置，可使用 Rectangle.Set 或 Rectangle.Offset 方法。

Rectangle.YMin 属性

目的

矩形底边的坐标。这是一个只读双精度属性。要更改矩形的位置，可使用 Rectangle.Set 或 Rectangle.Offset 方法。

ResolveObject 对象和 ResolveObjects 集合

ResolveObjects 集合作为 ResolveDataBindEx 事件的参数传递。它是一个备选图层的集合，可在使用自动数据绑定时绑定。

对象属性

- **ResolveObject.SourceMatch 属性**
- **ResolveObject.TableMatch 属性**
- **ResolveObject.TableName 属性**

集合属性

- **ResolveObjects.Count 属性**
- **ResolveObjects.Item 属性**

ResolveObject.TableName 属性

目的

此属性用于设置要分析的表名。

ResolveObject.SourceMatch 属性

目的

这是匹配表的源行的百分比。示例：如果源数据为 5 行，其中有 4 个匹配，则此值为 80。

ResolveObject.TableMatch 属性

目的

地图对象匹配源数据的百分比。示例：如果源数据为 5 行，其中有 4 个匹配，而地图有 50 个对象，则此值为 8。

ResolveObjects.Count 属性

目的

集合中的项数。

ResolveObjects.Item 属性

目的

此属性从集合返回 ResolveObject 对象。这是一个变量，您可以指定名称或基于 1 的索引编号。这是 ResolveObjects 集合的默认属性。

RowValue 对象和 RowValues 集合

RowValue 对象表示数据集中字段的单值。 RowValue 对象的集合由 Dataset.RowValues 返回。

RowValues 集合是一组可创建 RowValue 对象 RowValues 集合可以传递到 Layer.UpdateFeature、Layer.AddFeature 和 Feature.Update，以指定用于数据集中的字段的数据值，该数据集绑定到要更新的图层。

对象属性

- **RowValue.Dataset 属性**
- **RowValue.Field 属性**
- **RowValue.ReadOnly 属性**
- **RowValue.Value 属性**

集合属性

- **RowValues.Count 属性**
- **RowValues.Item 属性**
- **RowValues.ReadOnly 属性**

集合方法

- **RowValues.Add 方法**
- **Selection.Clone 方法**
- **RowValues.Remove 方法**
- **RowValues.RemoveAll 方法**

另请参阅

Layer.UpdateFeature 方法，第 363 页

Layer.AddFeature 方法，第 342 页

Feature.Update 方法，第 280 页

Dataset.RowValues 属性，第 260 页

RowValue.ReadOnly 属性

目的

此属性确定是否可以针对对象设置属性。

RowValue.Dataset 属性

目的

此属性用于指定值用于哪一 Dataset。

RowValue.Field 属性

目的

此属性用于指定值用于 Dataset 中的哪一字段。

RowValue.Value 属性

目的

此属性用于值指定用于字段的值。

RowValues.Count 属性

目的

此只读属性指定 RowValues 集合中的行数。

RowValues.ReadOnly 属性

目的

此只读属性指定是否可以针对集合设置属性。对于 Datasets.RowValues 集合，该属性为 True ；对于独立创建的 RowValue 对象，则为 False。

RowValues.Item 属性 （

目的

此属性从 RowValues 集合返回 RowValue 对象。

语法

OBJECT.**Item** (index)

部分	说明
OBJECT	表示 RowValues 集合
index	**变量：**此属性指定返回集合中的哪一项。可以为字段名或数字索引。

RowValues.Remove 方法

目的

此方法从集合去除指定的 RowValue 对象。

注：虽然从该集合中去除，但不影响 Dataset。

语法

OBJECT.**Remove** (index)

部分	说明
OBJECT	表示 RowValues 集合
index	**Variant：**此属性指定去除集合中的哪一项。可以为字段名或数字索引。

RowValues.Add 方法

目的

此属性向指定的 RowValue 集合添加 RowValue 对象。

语法

[RowValues=] OBJECT.**Add** (RowValue)

部分	说明
OBJECT	表示 RowValues 集合
RowValue	此方法将数字字段添加到 Fields 集合。

RowValues.RemoveAll 方法

目的

此方法从 RowValues 集合去除所有 RowValue 对象。

注：虽然从该集合中去除，但不影响 Dataset。

语法

OBJECT.**RemoveAll**

部分	说明
OBJECT	表示 RowValues 集合

RowValues.Clone 方法

目的

此方法从 Dataset 返回指定 RowValues 集合的复本。字段可以从复制的 RowValues 集合添加或删除，以进行较小的更新，例如只更新特定字段。直接从数据集获取 RowValues 集合 （不复制）将不允许您添加或删除字段。

注：miDatasetLayer 是唯一可以更新的数据集类型。

语法

[RowValues=]OBJECT.**Clone**

部分	说明
OBJECT	表示 RowValues 集合。

另请参阅

Feature.Update 方法，第 280 页

Selection 集合

MapX 基础函数是选择对象或记录，以便您可以对其执行其他任务，例如获取更多信息。 Selection 是 Feature 对象的集合。

Layer.Selection 集合是特殊的 Features 集合，最终用户可以在其中使用任意选择工具来添加或删除对象。您可以通过使用 Clone 方法来获取图层选择的快照，以返回 Features 对象。

MapX 为您提供了众多用于创建选择的命令和工具，例如选择工具、半径选择、选取框选择和边界选择工具。

要使用工具选择记录，可单击或框起关联的图形对象。

使用 Selection 集合中的属性和方法以及 Feature 对象，获取附加信息或控制集合的图元。

对象属性

- **Selection.Count 属性**
- **Selection.Item 属性**

对象方法

- **Selection.Add 方法**
- **Selection.ClearSelection 方法**
- **Selection.Clone 方法**
- **Selection.Common 方法**
- **Selection.Remove 方法**
- **Selection.Replace 方法**
- **Selection.SelectAll 方法**
- **Selection.SelectByID 方法**
- **Selection.SelectByPoint 方法**
- **Selection.SelectByRadius 方法**
- **Selection.SelectByRectangle 方法**
- **Selection.SelectByRegion 方法**

另请参阅

Layer 对象和 Layers 集合，第 340 页

Feature 对象和 Features 集合，第 271 页

Selection.Count 属性

目的

集合中的项数。

Selection.Item 属性

目的

从选择集合返回 Feature 对象 － 当前只支持数字索引。这是 Selection 集合的默认属性。

Selection.Add 方法

目的

此方法将 Feature 对象、 FindFeature 对象或 Features 集合中的所有图元添加到 Selection 集合 （UNION 集运算）。结果为此 Selection 集合在调用之前包含所有图元，以及对象中传入的所有图元。传入的对象将保持不变。

语法

OBJECT.**Add** (Source)

部分	说明
OBJECT	表示 Selection 集合。
Source	要添加的 Feature、 Features 或 FindFeature 对象。

另请参阅

Feature 对象和 Features 集合，第 271 页

FindFeature 对象，第 315 页

Selection.ClearSelection 方法

目的

此方法取消选择此图层中的所有图元。使用 Layers.ClearSelection 来从所有特此清空选择。

语法

OBJECT.**ClearSelection**

部分	说明
OBJECT	表示 Selection 集合。

Selection.Clone 方法

目的

此方法将集合复制为另一个 Features 集合对象。

语法

OBJECT.**Clone**

部分	说明
OBJECT	表示 Selection 集合。

Selection.Common 方法

目的

此方法将这一集合与一个 Feature 对象、 FindFeature 对象或 Features 集合合并，以便此 Selection 集合只包含两者中均存在的图元。（INTERSECT 集运算）。这将会从 Selection 集合中去除没有位于传入对象中的各个图元。传入对象将保持不变。

语法

OBJECT.**Common** (Source)

部分	说明
OBJECT	表示 Selection 集合。
Source	这是您将与 OBJECT 合并的 **Feature 对象**、**FindFeature 对象或 Features 集合**。

Selection.Remove 方法

目的

此方法将移除此集合中的 Feature 对象、 FindFeature 对象或 Features 集合中的所有图元 （SUBTRACT 集运算）。 Selection 集合将随后只包含不在传递入的对象中的图元。传入对象将保持不变。

语法

OBJECT**.Remove** (Source)

部分	说明
OBJECT	表示 Selection 集合。

部分	说明
Source	这是您将要从 OBJECT 移除的 **Feature 对象**、 **FindFeature 对象或 Features 集合**。

Selection.Replace 方法

目的

此方法使用 Feature 对象、Features 对象或 Features 集合取代集合中的内容。Selection 集合将随后只包含传递入的对象中的图元。传入对象将保持不变。

语法

OBJECT.**Replace** (Source)

部分	说明
OBJECT	表示 Selection 集合。
Source	这是您将取代 OBJECT 所用的 **Feature 对象**、 **FindFeature 对象或 Features 集合**。

Selection.SelectAll 方法

目的

选择图层中的所有图元。

语法

Map.Selection.SelectAll SelectionTypeConstant

另请参阅

SelectionTypeConstant

Selection.SelectByID 方法

目的

通过 FeatureID 或 FeatureKey 选择图元。 FeatureKey 是一个标识图元的字符串值。是由 **Feature.FeatureKey** 属性返回的值。 FeatureKey 可以在使用 FeatureID 的地方使用。

语法

OBJECT.**SelectByID** (*FeatureID, Flag)

部分	说明
OBJECT	表示 Selection 集合。
*FeatureID	要选择的图元的 FeatureID。您还可以使用 FeatureKey。
Flag	此属性控制这一选择的图元是添加、移除还是取代当前选择。取值为 **SelectionTypeConstants** 值。

注：尽管 FeatureID 参数仍然是 MapX 函数部分，但是为了简化和方便，建议您使用 FeatureKey。FeatureKey 用于确定表中的唯一记录。在此前的版本中， FeatureID 属性用于此目的，但是对于无缝和远程图层无法正常工作。 FeatureKey 属性适用于所有图层类型。

Selection.SelectByPoint 方法

目的

此方法选择图层中指定位置处的图元。

语法

OBJECT.**SelectByPoint** (X, Y, SelectFlag, [SearchResultFlag])

部分	说明
OBJECT	表示 Selection 集合。
X	要选择点所在的 X 坐标，双精度值 （经度）。
Y	要选择点所在的 Y 坐标，双精度值 （纬度）。
SelectFlag	此属性控制这一选择的图元是添加、移除还是取代当前选择。取值为 **SelectionTypeConstants** 值。
SearchResultFlag	此可选参数控制返回的内容。它取 **SearchResultTypeConstants** 的一个值或组合值。

注解

如果您选择区域，则 Selection.SelectByPoint 方法将选择包含该位置的区域。如果您选择点或线，则方法将选择该位置 3 个像素之内的点或线。

通过合并 （使用 OR 运算符）一个或多个 SearchResultTypeConstants，您可以定义您是否需要所有图元、最顶层的图元、所有区域、所有点、最顶端的点等。默认为返回所有图元。

Selection.SelectByRadius 方法

目的

选择点的指定半径之内的图层中的图元。标志 (SelectionTypeConstant) 控制这些选择的图元是添加、移除还是取代当前选择。

语法

OBJECT.**SelectByRadius** (X, Y, Radius, Flag)

部分	说明
OBJECT	表示 Selection 集合。
X	圆心的 X 坐标。双精度值 （经度）。
Y	圆心的 Y 坐标。双精度值 （纬度）。
Radius	搜索的半径。双精度值，指定到采用地图单位指定 X 和 Y 坐标的点的距离。
Flag	此属性控制所选对象是添加、移除还是取代当前选择。取值为 **SelectionTypeConstants** 值。

注解

如果且只有图元的质心在半径之内，则相应的图元才视为位于半径 “之内”。如果需要进一步限制选择标准，您可能需要使用 Layer 图像的 SearchWithinDistance 方法。

Selection.SelectByRectangle 方法

目的

选择矩形中的图元。

语法

OBJECT.**SelectByRectangle** (X1, Y1, X2, Y2, Flag)

部分	说明
OBJECT	表示 Selection 集合。
X1	矩形的第一个 X 坐标。双精度值。
Y1	矩形的第一个 Y 坐标。双精度值。
X2	矩形的第二个 X 坐标。双精度值。
Y2	矩形的第二个 Y 坐标。双精度值。

部分	说明
Flag	此属性控制所选对象是添加、移除还是取代当前选择。取值为 **SelectionTypeConstants** 值。

注解

如果且只有图元的质心在矩形之内，则相应的图元才视为位于矩形 “之内”。如果需要进一步限制选择标准，您可能需要使用 Layer 图像的 SearchWithinRectangle 方法。 Rectangle 是位于数字投影中的 Map 矩形，并且不显示屏幕矩形。

另请参阅

Layer.SearchWithinRectangle 方法，第 360 页

Selection.SelectByRegion 方法

目的

从区域中的图层选择图元。标志 (SelectionTypeConstant) 控制这些选择的图元是添加、移除还是取代当前选择。

语法

OBJECT.**SelectByRegion** (Layer, *FeatureKey, Flag)

部分	说明
OBJECT	表示 Selection 集合。
图层	所选区域所在的图层。取值为 Layer 对象。
FeatureKey	* 在其中进行选择的区域的 FeatureKey （字符串）。是 FeatureID 参数的替代参数。 FeatureID 的工作和以前相同，但建议使用新的 FeatureKey 参数。 FeatureKey 用于确定表中的唯一记录。在此前的版本中， FeatureID 属性用于此目的，但是对于无缝和远程图层无法正常工作。 FeatureKey 属性适用于所有图层类型。
Flag	此属性控制所选对象是添加、移除还是取代当前选择。取值为 **SelectionTypeConstants** 值。

注解

如果且只有图元的质心在半径之内，则相应的图元才视为位于区域 “之内”。如果需要进一步限制选择标准，您可能需要使用 Layer 图像的 SearchWithinFeature 方法。

另请参阅

Feature.FeatureKey 属性，第 274 页

Layer.SearchWithinFeature 方法，第 359 页

Layer 对象和 Layers 集合，第 340 页

SourceRow 对象和 SourceRows 集合

Dataset.SourceRows 属性用于从原始数据源取行集 （作为 SourceRow 对象），这些行将汇集在一起用于 MapX Dataset 中的行。

使用此对象来确定原始数据源中哪一行或哪些行将由 MapX Dataset 中的行表示。在使用 Selections 和 Feature 对象时此对象尤为实用。

对象属性

- **SourceRow.Row 属性**

集合属性

- **SourceRows.Count 属性**
- **SourceRows.Item 属性**

SourceRow.Row 属性

目的

包含源自原始数据源的行号。该属性是一个整数值，是只读的。

行号从 1 开始，并且按照数据从原始源数据读取的顺序依次排列。这是 SourceRow 对象的默认属性。

SourceRows.Count 属性

目的

包含集合中的 SourceRow 对象数。是基于 1 的整数值，是只读的。

SourceRows.Item 属性

目的

此属性从集合取 SourceRow 对象。索引用于指定要获取哪一 SourceRow。该索引是一个从 1 到 SourceRows.Count 的整数值，或者是 Feature 对象。这是 SourceRows 集合的默认属性。

语法

[SourceRow=]**OBJECT.Item (index)**

部分	说明
OBJECT	表示 SourceRows 对象。
index	要从中取值的 Dataset 中的行。这是一个变量，可以是行号后或 **Feature** 的对象。

Dataset.SourceRows 属性是一个 **SourceRows** 集合，其中对于原始数据源中的每个行，包含一个 SourceRow 对象，这些对象将聚集为 MapX **Dataset** 中的行。

您可以使用 Feature 对象作为 Row 参数，由于图元唯一标识数据集绑定到 Layer 中的单一的行。

State 对象

使用 State 对象可保存任何项目 （例如对象或变量）的当前状态，而您以后就可将该项目的状态恢复为当前所保存的状态。可以保存和恢复 MapX 对象或变量 （例如，字符串或布尔值）。

State 对象在 MapXtreme 应用程序对每一会话请求恢复 Map 对象状态时非常实用。 (请参阅第 461 页的 “State.Stream 属性”以获取信息 ... 如果要将 Datasets 和 Themes 保存和恢复到永久文件，以便在此后用于恢复， State 对象也非常使用。请参阅第 462 页的 “State.WriteToFile 方法”以获取详细信息。

注：此对象在 MapX Mobile 中不可用。

在 Visual Basic 中创建 State 对象

要在 Visual Basic 中使用 State 对象，需要向项目添加 "MapInfo MapX V5 State Type Library" 引用。您还可以随后使用对象类型 MapXStateLib.State 来创建新的状态对象。

以下示例创建新的 State 对象：

```
Dim objState as MapXStateLib.State
Set objState = new MapXStateLib.State
```

在 ASP 中创建 State 对象

要在 ASP 中使用 State 对象，需要使用 Server.CreateObject 和状态对象类 ID "MapX.State.5"。

以下示例创建新的 State 对象：

```
Dim objState
set objState = Server.CreateObject("MapX.State.5")
```

保存和恢复项目

您可以保存和恢复 Visual Basic 原有类型的单一元素 （例如字符串、变量、布尔值等）。不支持数组变量。

只要支持永久性，即可保存和恢复对象。以下 MapX 对象支持永久性：Map、 Annotation、 Annotations、 Layer、 Layers、 Labels、 LabelProperties、 Feature、 Dataset、 Datasets、 Theme、 Themes、 CoordSys、 Style、 Field 和 Fields。

有关保存和恢复项目的详细信息，请参阅 State.Save 和 State.Restore。

保存和恢复 Map 对象

保存 Map 对象时，其层次之中的所有 Map 属性和对象也将同时保存 （例如 Layers、 Title、 Datasets、 Themes 等）。唯一的例外是 Annotations 集合。如果要保存注释，需要在保存和恢复 Map 对象之后进行显式保存和恢复。

例如，要保存地图和注释，需要执行以下操作：

```
objState.Save objMap
objState.Save objMap.Annotations
```

要在随后恢复地图及其注释，需要执行以下操作：

```
objState.Restore objMap
```

```
objState.Restore objMap.Annotations
```

保存和恢复 Map 对象也将保存 Datasets 集合对象。但要完全恢复每一数据集， Datasets 集合可能还需要每一数据集外部资源 （例如 ADO 记录集）。有关详细信息，请参阅 “保存和恢复数据集”。

保存和恢复数据集

保存和恢复 Datasets 集合或 Map 对象时，您还需要保存和恢复地图中的每一数据集。

要创建数据集，需要使用外部数据源。这一来源可以是 ADO 记录集、 RDO 结果集或图层，具体取决于您正在创建的数据集类型。 MapX 将使用此来源创建数据集。要从状态对象恢复数据集，您需要为 MapX 提供这一来源，以便完全恢复数据集。您可以通过调用 Datasets.Restore 来实现这一操作。

如果现有数据集等同于正在恢复的数据集， Map.ReuseEquivalentOnRestore 属性允许 MapX 重复使用现有数据集。有关此属性的详细信息，请参阅 Map.ReuseEquivalentOnRestore 属性，第 414 页。

以下示例保存并恢复带有名为 Dataset1 数据集的 Map 对象：

```
' 保存地图：
objState.Save objMap
' 执行其他操作
...
' 如果等同，则在恢复状态时允许复用现有图层和 / 或数据集。
objMap.ReuseEquivalentOnRestore = True
' 恢复地图对象 ：
objState.Restore objMap
' 如果数据集仍未出现 ：
if not objMap.Datasets.Contains("Dataset1") then
    ' 设置数据集来源 （例如 ADO 记录集）：
    ...
    ' 完成数据集恢复：
    objMap.Datasets.Restore "Dataset1", source
end if
```

显式保存 Layers 集合或 Layer 对象。

在显式保存 Layers 集合或 Layer 对象时，不会保存和图层关联的 Datasets。如果显式保存 Layers 或 Layer 对象，必须显式保存 Datasets 或 Dataset 对象。

显式保存 Dataset 对象注意事项

保存和恢复单个数据集只保存有关该数据集的属性信息。为保存和恢复单个数据集，必须具有现有数据集。这意味着在对于您具有的数据集而言，需要恢复的唯一信息是数据集 （无论数据集是否用于标记图层）及其主题的标记状态。

另请参阅

Datasets.Contains 方法，第 265 页

Map.ReuseEquivalentOnRestore 属性，第 414 页

Dataset 对象和 Datasets 集合，第 257 页

对象方法

- **State.ReadFromFile 方法**
- **State.Reset 方法**
- **State.Restore 方法**
- **State.Save 方法**
- **State.WriteToFile 方法**

对象属性

- **State.Stream 属性**

State.ReadFromFile 方法

目的

此方法从二进制文件初始化 State 对象。

语法

OBJECT.**ReadFromFile** (strFile)

部分	说明
OBJECT	这是一个 State 对象。
strFile	String 参数，是文件路径，例如 "C:\Temp\State.sav"。

注解

在运行于因特网服务器端脚本 （例如 ASP）上使用此方法时，您需要确保执行脚本的用户帐户具有相应文件规格的读取权限。但切记如果需要同时调用 State.WriteToFile，则还需要写入权限。

另请参阅

State.WriteToFile 方法，第 462 页

State.Reset 方法

目的

此方法将状态对象恢复到其保存或初始化存储的起始状态。在恢复多个存储对象和 / 或变量时，此方法尤为实用。

示例

使用以下语句保存地图：

```
Stateobj.Save mapobj
```

使用以下语句恢复两个复本：

```
Stateobj.Restore mapobj1
```

```
Stateobj.Reset
Stateobj.Restore mapobj2
```

另请参阅

State.ReadFromFile 方法，第 458 页

State.Restore 方法，第 459 页

State.Save 方法，第 460 页

State.Stream 属性，第 461 页

State.Restore 方法

目的

此方法将项目恢复到 State 对象的内部存储。项目可以是支持永久性 （例如 Map 对象）的对象或变量（例如 String 变量）。您不能恢复数组 （例如 String(5) 变量）。为了正确恢复项目，您必须采用和 State.Save 相同的次数和顺序来调用 State.Restore。顺序不正确将会导致出错。

语法

OBJECT.**Restore** (item)

部分	说明
OBJECT	这是一个 State 对象。
item	这是要恢复的对象或变量。

示例

MapX 地图对象是永久性对象。要恢复此前保存的 Visual Basic 表单内嵌对象，需要执行以下操作：

```
objState.Restore Map1.Object
```

注：对象是 VB 中的一个属性，用于访问嵌入控件的 OLE 对象。

要恢复此前保存在 ASP 中的地图对象 （或任意其他 web 服务器应用程序），并且没有 Object 属性，因此您只能直接发送实际的对象：

```
objState.Restore objMap
```

注：objMap 必须为现有 Map 对象。

要恢复此前保存的 3 个变量，例如地图缩放、中点 X 坐标和中点 Y 坐标，可执行以下操作：

```
Dim nZoom
Dim nCenterX
Dim nCenterY
'保存状态
objState.Save Map1.Zoom
objState.Save Map1.CenterX
objState.Save Map1.CenterY
'恢复状态调用采用相同的顺序
```

```
objState.Restore nZoom
objState.Restore nCenterX
objState.Restore nCenterY
objMap.ZoomTo nZoom, nCenterX, nCenterY
```

注解

您可以多次调用此方法来恢复多个项目。

在恢复永久性对象时，必须具有现有对象，而不只是空对象变量。

您还可以多次调用 State.Restore 来恢复相同 State 对象中的多个多项或变量，只要初始调用的 State.Save 采用了等同的对象和变量次数和顺序即可。

如果在设置 State.Stream 属性或调用 State.Save 或 State.ReadFromFile 之后调用 State.Restore，您无需在第一次 State.Restore 调用之前调用 State.Reset，因为这是自动进行的。

您不能直接恢复对象的属性。您必须将其恢复到一个变量，然后设置对象属性。例如，要保存 Map.Zoom 属性，则执行以下操作：

```
objState.Save Map1.Zoom
```

但是要将其恢复，则必须执行以下措施：

```
Dim nZoom as Double
objState.Restore nZoom
Map1.Zoom = nZoom
```

另请参阅

State.Save 方法，第 460 页

State.Save 方法

目的

此方法将项目保存到 State 对象的内部存储。项目可以是支持永久性 （例如 Map 对象）的对象或变量（例如 String 变量）。您不能保存数组 （例如 String(5) 变量）。

语法

部分	说明
OBJECT	这是一个 State 对象。
item	这是要恢复的对象或变量。

示例

MapX 地图对象是永久性对象。要保存此前嵌入到 Visual Basic 表单中的地图对象，需要执行以下操作：

```
objState.Save Map1.Object
```

注：对象是 VB 中的一个属性，用于访问嵌入控件的 OLE 对象。

要保存 ASP 中的地图对象 （或任意其他 web 服务器应用程序），并且没有 Object 属性，因此您只能直接发送实际的对象：

```
objState.Save objMap
```

要保存 3 个变量，例如地图缩放、中点 X 坐标和中点 Y 坐标，可执行以下操作：

```
objState.Save objMap.Zoom
objState.Save objMap.CenterX
objState.Save objMap.CenterY
```

注解

您可以多次调用此方法来保存多个项目。

以下 MapX 对象支持永久性：Map、Annotation、Annotations、Layer、Layers、Labels、LabelProperties、Feature、 Dataset、 Datasets、 Theme、 Themes、 CoordSys、 Style、 Field 和 Fields。

另请参阅

State.Restore 方法，第 459 页

State.Stream 属性

目的

此属性将取或设置表示 State 对象内部存储的字符的字母字符串。

示例

您可以在 ASP 应用程序中的 State 对象，根据每个会话请求，来保存和恢复 Map 对象的状态。以下 ASP 代码片断将 Map 对象的当前状态通过隐含的 HTML 表单字段发送回客户端：

```
Dim objState
Set objState = Server.CreateObject("MapX.State.5")
objState.Save objMap
Response.Write("<INPUT TYPE=HIDDEN NAME=""State"" VALUE=" &
objState.Stream & ">")
```

以下 ASP 代码片断根据下一 HTTP 请求将 Map 对象恢复到其此前的状态：

```
Dim objState
Set objState = Server.CreateObject("MapX.State.5")
objState.Stream = Request.Form.Item("State")
objState.Restore objMap
```

注解

Stream 属性值只包含字母字符。它不包含任意特殊的 HTML 字符，以便可以作为字符串值写入到 HTML 页中。

此属性在 web 应用程序按照每一会话请求保存和恢复 Map 对象状态时非常实用。您可以将 Map 对象保存在 State 对象中，然后使用 Stream 属性，在隐含 HTML 表单字段上将状态发送回客户端。在接收到下一会话请求时，您可以将新 State 对象的 Stream 属性设置为隐含 HTML 表单字段的值，然后调用 State.Restore 来恢复 Map 对象。

切记保存具有大量信息的 Map 对象时 （例如具有若干图层的地图或具有若干图元的临时图层）将会明显增加客户端和服务器之间的数据传输量。保存 Stream 属性的值还可以采用其他解决方案，即恢复数据库中 Stream 对象的值并将关键字发送回客户端。此关键字随后可以用于从数据库检索 Stream 值。

另请参阅

State.Restore 方法，第 459 页

State.Save 方法，第 460 页

State.WriteToFile 方法

目的

此方法将 State 对象写入到二进制文件。

语法

OBJECT.**WriteToFile** (*strFile*)

部分	说明
OBJECT	这是一个 State 对象。
strFile	String 参数，是文件路径，例如 "C:\Temp\State.sav"。

注解

在运行于因特网服务器端脚本 （例如 ASP）上使用此方法时，您需要确保执行脚本的用户帐户具有相应文件规格的写入权限。

另请参阅

State.ReadFromFile 方法，第 458 页

Style 对象

Style 对象包含用于绘制符号、线、区域和文本的属性。该对象包含用于所有图元类型的属性，即使特定的图元类型只使用属性的子集。属性名起始处的关键字表示其应用的图元类型。

对象属性

- **Style.LineColor 属性**
- **Style.LineInterleaved 属性**
- **Style.LineStyle 属性**
- **Style.LineStyleCount 属性**
- **Style.LineSupportsInterleave 属性**
- **Style.LineWidth 属性**
- **Style.LineWidthUnit 属性**
- **Style.MinVectorSymbolCharacter 属性**
- **Style.MaxVectorSymbolCharacter 属性**
- **Style.RegionBackColor 属性**
- **Style.RegionBorderColor 属性**
- **Style.RegionBorderStyle 属性**
- **Style.RegionBorderWidth 属性**
- **Style.RegionBorderWidthUnit 属性**
- **Style.RegionColor 属性**
- **Style.RegionPattern 属性**
- **Style.RegionTransparent 属性**
- **Style.SupportsBitmapSymbols 属性**
- **Style.SymbolBitmapColor 属性**
- **Style.SymbolBitmapName 属性**
- **Style.SymbolBitmapOverrideColor 属性**
- **Style.SymbolBitmapSize 属性**
- **Style.SymbolBitmapTransparent 属性**
- **Style.SymbolCharacter 属性**
- **Style.SymbolFont 属性**
- **Style.SymbolFontBackColor 属性**
- **Style.SymbolFontColor 属性**
- **Style.SymbolFontHalo 属性**
- **Style.SymbolFontOpaque 属性**
- **Style.SymbolFontRotation 属性**
- **Style.SymbolFontShadow 属性**
- **Style.SymbolType 属性**
- **Style.SymbolVectorColor 属性**
- **Style.SymbolVectorSize 属性**

- **Style.TextFont 属性**
- **Style.TextFontAllCaps 属性**
- **Style.TextFontBackColor 属性**
- **Style.TextFontColor 属性**
- **Style.TextFontDblSpace 属性**
- **Style.TextFontHalo 属性**
- **Style.TextFontOpaque 属性**
- **Style.TextFontRotation 属性**
- **Style.TextFontShadow 属性**

对象方法

- **Style.Clone 方法**
- **Style.DrawLineSample 方法**
- **Style.DrawRegionSample 方法**
- **Style.DrawSymbolSample 方法**
- **Style.DrawTextSample 方法**
- **Style.ExportLineSample 方法**
- **Style.ExportRegionSample 方法**
- **Style.ExportSymbolSample 方法**
- **Style.ExportTextSample 方法**
- **Style.PickLine 方法**
- **Style.PickRegion 方法**
- **Style.PickSymbol 方法**
- **Style.PickText 方法**

Style.Clone 方法

目的

此方法返回独立的 Style 对象，该对象是另一 Style 对象的复本。

语法

OBJECT.**Clone**

OBJECT 占位符表示 Style 集合。

注解

在引用另一对象拥有的 Style 对象时 （例如，使用 objFeature.Style 表达式），您对样式所做出的任意更改都将自动影响到样式的父对象。

要创建修改的样式，而不影响父对象，可使用 Clone 方法来创建样式的复本。由此方法返回的样式是独立的对象，对此 Style 对象做出的更改不会自动影响其他任何对象。您可以将独立的 Style 对象分配给任意对象的 Style 属性。

只有在原始 Style 对象支持位图符号的情况下，通过 Clone 方法创建的 Style 对象才支持位图符号。

Style.DrawLineSample 方法

目的

如果用户要在 HDC 中绘制线样式的示例，则将会使用此方法。

语法

OBJECT.**DrawLineSample** (HDC, Rectangle)

部分	说明
HDC	示例绘制到的 HDC。
Rectangle	在示例将要绘制到的 DC 中定义像素坐标的 Rectangle 对象。如果示例不需要显示整个矩形，则示例将在矩形中居中放置。

Style.DrawRegionSample 方法

目的

如果用户要在 HDC 中绘制区域样式的示例，则将会使用此方法。

语法

OBJECT.**DrawRegionSample** (HDC, Rectangle)

部分	说明
OBJECT	表示 Style 对象。
HDC	示例绘制到的 HDC。
Rectangle	在示例将要绘制到的 DC 中定义像素坐标的 MapXLib.Rectangle。如果示例不需要显示整个矩形，则示例将在矩形中居中。

Style.DrawSymbolSample 方法

目的

如果用户要在 HDC 中绘制符号样式的示例，则将会使用此方法。此方法不适用于 MetaFile 设备上下文。

语法

OBJECT**.DrawSymbolSample** (HDC, Rectangle)

部分	说明
OBJECT	表示 Style 对象。
HDC	示例绘制到的 HDC。
Rectangle	在示例将要绘制到的 DC 中定义像素坐标的 MapXLib.Rectangle。如果示例不需要显示整个矩形，则示例将在矩形中居中。

Style.DrawTextSample 方法

目的

如果用户要在 HDC 中绘制文本样式的示例，则将会使用此方法。

语法

OBJECT**.DrawTextSample** (HDC, Rectangle, SampleText)

部分	说明
OBJECT	表示 Style 对象。
HDC	示例绘制到的 HDC。
Rectangle	在示例将要绘制到的 DC 中定义像素坐标的 MapXLib.Rectangle。如果示例不需要显示整个矩形，则示例将在矩形中居中。
SampleText	函数用于绘制 TextStyle 示例的文本。

Style.ExportLineSample 方法

目的

导出线样式示例。

语法

OBJECT.ExportLineSample (Destination, Format, Width, Height, [BackColor])

部分	说明
OBJECT	表示 Style 对象。
Destination	放置输出结果的文件规格，例如 C:\Temp\Symbol.bmp。如果指定 CLIPBOARD 而不是文件路径，则图像将置于剪贴板。

部分	说明
Format	输出格式。这将取 **ExportFormatConstants** 的值。**注：**如果指定 miFormatGIF（生成 GIF 文件），请读取**用于 GIF 文件用户的许可要求。**
Width	输出宽度取双精度值，以像素指定宽度。
Height	输出高度取双精度值，以像素指定高度。
BackColor	变量。控制背景颜色。指定 OLE_COLOR 值可以是特定的纯色，或 Windows 系统颜色之一，例如 “Window 背景色”。（大部分容器和 Visual Basic 一样，在其属性页中显示特殊的颜色采集器对话框，用于 OLE_COLOR 类型的属性。）

Style.ExportRegionSample 方法

目的

导出区域样式示例。

语法

OBJECT.ExportRegionSample (Location, Format, Width, Height, [BackColor])

部分	说明
OBJECT	表示 Style 对象。
Location	放置输出结果的文件规格，例如 C:\Temp\Symbol.bmp。如果指定 CLIPBOARD 而不是文件路径，则图像将置于剪贴板。
Format	输出格式。这将取 **ExportFormatConstants** 的值。**注：**如果指定 miFormatGIF（生成 GIF 文件），请读取**用于 GIF 文件用户的许可要求。**
Width	输出宽度取双精度值，以像素指定宽度。
Height	输出高度取双精度值，以像素指定高度。
BackColor	**Variant**：控制背景颜色。指定 OLE_COLOR 值可以是特定的纯色，或 Windows 系统颜色之一，例如 “Window 背景色”。（大部分容器和 Visual Basic 一样，在其属性页中显示特殊的颜色采集器对话框，用于 OLE_COLOR 类型的属性。）

Style.ExportSymbolSample 方法

目的

导出符号样式示例。

语法

OBJECT.**ExportSymbolSample (**Location, Format, Width, Height, [BackColor])

部分	说明
OBJECT	表示 Style 对象。
Location	放置输出结果的文件规格，例如 C:\Temp\Symbol.bmp。如果指定 CLIPBOARD 而不是文件路径，则图像将置于剪贴板。
Format	输出格式。这将取 **ExportFormatConstants** 的值。**注：**如果指定 miFormatGIF（生成 GIF 文件），请读取**用于 GIF 文件用户的许可要求。**
Width	输出宽度取双精度值，以像素指定宽度。
Height	输出高度取双精度值，以像素指定高度。
BackColor	变量。控制背景颜色。指定 OLE_COLOR 值可以是特定的纯色，或 Windows 系统颜色之一，例如 “Window 背景色”。（大部分容器和 Visual Basic 一样，在其属性页中显示特殊的颜色采集器对话框，用于 OLE_COLOR 类型的属性。）

Style.ExportTextSample 方法

目的

导出文本样式示例。

语法

OBJECT.**ExportTextSample (**Location, Format, Width, Height, [BackColor])

部分	说明
OBJECT	表示 Style 对象。
Location	放置输出结果的文件规格，例如 C:\Temp\Symbol.bmp。如果指定 CLIPBOARD 而不是文件路径，则图像将置于剪贴板。
Format	输出格式。这将取 **ExportFormatConstants** 的值。**注：**如果指定 miFormatGIF（生成 GIF 文件），请读取**用于 GIF 文件用户的许可要求。**
Width	文本输出宽度。取双精度值，以像素指定宽度。
Height	文本输出的高度。取双精度值，以像素指定高度。
SampleText	字符串。要导出的示例文本。
BackColor	变量。控制背景颜色。指定 OLE_COLOR 值可以是特定的纯色，或 Windows 系统颜色之一，例如 “Window 背景色”。（大部分容器和 Visual Basic 一样，在其属性页中显示特殊的颜色采集器对话框，用于 OLE_COLOR 类型的属性。）

Style.LineColor 属性

目的

包含线颜色。用于线性对象。这是一个 OLE_COLOR 值。

Style.LineInterleaved 属性

目的

读写布尔值属性。如果墨笔样式支持使用 LineSupportsInterleave 属性插入，则该墨笔样式可用于生成插入墨笔样式。插入式的墨笔样式可以在地图上创建交集的外观，用于覆盖交集和线。

Style.LineStyle 属性

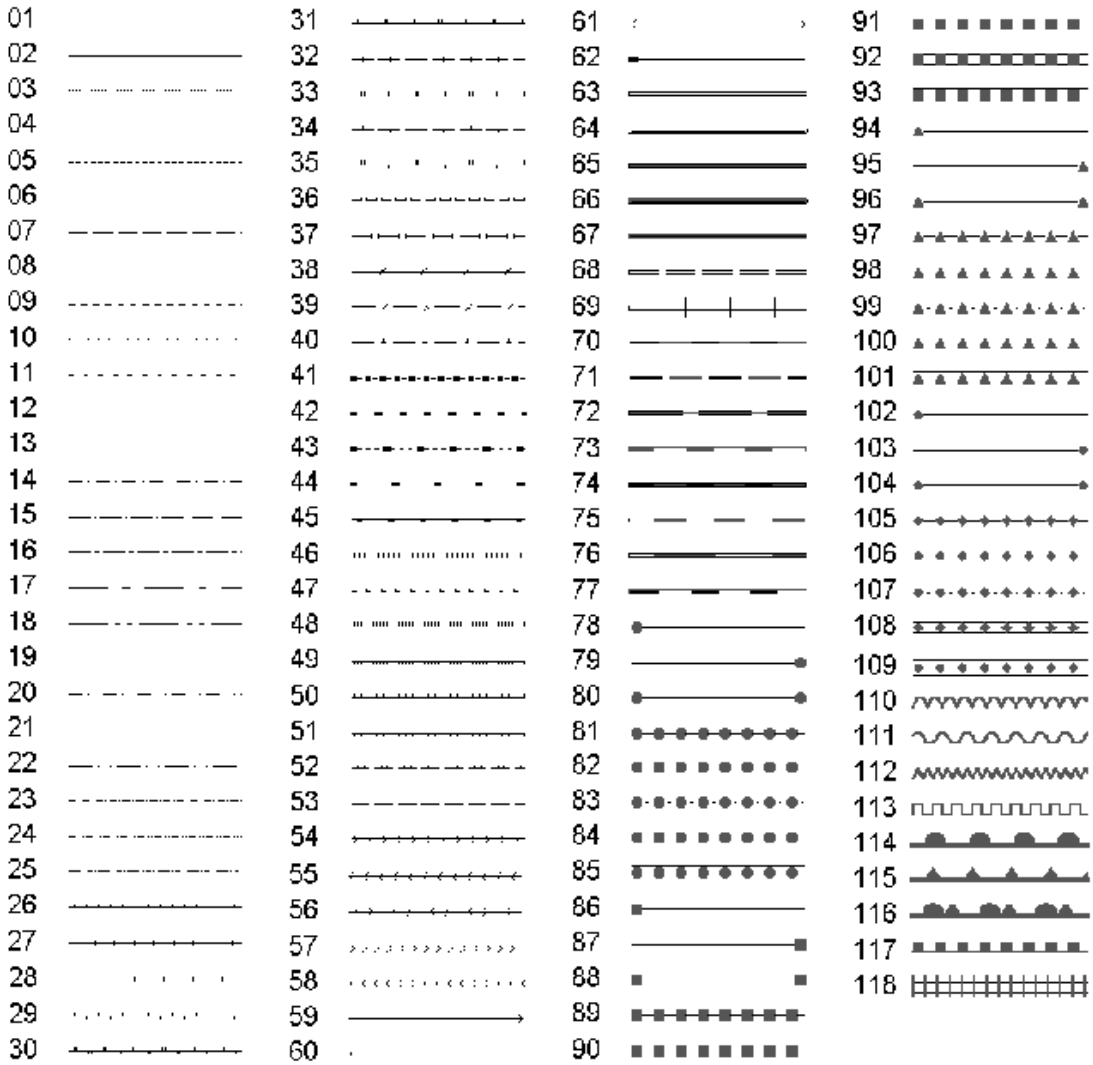
目的

此属性返回或设置线样式编号 （实线、虚线等），并用于显示线性图元。

注解

此属性是读写整数线样式编号：0 表示不可见线， 1 表示实线，较高的数字表示模式线，参阅下表。

您可以将任意 **PenStyleConstants** 值指定给此属性。不过，大部分线样式没有相应的常数。下表显示 LineStyle 数字的全集。



注：上表显示了默认样式。如果用户定制 mapx.pen 文件 （该文件定义线样式），则可用样式集可能不会匹配上表。使用 LineStyleCount 属性来确定 .pen 文件中的样式数量。

Style.LineStyleCount 属性

目的

这是一个只读整数值，表示可用线样式的数量。该数量包括从 mapx.pen 文件读取的墨笔样式。

Style.LineSupportsInterleave 属性

目的

只读布尔值属性。说明给定的墨笔形式是否支持插入。

另请参阅

Style.LineInterleaved

Style.LineWidth 属性

目的

此属性包含线宽，用于线性对象。这是一个整数值。

注解

您还可以使用十分之一磅或像素为单位，通过将 **Style.LineWidthUnit** 属性设置为 **StyleUnitConstants** 之一，来指定 Style.LineWidth 样式。

Style.LineWidthUnit 属性

目的

控制线宽的测量单位。宽度通过指定 **StyleUnitConstants** 值，使用十分之一磅或像素为单位指定。

另请参阅

Style.LineWidth

Style.PickLine 方法

目的

这将显示一个 Line 样式选择对话框，允许用户选择线样式属性。 Style 对象在用户单击对话框中的“OK”后更新为新属性。

语法

OBJECT.**PickLine**

Style.PickRegion 方法

目的

这将显示一个 Region 样式选择对话框，允许用户选择区域样式属性。 Style 对象在用户单击对话框中的“OK”后更新为新属性。

语法

OBJECT.**PickRegion**

Style.PickSymbol 方法

目的

这将显示一个 Symbol 样式选择对话框，允许用户选择符号样式属性。 Style 对象在用户单击对话框中的“OK”后更新为新属性。

语法

OBJECT.PickSymbol

Style.PickText 方法

目的

这将显示一个 Text 样式选择对话框，允许用户选择文本样式属性。 Style 对象在用户单击对话框中的“OK”后更新为新属性。

语法

OBJECT.PickText

Style.RegionBackColor 属性

目的

控制区域背景颜色，并在为区域对象指定 RegionPattern 而不是 miPatternSolid 时使用。这是一个 OLE_COLOR 值。

另请参阅

FillPatternConstants

Style.RegionBorderColor 属性

目的

包含区域边界颜色，用于区域对象。这是一个 OLE_COLOR 值。

Style.RegionBorderStyle 属性

目的

包含用于区域边界的线样式，用于区域对象。取值为 **PenStyleConstants** 值。

Style.RegionBorderWidth 属性

目的

包含用于区域边界的宽度，用于区域对象。该属性是一个整数值，以像素指定宽度。

Style.RegionBorderWidthUnit 属性

目的

控制区域边界宽度的测量单位。宽度通过指定 **StyleUnitConstants** 值，使用十分之一磅或像素为单位指定。

另请参阅

Style.RegionBorderWidth 属性，第 473 页

Style.RegionColor 属性

目的

包含区域颜色，用于区域对象。这是一个 OLE_COLOR 值。

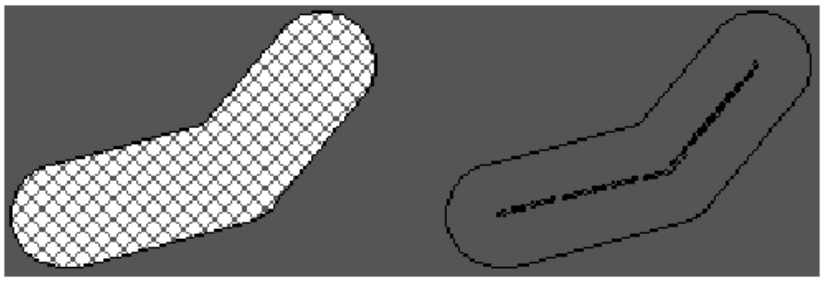
Style.RegionPattern 属性

目的

包含区域模式，用于区域对象。取值为 **FillPatternConstants** 值。

Style.RegionTransparent 属性

目的



lse 则表示不透明。

注解

为透明。如果地图上的图

区域之下的图元。

RegionTransparent 属性对于使用实填充模式的区域没有作用。

Style.SupportsBitmapSymbols 属性

目的

这是一个只读布尔值，表示样式对象是否支持位图符号。

注解

值为 True 表示 Symbol 对象支持位图符号。

当前，只有 Feature、 Layer 和 Theme 对象支持位图符号。

另请参阅

BitmapSymbol 对象 和 BitmapSymbols 集合，第 247 页

Style.SymbolBitmapColor 属性

目的

读写 OLE_COLOR 值，用于覆盖位图符号上的非白色像素。

注解

要令位图符号中的非白色像素显示为特定颜色，可执行以下操作：

1. 将 SymbolBitmapColor 属性设置为预期颜色。

2. 将 SymbolBitmapOverrideColor 属性设置为 True。

在将 TrueType 字体符号更改为位图符号时，换而言之，当您将 SymbolType 属性更改为 miSymbolTypeBitmap (1) 时， MapX 将 SymbolBitmapColor 属性设置为匹配 SymbolFontColor 属性。

另请参阅

BitmapSymbol 对象 和 BitmapSymbols 集合，第 247 页

Style.SymbolBitmapName 属性

目的

读写字符串值，表示用作位图符号的位图文件的名称。

注解

指定的名称不应包括完全目录路径。位图文件必须位于定制符号目录中，当前定义为：

```
<path>\CUSTSYMB
```

其中 <path> 是 MapX 公共文件安装路径，例如 C:\Program Files\Common Files\MapInfo Shared\MapXCommon。

另请参阅

BitmapSymbol 对象 和 BitmapSymbols 集合，第 247 页

Style.SymbolBitmapOverrideColor 属性

目的

读写布尔值，表示是否采用覆盖颜色绘制位图符号上的非白色像素。

注解

默认值为 false，表示位图符号中的非白色像素将显示为其默认颜色。如果需要所有的非白色像素采用覆盖颜色显示 （由 SymbolBitmapColor 属性指定的颜色），可将此属性设置为 True。

另请参阅

BitmapSymbol 对象 和 BitmapSymbols 集合，第 247 页

Style.SymbolBitmapSize 属性

目的

这是一个读写整数值，以磅为单位表示符号的大小。 72 磅等于 1 英寸。

注解

在将 TrueType 字体符号更改为位图符号时，换而言之，当您将 SymbolType 属性更改为 miSymbolTypeBitmap (1) 时， MapX 将 SymbolBitmapSize 属性设置为匹配 SymbolFont.Size 属性的值。例如，如果将 12 磅的 TrueType 字体符号更改为位图符号，则 SymbolBitmapSize 将等于 12。

SymbolBitmapSize 磅值从 1 至 48。如果位图符号小于 1，则 SymbolBitmapSize 将设置为 1。相应的，如果位图符号大于 48 磅，则 SymbolBitmapSize 设置为 48。

另请参阅

BitmapSymbol 对象 和 BitmapSymbols 集合，第 247 页

Style.SymbolBitmapTransparent 属性

目的

读写布尔值，包括位图符号的白色部分是否绘制为透明。

注解

默认值为 False，表示位图符号中的白色部分是否显示为白色。如果要令位图的白色部分透明，可将其设置为 True。

另请参阅

BitmapSymbol 对象 和 BitmapSymbols 集合，第 247 页

Style.SymbolCharacter 属性

目的

此属性指定要使用的符号。用于符号对象。此属性取 0 和 255 之间的整数值，并以 TrueType 字体集表示要使用的字符 ASCII 代码。

另请参阅

Style.SymbolFont 属性，第 476 页

Style.SymbolFont 属性

目的

此属性返回标准的 Font 对象 （IfontDispatch， OLE_FONT），允许您设置字体属性 （粗体、大小等），更改符号外观，并且可以只应用于 TrueType 字体符号。

注解

符号可以组合使用 TrueType 字体和该字体中的字符来定义 (Style.SymbolCharacter)。因此，如果您需要修改符号的外观，则可能需要获取并修改符号的 Style.SymbolFont 属性。

SymbolFont 属性为只读，但是其返回的 Font 对象属性则可以设置。

另请参阅

Style.SymbolCharacter 属性，第 476 页

设置字体属性，第 110 页

Style.SymbolFontBackColor 属性

目的

此属性包含符号背景色，用于符号对象，并可以只应用于 TrueType 字体符号。这是一个 OLE_COLOR 值。

Style.SymbolFontColor 属性

目的

包含符号的颜色。用于符号对象，可以只应用于 TrueType 字体符号。这是一个 OLE_COLOR 值。

Style.SymbolFontHalo 属性

目的

此读写属性控制符号周围是否绘制晕环。用于符号对象，可以只应用于 TrueType 字体符号。晕环是位于符号周围的 “缓冲”，有助于弥补其距离地图的偏移量。该属性是一个布尔值，可为单独图元将其设置为 True。

Style.SymbolFontOpaque 属性

目的

该属性为读写属性，控制符号是否显示背景色 （可为不透明）。用于符号对象，可以只应用于 TrueType 字体符号。是布尔值。

Style.SymbolFontRotation 属性

目的

读写整数值，表示旋转 TrueType 符号的度数。

注解

此属性应该设置为介于 0 和 360 之间的整数。顺时针旋转。

此属性只应用于 TrueType 字体符号，不适用于位图符号，后者不能旋转。

Feature、 Layer 和 Theme 对象支持旋转符号。如果对象不支持旋转符号，则此属性将被忽略。

Style.SymbolFontShadow 属性

目的

只应用于 Layer 样式和 Label 样式。此属性控制是否在符号下绘制阴影。用于符号对象，可以只应用于 TrueType 字体符号。是布尔值。

Style.MinVectorSymbolCharacter 属性

目的

此属性应用于 Symbol 样式对象。是只读整数属性，包含 Style.SymbolCharacter 的最小值，生成 Vector 符号。对于这些符号， Style.SymbolType 应该设置为 miSymbolTypeVector。

Style.MaxVectorSymbolCharacter 属性

目的

此属性应用于 Symbol 样式对象。是只读整数属性，包含 Style.SymbolCharacter 的最大值，生成 Vector 符号。对于这些符号， Style.SymbolType 应该设置为 miSymbolTypeVector。

Style.SymbolVectorColor 属性

目的

此属性应用于矢量符号 Style 对象。这是一个读写 OLE_COLOR 值，指定矢量符号的颜色。默认值为 miColorBlack。

Style.SymbolVectorSize 属性

目的

此属性应用于矢量符号 Style 对象。这是一个读写整数值，指定矢量符号的磅值。默认值为 12。

Style.TextFont 属性

目的

此属性返回标准的 Font 对象 （IfontDispatch， OLE_FONT），允许您设置字体属性 （粗体、大小等），更改文本图元的外观。

注解

TextFont 属性为只读，但是其返回的 Font 对象属性则可以设置。

Style.TextFontAllCaps 属性

目的

只应用于 Layer 样式和 Label 样式。控制文本是否以全部大写字母显示。用于文本对象。是布尔值。

Style.TextFontBackColor 属性

目的

包含文本背景色。用于文本对象。这是一个 OLE_COLOR 值。

另请参阅

OLE_COLOR 值，第 631 页

Style.TextFontColor 属性

目的

包含文本颜色。用于文本对象。这是一个 OLE_COLOR 值。

另请参阅

OLE_COLOR 值，第 631 页

Style.TextFontDblSpace 属性

目的

只应用于 Layer 样式和 Label 样式。控制文本是否以较大的字母间距显示。用于文本对象。是布尔值。

Style.TextFontHalo 属性

目的

此属性只应用于 Layer 样式和 Label 样式。控制是否在文本周围绘制光晕。用于文本对象。晕环是位于文本周围的 “缓冲”，有助于弥补其距离地图的偏移量。是布尔值。

Style.TextFontOpaque 属性

目的

控制文本是否显示背景色 （可为不透明）。用于文本对象，是布尔值。

Style.TextFontShadow 属性

目的

只应用于 Layer 样式和 Label 样式。控制是否在文本下绘制阴影。用于文本对象，是布尔值。

Style.TextFontRotation 属性

目的

读写属性，使用整数值表示旋转 Text 对象的度数。负整数表示顺时针旋转文本，正数表示逆时针旋转文本。它只对于 Feature 对象有效，如果设置 / 查询任何其他对象，则该属性将会忽略 / 未定义。

Style.SymbolType 属性

目的

读写属性的短精度值，表示如何显示点图元 （源自 TrueType 字体、位图或 VectorSymbol 的字符）。该属性应该设置为 **SymbolTypeConstants** 之一。

注解

要使用 TrueType 字体符号，可将其设置为 miSymbolTypeTrueTypeFont。

要更改为位图符号，可将其设置为 miSymbolTypeBitmap。

要更改为向量符号，可将其设置为 miSymbolTypeVector。

注：只有部分对象 （Feature、 Layer 和 Theme）支持位图符号。如果对象不支持位图符号，则尝试将 SymbolType 设置为 miSymbolTypeBitmap 将会报错。

另请参阅

BitmapSymbol 对象 和 BitmapSymbols 集合，第 247 页

Theme 对象和 Themes 集合

包含主题属性的主题对象。

每个数据集都包含一个 Themes 集合 (Dataset.Themes)。 Theme 集合具有用于从集合添加或移除 Theme 对象的方法和属性。

对象属性

- **Theme.AutoRecompute 属性**
- **Theme.ComputeTheme 属性**
- **Theme.DataMax 属性**
- **Theme.DataMin 属性**
- **Theme.Fields 属性**
- **Theme.Layer 属性**
- **Theme.Legend 属性**
- **Theme.Name 属性**
- **Theme.Properties 属性**
- **Theme.ThemeProperties 属性**
- **Theme.Type 属性**
- **Theme.Visible 属性**

对象方法

- **Theme.ThemeDlg 方法**

集合属性

- **Themes.Count 属性**
- **Themes.Item 属性**

集合方法

- **Themes.Add 方法**
- **Themes.Remove 方法**
- **Themes.RemoveAll 方法**

Theme.AutoRecompute 属性

目的

控制重新计算主题范围的时间。该属性是一个布尔值，默认值为 TURE。

注解

当主题属性更改时 （例如 ThemeProperties.NumRanges）， Theme 对象将会重新计算。不过，这可能非常耗时。如果要更改若干属性，则您应该将 AutoRecompute 设置为 FALSE，更改属性，然后将 AutoRecompute 设置为 TRUE，只重新计算一次。

Theme.ComputeTheme 属性

目的

此属性控制主题是否可以从表上的数据计算。该属性是一个布尔值，默认值为 True。值为 True 将会从原始数据计算主题。

注解

如果值设置为 False，则将会创建不可见的主题对象，带有 10 个用于单值主题的范围， 5 个用于范围主题的范围。您可以手工设置最小值和最大值，使用 **Theme.DataMin 属性** 和 **Theme.DataMax 属性**定义主题。 **ThemeProperties.NumRanges** 和 **IndividualValueCategory.Value** 属性用于进行修改。

对于范围主题，可以手工设置主题范围或计算由最小值 (Theme.DataMin) 和最大值 (Theme.DataMax) 确定的同等大小范围。

将此属性设置为 false 时， **Legend.ShowEmptyranges 属性**必须设置为 true，以便令主题的图例文本可见。这是由于 MapX 在 ComputeTheme 为 false 时不计算主题类别计数。

Theme.DataMax 属性

目的

在 Theme.ComputeTheme 属性或 Themes.Add 方法的 ComputeTheme 参数设置为 FALSE 时，确定设置主题范围或计算同等大小的范围主题范围的最大值。

另请参阅

Theme.ComputeTheme 属性，第 482 页

Theme.DataMin 属性，第 482 页

Theme.DataMin 属性

目的

在 Theme.ComputeTheme 属性或 Themes.Add 方法的 ComputeTheme 参数设置为 FALSE 时，确定设置主题范围或计算同等大小的范围主题范围的最小值。

另请参阅

Theme.ComputeTheme 属性，第 482 页

Theme.DataMin 属性，第 482 页

Theme.Fields 属性

目的

返回只读 Fields 集合，表示由作为主题基础的 Dataset 所用的字段集。

Theme.Layer 属性

目的

这是只读属性，返回表示主题基于的图层的 Layer 对象。

另请参阅

Layer 对象和 Layers 集合，第 340 页

Theme.Legend 属性

目的

每个 Theme 对象均有 Legend 对象 (Theme.Legend)。 Legend 对象包含控制主题图例显示的属性。每个 ThemeCategory 对象 （RangeCategory、 IndividualValueCategory 或 MultiVarCategory）在包含在 LegendText 对象的图例中都有一个条目。

另请参阅

Legend 对象，第 379 页

Theme.Name 属性

目的

Theme 的名称在 Themes 集合中必须唯一。这一读 / 写属性或者指定为 Themes.Add 方法的参数，或者由 MapX 在主题创建时生成。这是 Theme 对象的默认属性。

注解

该 Name 属性只能和 Theme.Item 属性和 Themes.Remove 方法一起使用。

另请参阅

Themes.Add 方法，第 485 页

Themes.Item 属性，第 486 页

Themes.Remove 方法，第 487 页

Theme.Properties 属性

目的

这是包含在 Theme 对象中的属性集，由正在使用中的主题地图确定类型。

注：在 MapX 3.5 中，我们将 Theme.Properties 重命名为 Theme.ThemeProperties，但是保留了对象模型中 Theme.Properties，以便代码向下兼容。但 Theme.ThemeProperties 是获取主题属性的首选方法。

另请参阅

Theme.ThemeProperties 属性，第 484 页

ThemeProperties 对象，第 488 页

Theme.ThemeDlg 方法

目的

打开普通对话框，允许用户控制 Theme 的样式。打开的特定对话框将匹配 Theme 对象的类型 （范围、点密度、饼形、条形、单值或标记符号）。如果用户单击 “确定”，则在对话框中所做的更改将立即应用于地图上的主题。

语法

[Boolean=]OBJECT.**ThemeDlg** ([HelpFile], [HelpID])

部分	说明
OBJECT	表示 Theme 对象。
HelpFile	HelpFile 是可选参数，是 .hlp 文件的路径名，包含用于对话框的帮助主体。
HelpID	HelpID 是可选参数，引用给定 .hlp 文件中的特定帮助主题 ID。

注解

如果没有给定任何给定参数，则帮助按钮将不会出现在对话框上。

如果用户单击 “确定”，则 ThemeDlg 的返回值为 True ；如果用户单击 “取消”则为 False。

Theme.ThemeProperties 属性

目的

包含有关主题属性的对象。包含在 ThemeProperties 对象中的属性集，由 Theme 对象 （主题地图类型）的类型确定。

另请参阅

ThemeProperties 对象，第 488 页

Theme.Type 属性

目的

Theme 对象的主题类型。这是一个 **ThemeTypeConstants** 值，是只读属性。

Theme.Visible 属性

目的

此属性设置主题是否可见。该属性是一个布尔值，默认值为 TRUE。

Themes.Add 方法

目的

此方法创建主题，并将其添加特定 Dataset 的 Themes 集合。

注解

在创建主题时，计算包含许多行的图层 （例如逐层细化图层或服务器图层）的范围时可能会需要一段时间。 ComputeTheme 参数允许您为任意主题类型创建非计算主题。非计算主题使您不必自动计算范围即可创建主题。然后可以自己创建范围。对于逐层细化图层和服务器图层，这种方法速度较快。

注：MapX 将根据空 Dataset 创建空主题。此时将创建主题，并将其添加到主题集合，但是由于以没有数据的 Dataset 为基础，因此主题将不可见。

没有行的 Dataset 只可能在程序员出错的情况下产生。例如，如果意外将 USA 数据和 Canada 图层绑定，则结果将为 Dataset 没有行 （即没有数据）。如果尝试基于该 Dataset 创建主题，将会创建一个空主题。程序员应该在创建主题之前检查 Dataset 中是否有任何行。

语法

OBJECT.**Add** ([Type], [Field], [Name], [ComputeTheme])

部分	说明
OBJECT	表示 Themes 对象。
Type	表示要创建的主题地图类型。取值为 **ThemeTypeConstants** 值。这是一个可选参数，如未指定 （或指定为 miThemeAuto），则 MapX 将尝试根据导入的字段数以及已显示的其它主题类型来选择适合的默认主题，如下所示。如果 MapX 无法选择默认主题类型，则将会出错。

部分	说明
Field(s)	指定主题地图的一个或多个字段。字段可以通过名称、索引或 Field 对象指定。如果您正在使用多个变量创建主题 （例如条形图或饼图），可传入 Fields 集合或字段名、索引或 Field 对象的数组。该参数是可选参数，如果未指定， MapX 将使用 Dataset 的第一个数字字段。
Name	指定主题地图的名称。这是一个字符串参数。该参数是可选参数，如果未指定，MapX 将生成一个名称，例如 StatesBySales。
ComputeTheme	布尔值。默认值为 True，从表数据计算主题。如果值设置为 False，则将会创建不可见的主题对象，带有 10 个用于 IndividualValue 主题的范围， 5 个用于范围主题的范围。您还可以手动设置最小值和最大值，使用 Theme.DataMin 和 Theme.DataMax 定义主题。对于范围主题，您可以手动设置主题范围，或计算由最小值 (Theme.DataMin) 和最大值 (Theme.DataMax) 确定的同等大小的范围。

另请参阅

Field 对象和 Fields 集合，第 302 页

Theme.DataMax 属性，第 482 页

Theme.DataMin 属性，第 482 页

Theme.Name 属性，第 483 页

Theme.Type 属性，第 485 页

MapX 字段命名规范，第 507 页

Themes.Count 属性

目的

Themes 集合中的 **Theme** 对象数量。这是一个只读属性。

Themes.Item 属性

目的

从 Themes 集合检索 Theme 对象。采用参数指定具体主题。参数是表示基于 1 的索引或主题名的变量。这是 Themes 集合的默认属性。

语法

OBJECT**.Item** (index)

部分	说明
OBJECT	表示 Themes 对象。

部分	说明
index	主题对象，基于 1 的索引编号或主题名称。

另请参阅

Theme.Name 属性，第 483 页

Themes.Remove 方法

目的

从 Themes 集合去除指定的 Theme 对象。

注：如果您去除项目，集合索引将再次计数，填充由已删除项目留下的空洞。

语法

OBJECT**.Remove** (index)

部分	说明
OBJECT	表示 Themes 对象。
index	主题对象，基于 1 的索引编号或主题名称。

另请参阅

Theme.Name 属性，第 483 页

Themes.RemoveAll 方法

目的

从集合中移除所有 Theme 对象。

语法

OBJECT**.RemoveAll**

ThemeProperties 对象

ThemeProperties 对象是 Theme 对象的属性之一，包含定义 Theme 的信息。

注：Theme.Properties 仍然和此前一样工作，但是建议您使用 ThemeProperties 对象，以便利用其改善的功能。

对象属性

- **ThemeProperties.ApplyAttribute 属性**
- **ThemeProperties.BarFramed 属性**
- **ThemeProperties.BarFrameStyle**
- **ThemeProperties.BarGraduatedStack 属性**
- **ThemeProperties.BarIndependentScale 属性**
- **ThemeProperties.BarStacked 属性**
- **ThemeProperties.BarWidth 属性**
- **ThemeProperties.BorderStyle 属性**
- **ThemeProperties.ColorMethod 属性**
- **ThemeProperties.DataValue 属性**
- **ThemeProperties.DistMethod 属性**
- **ThemeProperties.DotColor 属性**
- **ThemeProperties.DotSize 属性**
- **ThemeProperties.Graduated 属性**
- **ThemeProperties.GraduateSizeBy 属性**
- **ThemeProperties.Independent 属性**
- **ThemeProperties.IndividualValueCategories 属性**
- **ThemeProperties.InflectionColor 属性**
- **ThemeProperties.InflectionRange 属性**
- **ThemeProperties.InflectRanges 属性**
- **ThemeProperties.MultivarCategories 属性**
- **ThemeProperties.NegativeSymbolStyle 属性**
- **ThemeProperties.NumRanges 属性**
- **ThemeProperties.PieClockwise 属性**
- **ThemeProperties.PieGraduated 属性**
- **ThemeProperties.PieHalfPies 属性**
- **ThemeProperties.PieStartAngle 属性**
- **ThemeProperties.PositiveSymbolStyle 属性**
- **ThemeProperties.RangeCategories 属性**
- **ThemeProperties.RoundBy 属性**
- **ThemeProperties.RoundRanges 属性**
- **ThemeProperties.ShowNegativeValues 属性**

- **ThemeProperties.Size 属性**
- **ThemeProperties.SpreadBy 属性**
- **ThemeProperties.SymbolStyle 属性**
- **ThemeProperties.ValuePerDot 属性**
- **ThemeProperties.Width 属性**

ThemeProperties.AllowEmptyRanges 属性

目的

这是读写布尔值属性，控制是否在范围主题中允许空范围。对于新主题默认为 False，表明在范围主题中不允许空范围。

注解

如果 ThemeProperties.DistMethod 属性设置为 miCustomRanges (value: 0)，则将允许范围为空，无论 AllowEmptyRanges 属性取何值。

另请参阅

Legend.ShowEmptyRanges 属性，第 383 页

ThemeProperties.DataValue 属性

目的

适用于 GraduatedSymbol、 Pie 和 Bar 主题。 **Size** 属性与其组合使用，可控制主题图形在指定值处的大小。是双精度值，是主题地图在由 **Size** 属性指定的大小处绘制处的值。该属性的默认值设置为地图绘制图元的最大数据值。

另请参阅

ThemeProperties.Size 属性，第 492 页

ThemeProperties.DistMethod 属性

目的

此属性应用于 Ranged 主题。控制 Theme 对象重新计算之后，如何创建范围 （在 RangeCategories 集合之中）。此属性取值为 **DistribMethodConstants** 的值，默认值为 miEqualCountPerRange。可用的分布类型如下：

分布	说明
miEqualRangeSize	取最大数据值和最小数据值，通过等分数据值的范围 （例如 0-25、 25-50、 51-75、 76-100）来选择范围。

分布	说明
miEqualCountPerRange	尝试采用每个范围包含相同数量的图元的方法来指定范围。
miCustomRanges	您将在 RangeCategories 集合中指定自己的范围 （MapX 不会计算范围）。
miNaturalBreak	范围的分割通过使用尝试最小化数据值和数据平均值之间的差异，在每个范围基础上最小化的算法来确定。
miStandardDeviation	中间范围在数据均值处中断，高于和低于中间范围的范围高于或低于平均值一个标准差。

另请参阅

RangeCategory 对象和 RangeCategories 集合，第 434 页

ThemeProperties.DotSize 属性

目的

此属性应用于 DotDensity 主题。控制由 DotDensity 主题地图使用的点的大小。取值为 **DotSizeConstants**，默认值为 **miDotSizeSmall**。

ThemeProperties.Graduated 属性

目的

此属性应用于 Pie 主题。此属性控制饼图的大小是否根据切片的总值分级。该属性是一个布尔值，默认值为 TRUE。

注：此属性已经由 ThemeProperties.PieGraduated 取代。但是它仍然存在，以便向后兼容。

另请参阅

ThemeProperties.PieGraduated 属性，第 494 页

ThemeProperties.Independent 属性

目的

此属性应用于 Bar 主题。此属性控制柱体的数据值是否应独立对待 （不是人口和平均收入等可比较的值）。该属性是一个布尔值，默认值为 FALSE。对于多值柱状图主题，单个图元的每个柱体的数据与主题中其它字段的柱体无关，或数据字段的最大值明显不同，开发人员应将该属性设置为 True。人口柱状图主题就是该属性的一个示例，其中一个柱体可能代表州的人口，另一个柱体可能代表出口的等级。人口数据可能是百万级，而等级只是在 1 到 50 之间。如果 Independent 属性设置为 True，人口最多的州的人口柱体将等于出口柱体中最高等级状态的高度。如果 Independent 属性仍设置为 False，将很难从出

口等级柱状图获得任何含义，因为排在第一位的州如果只有一个人，出口柱状图将使用州人口柱状图的大小。

注：BarIndependentScale 目前是此属性的首选名称。

另请参阅

ThemeProperties.BarIndependentScale 属性，第 494 页

ThemeProperties.IndividualValueCategories 属性

目的

此属性返回 IndividualValueCategories 集合，用于单值主题地图。

另请参阅

IndividualValueCategory 对象和 IndividualValueCategories 集合，第 326 页

ThemeProperties.MultivarCategories 属性

目的

此属性返回 MultivarCategories 集合，用于使用多个变量的主题地图 （饼图或条形图）。

另请参阅

MultivarCategory 对象和 MultivarCategories 集合，第 420 页

ThemeProperties.NumRanges 属性

目的

此属性应用于 Ranged 范围。它包含范围主题地图的范围数量。该属性是一个整数值，默认为 5。

ThemeProperties.RangeCategories 属性

目的

此属性返回 RangeCategories 集合，用于范围主题地图。

另请参阅

RangeCategory 对象和 RangeCategories 集合，第 434 页

ThemeProperties.Size 属性

目的

此属性应用于 Bar 和 Pie 主题。此属性与 DataValue 组合使用，可控制主题图形在指定值处的大小。该属性使用 Paper 单位 (PaperUnit) 指定主题图形的高度。该属性是一个双精度值，默认值为 .25 英寸。

另请参阅

Map.PaperUnit 属性，第 412 页

ThemeProperties.DataValue 属性，第 489 页

ThemeProperties.SpreadBy 属性

目的

控制自动传播如何在范围主题地图上完成。 MapX 可以计算两色和之间的一组颜色或两种大小之间的计算大小。

注解

取值为 **SpreadByConstants** 值。要控制用于 ThemeProperties.RangeCategories(x).Style 值的起止设置的颜色或大小，其中 x = 1 代表第一个范围 （起始值）， x 是用于最后一个范围的 ThemeProperties.NumRanges （截止值）。

注：如果您正在使用主题范围，并且分别指定范围颜色 （例如每个范围），则 ThemeProperties.SpreadBy 必须设置为 miSpreadByNone。

ThemeProperties.SymbolStyle 属性

目的

此属性应用于 GraduatedSymbol 主题。此属性控制使用的符号以及按 DataValue 属性指定的值绘制的符号大小。这是一个 Style 对象，是只读属性 （尽管 SymbolStyle 的属性为读写）。

注：ThemeProperties.PositiveSymbolStyle 目前是此属性的首选名称。

另请参阅

ThemeProperties.PositiveSymbolStyle 属性，第 496 页

Style 对象，第 463 页

ThemeProperties.DataValue 属性，第 489 页

ThemeProperties.ValuePerDot 属性

目的

此属性应用于 DotDensity 主题。此属性指定点表示的值。例如，如果 ValuePerDot 为 2000，并且该区域人口为 48,000，则将在该区域内绘制 24 个点。该属性是一个双精度值，默认值为合理的大小。

ThemeProperties.Width 属性

目的

此属性应用于 Bar 主题。使用 Paper 单位 (Map.PaperUnit) 指定柱状图主题的每个柱体的宽度。该属性是一个双精度值，默认值为 .25 英寸。

另请参阅

Map.PaperUnit 属性，第 412 页

ThemeProperties.PieClockwise 属性

目的

该属性是一个布尔值，应用于饼图主题。它控制饼图边缘绘制的方向，值为 true 时顺时针方向绘制。默认值为 true。

另请参阅

ThemeProperties.PieStartAngle 属性，第 493 页

ThemeProperties.PieStartAngle 属性

目的

此属性应用于 Pie 图主题。是指定开始绘制第一个饼图边缘时的角度的整数属性。使用度为测量单位，逆时针增加， 0 度指向右侧。默认值为 180 度。

另请参阅

ThemeProperties.PieClockwise 属性，第 493 页

ThemeProperties.PieHalfPies 属性

目的

该属性是一个布尔值，应用于饼图主题。如果为 True，则将绘制半幅饼图来取代全幅饼图。默认值为 false。

ThemeProperties.PieGraduated 属性

目的

该属性是一个布尔值，应用于饼图主题。控制饼图切片的大小是否相同。如果为 True，则饼图切片的总计大小将按其总计值来标记。如果为 False，则所有饼图切片的大小均相同。默认值为 True。

注：此属性取代 ThemeProperties.Graduated。 PieGraduated 时首选属性。

另请参阅

ThemeProperties.GraduateSizeBy 属性，第 497 页

ThemeProperties.BarStacked 属性

目的

该属性是一个布尔值，应用于条形图主题。如果为 True，则条形图中的条块将相互堆叠在顶部，条块侧边接触。默认值为 false。

另请参阅

ThemeProperties.BarGraduatedStack 属性，第 494 页

ThemeProperties.BarGraduatedStack 属性

目的

该属性是一个布尔值，应用于 ThemeProperties.BarStacked 属性为真的条形图。控制条形图的总计大小是否相同。如果为 True，则条形图中的条块将按照图的总计值分级；否则所有图形的总计大小都相同。默认值为 True。

另请参阅

ThemeProperties.BarStacked 属性，第 494 页

ThemeProperties.GraduateSizeBy 属性，第 497 页

ThemeProperties.BarIndependentScale 属性

目的

此属性应用于 Bar 主题。控制柱体的数据值是否应独立对待 （不是人口和平均收入等可比较的值）。该属性是一个布尔值，默认值为 False。对于多值柱状图主题，单个图元的每个柱体的数据与主题中其它字段的柱体无关，或数据字段的最大值明显不同，开发人员应将该属性设置为 True。人口柱状图主题就是该属性的一个示例，其中一个柱体可能代表州的人口，另一个柱体可能代表出口的等级。人口数据可能是百万级，而等级只是在 1 到 50 之间。如果 Independent 属性设置为 True，人口最多的州的人口柱

体将等于出口柱体中最高等级状态的高度。如果 Independent 属性仍设置为 False，将很难从出口等级柱状图获得任何含义，因为排在第一位的州如果只有一个人，出口柱状图将使用州人口柱状图的大小。此属性取代 ThemeProperties.Independent。

ThemeProperties.BarWidth 属性

目的

该属性是一个双精度值，应用于 Bar 主题。使用 Paper 单位 (Map.PaperUnit) 指定柱状图主题的每个柱体的宽度。该属性是一个双精度值，默认值为 .25 英寸。此属性取代 ThemeProperties.Width 属性。

另请参阅

Map.PaperUnit 属性，第 412 页

ThemeProperties.BarFramed 属性

目的

此属性是一个布尔值，设置为 false，指定条块对象是否在框架内绘制。

ThemeProperties.BarFrameStyle

目的

确定条形图框架样式的 Style 对象。

另请参阅

Style 对象，第 463 页

ThemeProperties.BorderStyle 属性，第 495 页

ThemeProperties.BorderStyle 属性

目的

影响条形图和饼图的 Style 对象。控制围绕饼图或条形图的边界的线形样式。

另请参阅

Style 对象，第 463 页

ThemeProperties.BarFrameStyle，第 495 页

ThemeProperties.ShowNegativeValues 属性

目的

该属性是一个布尔值，应用于 Graduated Symbol 主题。如果为 True，则负值将按其绝对值处理，符号表示 -4000 和符号表示 4000 的大小相同。如果为 False，则将会显示负值。默认值为 false。

另请参阅

ThemeProperties.PositiveSymbolStyle 属性，第 496 页

ThemeProperties.NegativeSymbolStyle 属性，第 496 页

ThemeProperties.PositiveSymbolStyle 属性

目的

此属性应用于 GraduatedSymbol 主题。控制用于所有正数的符号，以及按 DataValue 属性指定的值绘制的符号大小。这是一个 Style 对象，读写属性 （尽管 Style 对象的属性为读写）。

注：此属性取代 ThemeProperties.SymbolStyle。

另请参阅

Style 对象，第 463 页

ThemeProperties.ShowNegativeValues 属性，第 496 页 y

ThemeProperties.NegativeSymbolStyle 属性，第 496 页

ThemeProperties.DataValue 属性，第 489 页

ThemeProperties.NegativeSymbolStyle 属性

目的

该属性应用于 Graduated Symbol 主题，其 ThemeProperties.ShowNegativeValues 属性为真。是一个 Style 对象，控制用于主题所有负数的符号。读写属性 （尽管 Style 对象的属性为读写）。

另请参阅

Style 对象，第 463 页

ThemeProperties.ShowNegativeValues 属性，第 496 页 y

ThemeProperties.NegativeSymbolStyle 属性，第 496 页

ThemeProperties.DataValue 属性，第 489 页

ThemeProperties.GraduateSizeBy 属性

目的

此属性应用于 Graduated Symbol 主题，以及设置为分级的饼图和条形图主题。它必须设置为短精度值，匹配一个 **GraduationConstants** 的值。控制数据值和地图上相应符号或图形的大小之间的关系。默认值为 miGraduateBySquareRoot。

另请参阅

ThemeProperties.PieGraduated 属性

ThemeProperties.BarGraduatedStack 属性

ThemeProperties.DotColor 属性

目的

此属性应用于 DotDensity 主题。控制主题中的点的颜色。这是一个 **OLE_COLOR** 值。

另请参阅

OLE_COLOR 值，第 631 页

ThemeProperties.RoundRanges 属性

目的

该属性是一个布尔值，应用于 Ranged 主题。控制是否圆整范围的边界。默认值为 True。值的圆整量由 ThemeProperties.RoundBy 属性控制。

另请参阅

ThemeProperties.RoundBy 属性，第 497 页

ThemeProperties.RoundBy 属性

目的

该属性是一个双精度值，应用于 Ranged 主题。指定将范围圆整到间隔。只在 ThemeProperties.RoundRanges 为 True 时适用。

注解

如果 RoundBy 为 100，计算范围是 420-560 和 560-930，则将分别圆整到 400-600 和 600-900。默认的 RoundBy 通过 10 次幂来计算，最接近于最大值和最小值之间的差异。例如，如果最大值和最小值之间的差为 11,554，则 RoundBy 属性的默认值将为 100。

另请参阅

ThemeProperties.RoundRanges 属性，第 497 页

ThemeProperties.InflectRanges 属性

目的

该属性是一个布尔值，应用于 Ranged 主题。在范围主题中，可用弯曲点来将数据和范围分割为两个部分。在 InflectRanges 为 true 时，范围主题中的颜色将从顶部范围颜色延展到 InflectionColor，然后再从 InflectionColor 延展到底部范围颜色。默认值为 false。

注解

例如，如果 InflectionColor 是 miColorWhite，并且顶部和底部范围颜色分别为蓝色和红色，则颜色将从蓝到白，再从白到红。如果 InflectRanges 为 False，则中间范围的颜色将为红蓝两色的混色所替代。

另请参阅

ThemeProperties.InflectionRange 属性，第 498 页

ThemeProperties.InflectionColor 属性，第 498 页

ThemeProperties.RangeCategories 属性，第 491 页

ThemeProperties.InflectionRange 属性

目的

该属性是一个短精度值，应用于 Ranged 主题。控制哪一范围取 InflectionColor 指定的颜色。InflectionRange 上下的范围将为分别为 InflectionColor 和顶部与底部范围颜色的混色。

注解

此属性只在 ThemeProperties.InflectRanges 为 True 时适用。默认值为 2 （第二个范围）。

另请参阅

ThemeProperties.InflectionRange 属性，第 498 页

ThemeProperties.InflectionColor 属性，第 498 页

ThemeProperties.InflectionColor 属性

目的

该属性是一个 **OLE_COLOR** 值，应用于 ThemeProperties.InflectRanges 为真的 Ranged 主题。表示弯曲范围的颜色。弯曲的范围在 ThemeProperties.InflectionRange 属性中指定。默认值为 miColorWhite。

另请参阅

ThemeProperties.InflectRanges 属性，第 498 页

ThemeProperties.ColorMethod 属性

目的

该属性是一个短精度值，应用于 Ranged 主题。该属性应该设置为 **ColorSpreadingMethodConstants** 之一。指定用于插入到顶部和底部范围色之间的方法，用于取中间范围颜色。只在 ThemeProperties.SpreadBy 设置为 miSpreadByColor 时有效。

另请参阅

ThemeProperties.SpreadBy 属性，第 492 页

ThemeProperties.ApplyAttribute 属性

目的

这是一个短精度属性，应该设置为 **ApplyAttributeConstants** 之一。适用于 Ranged 或 Individual Value 主题。控制将哪一主题类别的样式属性实际应用于主题。可用于有选择性地指定主题类别的颜色或大小，不影响其他对象的外观。默认值为 miApplyAttributeAll。

另请参阅

ThemeProperties.RangeCategories 属性，第 491 页

RangeCategory 对象和 RangeCategories 集合，第 434 页

ThemeProperties.IndividualValueCategories 属性，第 491 页

IndividualValueCategory 对象和 IndividualValueCategories 集合，第 326 页

Title 对象

每个地图都包含一个标题 （**Map.Title** 属性），其中存储标题、样式和位置。

对象属性

- **Title.Border 属性**
- **Title.Caption 属性**
- **Title.Editable 属性**
- **Title.Position 属性**
- **Title.TextStyle 属性**
- **Title.Visible 属性**
- **Title.X 属性**
- **Title.Y 属性**

Title.Border 属性

目的

控制是否绘制标题边界。该属性是一个布尔值，默认值为 True。

Title.Caption 属性

目的

包含 **Title** 的文本。这是字符串值。这是 Title 对象的默认属性。

Title.Editable 属性

目的

控制标题是否可由最终用户编辑。通过此属性，我们可表达用户是否可以单击、移动、调整大小或更改其文本。该属性是一个布尔值，默认值为 True。

Title.Position 属性

目的

包含表示文本如何根据 X,Y 坐标绘制的值。取 **PositionConstants** 值，默认为 miPositionCC。

Title.TextStyle 属性

目的

包含标题样式的 Style 对象。

Title.Visible 属性

目的

控制标题是否可见。该属性是一个布尔值，默认值为 True。

Title.X 属性

目的

包含标题的 X 坐标。该属性是一个浮点值，表示屏幕坐标。标题位置由 X,Y 坐标和 Position 属性一起控制，后者控制标题和 X,Y 坐标值的相对位置。

Title.Y 属性

目的

包含标题的 Y 坐标。该属性是一个浮点值，表示屏幕坐标。标题位置由 X,Y 坐标和 Position 属性一起控制，后者控制标题和 X,Y 坐标值的相对位置。

Variable 对象和 Variables 集合

MapX 在表达式中支持变量替代。 Variable 对象和 Variables 集合是 MapX 在表达式中混合属性和地理标准的关键要素。通过将 Variables Collection （即一组 Variable 对象）作为可选参数传递给 **Layer.Search** 方法，表达式即可包含对 MapX 应用程序中的任意变量名的引用。

对象属性

- **Variable.Name 属性**
- **Variable.Value 属性**

集合属性

- **Variables.Count 属性**
- **Variables.Item 方法**

集合方法

- **Variables.Add 方法**
- **Variables.Clone 方法**
- **Variables.Remove 方法**
- **Variables.RemoveAll 方法**

注：默认属性为 Variables.Item，并且不可创建 Variable 对象。此外，对于长度和字符集没有限制。

注解

变量是一个绑定到可变对象的名称。当表达式读取无法解析的标识符时，它将尝试在 Variables 集合中进行解析。如果程序员在 Variables 集合的问题中添加了 Variable 集合，则将会执行表达式。变量可以为任意类型 （字符串、数字等）

注：在将图元用作变量时，如果搜索的 Feature 对象和图层具有相同坐标系，则可以获取最佳性能。这样可以避免快速多次的坐标转换。

Dataset 名称和 Dataset 字段名优先级高于变量名。

Variables 集合是可以创建的。

请注意对于 Feature 对象，在调用 Variables.Add 期间创建了图元的复本。如果在此后修改相应图元，则变量不会体现所做修改。您需要通过 Variables.Add 方法或 Variable.Value 属性来更新 Variable 的值。

Variable.Name 属性

目的

此只读属性指定置于变量集合中的变量的名称。变量名在集合中不区分大小写。

另请参阅

MapX 字段命名规范，第 507 页

Variable.Value 属性

目的

此只读属性指定置于 Variables 集合中的变量值。

Variables.Count 属性

目的

此只读属性包含用于 Variables 集合中的变量数量的整数值。

Variables.Add 方法

目的

此方法向 Variables 集合添加 Variable 变量，并且是 Variables 集合 的默认属性 / 方法。

语法

[Variables collection=]OBJECT.**Add**(Name, Value)

部分	说明
OBJECT	表示 Variable 对象。
Name	用于要添加的变量的名称的字符串值。
值	**Variant**：要添加的变量的值。（参阅 “注释”）。

注解

Variables.Add 方法向集合添加新的变量。如果同名变量已经位于集合之中，则它将使用新的变量值更新现有变量

变量可以为任意类型，例如数字、字符串、日期等）此外，还可以是 Idispatch，其中的对象是 MapX Feature 对象。

另请参阅

Feature 对象和 Features 集合，第 271 页

MapX 字段命名规范，第 507 页

Idispatch 表，第 633 页

Variables.Clone 方法

目的

此方法选择 Variables 集合中的指定对象，返回带有源自指定集合的变量复本的新 Variables 集合。

语法

[Variable object=]OBJECT.**Clone**

部分	说明
OBJECT	表示 Variables 集合。

Variables.Item 方法

目的

此方法从 Variables 集合特定的 Variable 对象。这是变量集合的默认方法。

语法

[Variable object=]OBJECT.**Item**(Index)

部分	说明
OBJECT	表示 Variables 集合。
Index	**Variant**：指定从 Variables 集合中返回哪一项。可以为基于 1 的索引或变量名。

Variables.Remove 方法

目的

此方法从 Variables 集合移除指定的 Variable 对象。

语法

OBJECT.**Remove**(Index)

部分	说明
OBJECT	表示 Variables 集合。
Index	**Variant**：指定从 Variables 集合中移除哪一项。可以为基于 1 的索引或变量名。

Variables.RemoveAll 方法

目的

此方法从 Variables 集合移除所有 Variable 对象。尽管为空，但在向其添加其他 Variable 对象之前，Variables 集合将不受影响。

语法

[Variable object=]OBJECT.**Clone**

部分	说明
OBJECT	表示 Variables 集合。

附录 A: MapX 字段命名规范

有效字段名称的构成

- 该名称必须不得以数字 (0-9) 或 # 字符开始。
- 名称只能包含以下字符：
  - A-Z 和 a-z
  - 0-9
  - # （磅符号）
  - _ （下划线）
  - ~（波浪线）
- 该名称可以包含以下字符，但是只能应用于最后位置：
  - $
  - @
  - %
  - !
  - &
- 名称的长度必须小于 32 个字符
- 不允许使用以下名称
  - OBJ
  - OBJECT
  - ROWID
  - COL***n***，其中 ***n*** 是一个数字 （例如 COL1 和 COL42 无效但 COL1x 有效）

何处实施有效的字段名称

目前 MapX 只在创建新的 TAB 文件时验证字段名 （通过指定 Layers.Add， LayerInfo.Type 的值为 miLayerInfoTypeNewTable）。按照以上规则指定的字段名均为有效，但如果名称长度大于 31 个字符，则将在创建 TAB 文件 （和图层）时截短为 31 个字符，不会显示错误消息。

在添加数据集、基于表达式添加字段 (Datasets.AddField)、创建新的临时图层 （miLayerInfoTypeTemp 为 LayerInfo.Type 值的 Layers.Add）以及创建变量 (Variables.Add) 时可能使用了无效的字段名。这些字段名可能会在 MapX 中导致意外行为。

注：强烈建议 MapX 应用程序只使用有效的字段名来用于数据集字段名和变量名。

使用无效的字段名可能在以下方面导致意外行为：

Layer.Search ：必须对传入 Layer.Search 的表达式进行分析。包含空格、数值运算符等的字段名将会被错误解释，并且将导致错误或意外结果。

主题 - 在添加主题时，可以通过名称或索引指定字段或字段集合。如果指定一个字段名，但这个字段名以数字开头，则 MapX 会将该字段名错误解释为字段索引。该主题将会在具有该索引 （如果具有一个索引）的字段上创建，或者引发错误。例如，名为 “1990”的字段无效，但它可以存在。用 “1990”为字段调用 Themes.Add 将很可能引发一个异常。

Fields.Item 或 Fields("name") - 可以在字段集合中通过名称指定一个字段。错误命名的字段可能导致与上述 “主题”下提及原因相同的意外结果。

Dataset.AddField - 对于新字段的名称，将接受无效字段名。但是，如果表达式使用名称无效的其它字段的名称，则 AddField 方法或者为这一新字段引发一个异常，或者产生意外结果 （类似于 Layer.Search）。

变量名称 (Variables.Add) - 变量名称应遵守与字段相同的命名规则，因为它们都在表达式中使用。

附录 B: 调试器定制数据集支持

即使无法通过标准的 MapX 数据绑定技术访问数据源，您也可以创建自己的定制数据集支持。 MapX 3.0 和更高版本提供了运行时可扩展的体系结构，允许您通过基于 COM 的数据集接口插入定制数据集类型。

创建定制数据集支持是一个高级主题。这一讨论假定您已经对创建 COM 对象比较熟悉。

静态数据集对象概览

静态数据集对象是一个 COM 对象， MapX 可以使用其来从任意数据源检索数据。数据取回后在数据源绑定时聚集 （通过调用 MapX 的 Datasets.Add(…) 方法），然后通过 MapX 缓存。源数据的更改在强制刷新 MapX 的缓存的数据时 （通过调用 MapX DataSet.Refresh(…) 方法）才会体现在地图中。以此方式检索和缓存数据所需的所有行为抽象为三个接口：IMMapXDataSet、 IMMapXStaticDataSet 和 IMMapXColumnInfoContainer。静态 DataSet 对象必须实施所有这三个接口 （注意 IMMapXStaticDataSet 即派生自 IMMapXDataSet）。请参阅以下内容获取有关各个接口的约定封装的详细信息。

特殊注册

在注册之后，静态 DataSet 对象服务器必须在注册表中注册特殊的条目，以便 MapX 知道何时来使用它：

添加键：

```
HKEY_LOCAL_MACHINE\SOFTWARE\MapInfo\MapX\DatasetEngines\<DatasetID>
```

添加字符串值：

```
HKEY_LOCAL_MACHINE\SOFTWARE\MapInfo\MapX\DatasetEngines\<DatasetID>
\DE_CLSID =
```

静态 Dataset 对象的 CLSID （采用 {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx} 的形式）。

其中 <DatasetID> 是一个用于静态 DataSet 对象的唯一数字标识符。当 Map 的 DataSet.Add(…) 方法的 Type 参数为此值时，将请求您的服务器提供静态 DataSet 对象，以便 MapX 最终借助于其检索数据。 MapInfo 保留 0 到 1000 的值，这些值不得使用。

注：请勿忘记在注销服务器时删除这些注册表项条目！

接口

IMMapXDataSet 接口

```
[
   uuid(96e0f395-caec-11d0-9d99-00aa00a478cb),
   helpstring("IMMapXDataset Interface"),
   version(1.0)
]
interface IMMapXDataset : IUnknown
{
   HRESULT Init(
      [in] short sType,
      [in] VARIANT* pvSourceData,
      [in] VARIANT* pvFields );
   HRESULT GetSample(
      [in] long lColNum,
```

```
[in] long lNumSampleValuesRequested,
[in] VARTYPE vtRequested,
[out] VARIANT * pvarData,
[out] long * pNumRecordsFetched);
```

IMMapXDataSet::Init

HRESULT Init ();

[in]	short	sType	请求的数据集类型 （DataSets.Add 方法的 Type 参数）。 MapX 已经将您的服务器确定为此类数据集的服务器 （通过您在注册 COM 服务器时注册的注册表条目）。
[in]	VARIANT *	pvSourceData	源数据的引用。（DataSets.Add 方法的 SourceData 参数）。例如，在绑定类型为 miDataSetODBC 时此参数包含 ODBCQueryInfo 对象，绑定类型是 miDataSetDAO 时为 IDAORecordset 指针。
[in]	VARIANT *	pvFields	包含 VARIANTS 数组。每个 VARIANT 从客户端程序员要绑定到匹配的图层源数据确定 （通过名称或索引）行。如果空白 (pvFields->vt = =VT_EMPTY)，所有来自源数据的列均应绑定。

说明

初始化静态 DataSet 对象这一方法将在静态 DataSet 对象创建之后立即调用。它为静态 DataSet 对象提供了一个建立数据源连接的机会，并确定源数据中哪些列将绑定到该地图。静态 DataSet 对象应该在从此调用返回之前，构建内部的 ColumnInfo 对象集合 （实施 IMMapXColumnInfo 的对象）。对于每个绑定列，该集合必须包括一个初始化的 ColumnInfo 对象。此集合将通过 IMMapXColumnInfoContainer 接口向 MapX 开放。

返回值

成功时将返回 S_OK。

如果内存用尽，则将会返回 E_OUTOFMEMORY。

出现任何其他失败时将返回 E_FAIL。

IMMapXDataset::GetSample

HRESULT GetSample();

[in]	Long	lColNum	用于获取示例数据的源数据列。
[in]	Long	lNumSamplesRequested	请求的示例数据值的数量。
[in]	VARTYPE	vtRequested	返回数据的首选类型。
[out]	VARIANT *	pvData	接收示例数据的 VARIANT。返回 GetSample 时，此 VARIANT 应该包含请求类型的数组。MapX 将尝试将转换不同类型的数据，但是这样做的效率明显较低。
[out]	Long *	plNumSamplesValuesFetched	pvData 中的示例数据值的数量。

说明

用于从给定源数据列收集数据示例。注意返回的变量 (pvData) 必须包含**数组**，即使该数组只包含单一数据值。

返回值

成功时将返回 S_OK。

如果内存用尽，则将会返回 E_OUTOFMEMORY。

出现任何其他失败时将返回 E_FAIL。

IMMapXStaticDataset 接口

```
[
   uuid(2e6d4cc0-d132-11d0-9da0-00aa00a478cb),
   helpstring("IMMapXStaticDataset Interface"),
   version(1.0)
]
interface IMMapXStaticDataset : IMMapXDataset
{
   HRESULT FetchData(
      [in] long lColNum,
      [in] long lIdxRow,
      [in] VARTYPE vtRequested,
```

```
            [out] VARIANT* pvarData,
            [out] BOOL * pbNoMoreData);
        HRESULT BeginFetch();
        HRESULT EndFetch();
    }
```

IMMapXStaticDataset::FetchData

HRESULT FetchData();

[in]	Long	lColNum	用于取数据的源数据列。
[in]	Long	lIdxRow	用于取数据的源数据行。
[in]	vtRequested	vtRequested	返回数据的首选类型。
[out]	VARIANT *	pvarData	存放取回数据的 VARIANT。应该为 vtRequested 确定的类型。 MapX 将尝试将转换不同类型的数据，但是这样做的效率明显较低。
[out]	BOOL *	pbNoMoreData	表示取操作已经超出了可用源数据结尾的标志。如果源数据为 10 行，则在 MapX 尝试取第 11 行数据时设置此标志。

说明

从数据源取一个数据单元。

返回值

成功时将返回 S_OK。

如果内存用尽，则将会返回 E_OUTOFMEMORY。

出现任何其他失败时将返回 E_FAIL。

IMMapXStaticDataset::BeginFetch

HRESULT BeginFetch();

说明

表示取序列即将开始。可用于分配 / 设置取数据所需的资源。

返回值

成功时将返回 S_OK。

如果内存用尽，则将会返回 E_OUTOFMEMORY。

出现任何其他失败时将返回 E_FAIL。

IMMapXStaticDataset::EndFetch

HRESULT EndFetch();

说明

表示取序列已经结束。可用于取数据的空闲资源。

返回值

成功时将返回 S_OK。

如果内存用尽，则将会返回 E_OUTOFMEMORY。

出现任何其他失败时将返回 E_FAIL。

IMMapXColumnInfoContainer 接口

```
[
    uuid(1e584f00-d2a5-11d0-9da3-00aa00a478cb),
    helpstring("IMMapXColumnInfoContainer Interface"),
    version(1.0
]
interface IMMapXColumnInfoContainer : IUnknown
{
    HRESULT GetColumnInfoByName(
        [in] BSTR bstrColumnName,
        [out] IMMapXColumnInfo** ppIMMapXColumnInfo);
    HRESULT GetColumnInfoByIndex(
        [in] long lIndex,
        [out] IMMapXColumnInfo** ppIMMapXColumnInfo);
    HRESULT GetColumnInfoEnumerator(
        [out] IMEnumMapXColumnInfo** ppIMEnumMapXColumnInfo);
}
```

IMMapXColumnInfoContainer::GetColumnInfoByName

HRESULT GetColumnInfoByName();

[in]	BSTR	bstrColumnName	源数据列的名称。

[out]	IMMapXColumnInfo **	ppIMMapXColumnInfo	由 bstrColumnName 标识的 ColumnInfo 对象

说明

用于按照列名检索 ColumnInfo 对象。

返回值

成功时将返回 S_OK。

如果内存用尽，则将会返回 E_OUTOFMEMORY。

出现任何其他失败时将返回 E_FAIL。

IMMapXColumnInfoContainer::GetColumnInfoByIndex

HRESULT GetColumnInfoByIndex ();

[in]	Long	lIndex	源数据列的索引。
[out]	IMMapXColumnInfo **	ppIMMapXColumnInfo	由 lIndex 标识的 ColumnInfo 对象

说明

用于按照列号检索 ColumnInfo 对象。

返回值

成功时将返回 S_OK。

如果内存用尽，则将会返回 E_OUTOFMEMORY。

出现任何其他失败时将返回 E_FAIL。

IMMapXColumnInfoContainer::GetColumnInfoEnumerator

HRESULT GetColumnEnumerator( );

[out]	IMEnumMapXColumnInfo **	ppIMEnumMapXColumnInfo	计数器

说明

用于检索 ColumnInfo 计数器对象 （实施 IMMapXEnumColumnInfo 的对象）。此对象可用于枚举包含在数据集中的 ColumnInfo 对象集合。

返回值

成功时将返回 S_OK。

如果内存用尽，则将会返回 E_OUTOFMEMORY。

出现任何其他失败时将返回 E_FAIL。

ColumnInfo 计数器对象概览

ColumnInfo 计数器对象可用于模拟包含在 DataSet 中的 ColumnInfo 对象集合。它实施接口 IMEnumMapXColumnInfo （如下所述）

IMEnumMapXColumnInfo 接口

```
[
   uuid(96e0f396-caec-11d0-9d99-00aa00a478cb),
   helpstring("IMEnumColumnInfo Interface"),
   version(1.0)
]
interface IMEnumMapXColumnInfo : IUnknown
{
   HRESULT Next([in] ULONG celt, [out] IMMapXColumnInfo ** rgelt,
[out] ULONG * pceltFetched);
   HRESULT Skip([in] ULONG celt);
   HRESULT Reset();
   HRESULT Clone([out] IMEnumMapXColumnInfo ** ppIMMapXColumnInfo);
}
```

IMEnumMapXColumnInfo 是标准枚举接口的示例，它根据 IenumUnknown 建模，后者用于枚举 IMMapXColumnInfo 指针。有关 IenumUnknown 的完整文档，请参阅 OLE32 SDK。

ColumnInfo 对象概览

ColumnInfo 对象包含有关源数据的列的信息。它实施接口 IMMapXColumnInfo （如下所述）

IMMapXColumnInfo 接口

```
[
   uuid(96e0f394-caec-11d0-9d99-00aa00a478cb),
   helpstring("IMMapXColumnInfo Interface"),
   version(1.0)
]
```

```
interface IMMapXColumnInfo : IUnknown
{
    HRESULT Init(
        [in] BSTR bstrName,
        [in] VARTYPE vt,
        [in] long lColNum);
    HRESULT GetName([out] BSTR *pbstrName);
    HRESULT GetDataType([out] VARTYPE *pvt);
    HRESULT GetColumnNumber([out] long *plColNum);

}
```

IMMapXColumnInfo::Init

HRESULT Init(

[in]	BSTR	bstrName	源数据列的名称。
[in]	VARTYPE	vt	源数据列的类型。
[out]	long	lColNum	源数据列的索引。);

说明

初始化 ColumnInfo 对象。 MapX 绝不会调用此方法，它取决于 DataSet 对象在其创建之后初始化 ColumnInfo 对象。注意 MapX 当前只支持数字和字符串数据类型。

返回值

成功时将返回 S_OK。

如果内存用尽，则将会返回 E_OUTOFMEMORY。

出现任何其他失败时将返回 E_FAIL。

IMMapXColumnInfo::GetName

HRESULT GetName(

[out]	BSTR *	pbstrName
源数据列的名称。	);	

说明

检索源数据列的名称。

返回值

成功时将返回 S_OK。

如果内存用尽，则将会返回 E_OUTOFMEMORY。

出现任何其他失败时将返回 E_FAIL。

IMMapXColumnInfo::GetDataType

HRESULT GetDataType(

[out]VARTYPE *pvtThe type of the source data column.);

说明

检索源数据列的数据类型。注意 MapX 当前只支持数字和字符串数据类型。

返回值

成功时将返回 S_OK。

如果内存用尽，则将会返回 E_OUTOFMEMORY。

出现任何其他失败时将返回 E_FAIL。

IMMapXColumnInfo::GetColumnNumber

HRESULT GetColumnNumber(

[out]long *pColNumThe column number of the source data column.);

说明

检索源数据列的列号。

返回值

成功时将返回 S_OK。

如果内存用尽，则将会返回 E_OUTOFMEMORY。

出现任何其他失败时将返回 E_FAIL。

附录 C: 术语词汇表

ASCII

即美国标准信息交换代码的缩写。 ASCII 是在大多数微型计算机、计算机终端和打印机中使用的标准代码，它将字符表示为数字。其中不仅包括可打印字符，还包括表示回车、退格等控制代码。

Geocode

将 X 和 Y 坐标分配给表或数据库中的记录的过程，以便记录可以显示为地图上的对象。

HTML 页

请参阅 “静态页”。

Internet 服务器应用程序编程接口 (Internet Server Application Programming Interface, ISAPI)

用于 Microsoft 的 Internet Information Server (IIS) 的 API。 ISAPI 提供和 CGI 大致相同的函数，但是没有创建用于每个请求的新进程的开支。此外， ISAPI 提供了错误处理和安全性的函数。使用 ISAPI 的不利之处在于该应用程序只能在 Windows NT/IIS 系统上运行，并且该应用程序如果编写得不够好，将会导致 IIS 崩溃。

Netscape 服务器应用程序编程接口 (Netscape Server Application Programming Interface, NSAPI)

用于 Netscape 服务器的 API。 NSAPI 提供和 CGI 大致相同的函数，但是没有创建用于每个请求的新进程的开支。此外， NSAPI 提供了错误处理和安全性的函数。使用 NSAPI 的不利之处在于该应用程序只能在运行 Netscape 服务器的系统上运行，并且该应用程序如果编写得不够好，将会导致 Netscape 服务器崩溃。

ODBC 驱动程序

ODBC 驱动程序是一个动态链接库 (.DLL) 文件， MapInfo 使用该文件连接 SQL 数据库。每种类型的 SQL 数据库均需要不同的 ODBC 驱动程序。

Web 服务器

运行超文本传输协议和 Web 服务器软件的计算机系统。Web 服务器从 Web 用户的浏览器接收基于 URL 的 HTTP 请求，并将 HTML 页面发送回浏览器。 Web 服务器可以管理一个或多个 Web 站点。例如，一个商用服务器，通常具有多个 Web 站点。提供动态页服务的 Web 服务器必须运行 HAHTsite Application Server 的实例。

Web 浏览器

客户端的软件，从 Web 服务器请求 Web 页面 （通过发送 URL），向用户显示为超文本、图片和其他资源 （如音频和视频）的页面。

Web 页面

由 Web 浏览器显示的 HTML 文件的通用术语。 HAHTsite 可以创建静态和动态 Web 页面。请参阅 “静态页”和 “动态页”。

Web 站点

描述驻留在 Web 服务器上的逻辑相关 Web 页面的集合的通用术语。 HAHTsite Web Application 是由 HAHTsite IDE 创建的 Web 站点，并且由 HAHTsite Application Server 运行。

WWW 链接

可以插入到网页上的完整 URL。 HAHTsite 只将 WWW 链接的字符串传递到项目页面。您需要确保链接的字符串 (URL) 有效。

边界区域

在 GIS 中，边界是地图上由边框封闭的区域。威尔士、马尼托巴湖和阿根廷在地图上都将表示为边界。请注意单一边界可以包含多个多边形。因此，印度尼西亚是一个单一边界，但其中报行多个多边形。

超文本标记语言 (Hypertext Markup Language, HTML)

一种纯文本 (ASCII) 语言，开发人员用于创建 Web 页面，可以在不同的计算平台上使用不同的 Web 浏览器显示。 HTML 使用标记指定文档不同部分的结构。 HTML 支持链接 （使用 URL）指向其他 Web 文档和文件。 HAHTsite IDE 支持 HTML 2.0 和 HTML 3.2 标准草案中的众多新功能。

超文本传输协议 (Hypertext Transfer Protocol, HTTP)

在 Web 客户端和 Web 服务器之间基于消息的网络接口。 HTTP 在 TCP/IP 的基础之上运行。

超文本链接

Web 页面中突出显示的文本，链接另一个 Web 页面 （Web 上的其他地方）或同一 Web 页面的不同部分。从 Web 浏览器单击超文本链接 （通常有下划线文本表示），用户即可跳转至链接指向的位置。超文本链接通过 URL 实施 （请参阅 URL － 统一资源定位符）。

迪卡尔坐标系

使用并未绑定到任何现实世界系统的 x 和 y 坐标的坐标系。大部分 CAD 绘图均使用此方法的注册对象（例如球轴承总成图、建筑平面图）。如果图形使用迪卡尔坐标，则图形一角的坐标可能为 (0, 0)。

迪卡尔坐标

采用平面上的 x 和 y 值的几何对象常规表示。

地产产权

用于从地理上定义给定地区内的地产产权或土地所有权的地图集。税收地图就是一个地产产权地图的示例。土地注册、评估角色和税收地图包括地产产权。

地理信息系统 (Geographic Information System, GIS)

一套有组织的计算机硬件和软件集合，设计用于有效创建、控制、分析和显示各种类型的地理或空间参考数据。 GIS 允许在其他情况下难于进行的复杂空间操作。

非地球地图

其中的对象没有显式引用地球表面地点的地图。平面图是其典型的示例。

服务器端图像地图 (SSIM)

驻留在 Web 服务器上的图像地图文件，由 CGI-BIN 脚本处理。此文件和使用图像地图的 HTML 页面有所区别。当用户单击地图上的面积时，浏览器将点击的坐标发送到 Web 服务器。 Web 服务器处理相应信息，将用户的点击转换为 URL。与客户端图像地图 (CSIM) 相对

公共网关接口 (Common Gateway Interface, CGI)

Web 客户端向 Web 服务器上的程序 （CGI 程序）传递参数、运行程序和接收程序结果 （在 HTML 中）的方式。 CGI 程序可以使用任意在服务器上运行的编程语言编写 （例如 C++ 或 PERL）。按照常规，CGI 程序存储在服务器的 CGI-BIN 目录中。

绘图法

制图的艺术和科学。在 GIS 中，它还是数据的图形表示和可视解释。

集成开发环境 (Integrated Development Environment, IDE)

在一个统一的环境中具有所有必要开发函数的软件工具。换而言之，功能全面的软件工具套件，包括扩展语言，带有完整的一组内置函数、数据类型和控制结构、专业的调试器调试器并支持团队开发。

经度，纬度，小数度数

度数 （纬度和经度）是用于表示地球表面上的地点的坐标。经度或 X 坐标表示地点的东西向位置，其中位于本初子午线以西的任意地点具有负的 X 值。纬度或 Y 坐标表示地点的南北向位置，其中位于赤道以南的任意地点具有负的 Y 值。

静态页

标准的 HTML 兼容的 Web 页面。静态页面以 HTML 格式存储在 Web 服务器上，并且只在页面的新版本传输到服务器时更新。和动态页相比，静态页不由 HAHTsite Application Server 创建，静态页直接发送到用户的浏览器。开发人员可以将静态页导入到 HAHTsite 项目中。

经线

地图上的竖直线，从北极至南极，用于描述点的东西向位置。位置以本初子午线 （0 度）的东向度数表示 （至 -180.0 度）或西向度数 （至 +180.0 度）表示。经线最大限度地远离了赤道，并相交于两个极点，因此不会平行。

经线 / 纬线

MapInfo 用于表示地图上地理对象的默认坐标系。

客户端脚本

由浏览器执行的 JavaScript 或 VBScript 代码。客户端脚本可以是独立的语句、函数或事件处理程序。

客户端图像地图 (CSIM)

嵌入到 HTML 页面中的图形地图。该地图包含 MAP 标记，命名图像地图；以及一系列 AREA 标记，每个标记说明一个区域 （或热点）和该区域的 URL。当用户单击区域时，浏览器请求关联的 URL。与服务器端图像地图 (SSIM) 相对

框架

框架将 Web 浏览器分成若干个区域，其中每个区域均可以包含单独的可滚动页面。框架具有一个名称和一个源 URL，指定了在框架初次加载时，浏览器在其中显示的文档。请参阅 “框架页”。

框架页

HTML 页面类型之一，包含一组框架或框架集。框架集定义了页面中各个框架的位置和大小。

当用户使用指向框架页的链接时， Web 浏览器将先加载框架集，然后加载由框架集中每个框架的资源 URL 所引用的页面。请参阅 “框架”。

事件处理程序

事件处理程序可以使用 JavaScript 或 VBScript 编写，是页面对象的属性之一。例如，事件处理程序说明在用户单击按钮或选择列表框中文本时应该做些什么。 VBScript 和 JavaScript 均显式支持定义的事件处理程序 – 例如有关 “单击”和 “选择”的处理程序。此外，您可以定义替代显式事件处理程序的函数。类似函数称为隐含事件处理程序。

缩放图层

确定 “地图”窗口中图层可见的范围 （例如 0-3 英里， 2-5 英里等）的设置。

填充模式

用于填充封闭物体的设计和颜色方案。

统一资源定位符 (Uniform Resource Locator, URL)

超文本链接或图像地图的底层实现，包含 Web 页面地址或互联网上某处文件的地址。 URL 包含有关要使用的网络协议的信息 （通常是 HTTP）以及到该页或文件的路径。 http://www.haht.com/index.html 就是一个 URL 示例，它指向 HAHT Software 的 Web 站点的索引页。

投影

将地表特征的位置转换为二维平面 （例如纸制地图）位置的数学模型。由于地图要在平面上表示球形对象 （地球），因此所有投影都具有一定的角度扭曲。地图投影可以保留面积、距离、形状或方向，但是只有球体才可以保留全部这些属性。某些投影 （例如墨卡托）制作的地图非常适合于导航。某些投影 （例如朗伯等面积投影）制作的地图非常适合于可视化分析。

图层

MapInfo MapX 地图的基本构建模块。 MapInfo 地图通常包括若干分层图层 （例如在城镇或邮政编码边界的图层之上分层的街道数据图层）。当表出现在 “地图”窗口中时，它将在该 “地图”窗口中占用一个图层。通常每个图层相应于一个打开的表，此外还有特殊的 “装饰图层”，其中包含表示临时地图注释 （例如文本）的地图对象。

图像地图

定义图片的 “热点”位置、形状和大小的文件 （JPEG 或 GIF）。 URL 关联到每个热点，因此一个图片可以执行若干个不尽相同的操作。例如，图形的一部分可能允许用户向站点管理员发送邮件，而另一部分则可能允许下载文件，而第三个部分可能会将其带回到站点的主页。另请参阅客户端图像地图 (CSIM) 与服务器端图像地图 (SSIM).

网格

在地图上显示的水平 （纬线）和竖直 （经线）线构成的间距规则的网格 （例如每 5 度，或每 15 度）。用于建立参考框架。

纬线

地图上的水平线，从赤道处的 0 度增长至南 （-90.0 度）北 （+90.0 度）两极的 90 度。用于描述点的南北向位置，通常以度或小数度数在赤道上、下测量。

位图

显示为点或位的数组的屏幕图像。软件通常生成位图 （光栅）或面向对象 （矢量）的文件。

向导

一系列对话框，展示简单的表单、屏幕和提示，方便并简化开发人员操作。引导您创建新站点模板的向导就是向导的示例之一。

项目链接

指向项目中的页的引用。此外，项目链接可以引用页面上的书签。项目链接位于您正在使用的项目所在的本地。当您向另一页面拖放页的图标 （从项目窗口）时，即在创建项目链接。

像素

图像元素的简称。是计算机屏幕上可以显示的最小的点。如果屏幕的分辨率为 1,024 x 768，则屏幕从左到右显示 1,024 个像素，从上到下显示 768 个像素。屏幕上的每个字符、对象或线条均由一定数量的像素构成。

小数度数

用小数表示部分度数的方法。许多纸制地图均采用度、分、秒 （例如 40_30ι10I）来表示坐标，其中分和秒均是度数的分数。30 分等于半度，30 秒等于半分。但 MapInfo 以小数度数表示坐标（例如 72.558 度），其中部分度数表示为小数。因此，经度 40 度 30 分在 MapInfo 中表示为 40.5 度。

应用程序状态

请参阅 “状态”。

站点

项目发布到的目标服务器。

站点说明

说明项目发布到的目标服务器的项目项。站点说明，存储在 Sites 文件夹中，包括有关目标服务器的配置信息，以及有关如何和何处传输项目的静态或动态页的信息。

质心

通常是地图对象的中心。对于大多数地图对象，质心位于物体的中部 （在物体的南北延展方向的中心和东西延展方向的中心）。在某些情况下，质心不是中点，因为质心必须位于物体本身之上。例如对于新月形区域的物体，物体的中点实际上可能位于该区域限定的范围之外，但是质心则始终位于区域之中。

主页

通常是 Web 应用程序的起点，主页是用户访问 Web 站点时默认显示的页面。

在 theHAHTsite IDE 之内，Web 文件夹及其每个子文件夹可以具有主页。Web 文件夹的主页即该应用程序的主页。

子午线

从北极到南极的线或线的一部分。是一条经线。

坐标

迪卡尔坐标系中的 (x,y) 位置，或是地球坐标系中的经度和纬度位置。坐标表示地图上相对于其他位置的位置。地球坐标系可使用赤道和格林威治本初子午线作为固定的参考点。平面坐标系根据固定参考位置的距离描述二维坐标 x,y 位置，通常位于第一象限，以令所有坐标均为正数。

坐标系

坐标系用于创建几何对象的数字表述。几何对象中的每一点均由一对数字表示。这些数字是该点的坐标。在绘图法中，坐标系和投影密切相关。您通过为投影参数提供特定的值创建坐标系。

附录 D: 使用 Geodictionary Manager

Geodictionary 是包含与 MapInfo 表有关的注册信息的文件， MapX 可以在自动数据绑定期间匹配这些表。在 Geodictionary 中只应注册可以或将要与之进行匹配的 MapInfo 表。

注：不需要注册应用程序在 Geodictionary 中使用的每一 .tab 文件，实际上，在注册不需要的文件时会造成一些系统开销。

Geodictionary Manager 应用程序用于操作 Geodictionary。可以通过图形用户界面或命令行参数运行 Geodictionary Manager 可执行文件 (GeoDictionaryManager50.exe)。这些命令行选项用于从安装程序调用 GeoDictionaryManager40.exe，从而注册 MapX 应用程序可能需要与 Geodictionary 进行匹配的表。

背景信息

以前的 MapX 版本使用为 Microsoft Map 开发的 MapInfo Data Installer 来管理 Geodictionary。该 Data Installer 过去还管理 geoset （可以一起显示的 MapInfo 表的组合），这已不适合 **Geodictionary Manager** 并会导致混乱。现在应使用 **Geoset Manager** 应用程序来创建和管理 geoset。**Geodictionary Manager** 将永远不会修改 geoset，并且也不会删除用户磁盘上的文件。

在 MapX 5.0 （或更高版本）中，永久的 Geodictionary 文件不再是 MapX 运行所必需的。

注：已对命令行选项进行了更改，要求包含空格的文件名或路径名需要用双引号引起来。因此，以前可以正常使用的某些命令行现在可能无法使用了。

Geodictionary 文件

Geodictionary 包含在通常称作 geodict.dct 的二进制格式的文件中。该 geodictionary 文件可以使用注册表项 HKEY_LOCAL_MACHINE\SOFTWARE\MapInfo\MapX 5.0\GeoDictionary 定位或禁用。对于该注册表项有三种可能的有效值：

- **可以包含 Geodictionary 文件的完整文件规格** （例如 “C:\Program Files\MapInfo MapX\Maps\geodict.dct”）。结果就是数据目录设置为指向该注册表项所指定的 Geodictionary 文件的路径。内存中 geodictionary 是从 Geodictionary 文件初始化的，并且 Geodictionary 中包含的图层也可以自动与其进行匹配。添加到地图中的图层被添加到内存中的 geodictionary，这样就可以自动与其相匹配。
- **可以只包含路径说明** （例如 “C:\Program Files\MapInfo MapX\Maps\”）。数据目录被设置为该注册表项指定的路径。内存中 geodictionary 被初始化为空。添加到地图中的图层仍被添加到内存中的 geodictionary，这样就可以自动与其相匹配。
- **完全不可以存在**。内存中 geodictionary 被初始化为空。添加到地图中的图层仍被添加到内存中的 geodictionary，这样就可以自动匹配。

注：MapX 的卸载将留下 geodict.dct。这是预计的情况，因为 Geodictionary 文件可能在安装前已存在 （来自早期的 MapX 版本），并且可由其它应用程序共享。

多个 MapX 应用程序可以共享同一 Geodictionary 文件，也可以单独使用它们。为使 MapX 应用程序可以使用其自己的 Geodictionary 文件 （或者完全不使用任何 Geodictionary 文件），可以在设计阶段更改 **Map.GeoDictionary** 属性，以指向不同的注册表项。然后，在运行时初始化地图控件时， MapX 将查询以下注册表项：

HKEY_LOCAL_MACHINE\SOFTWARE\MapInfo\MapX 5.0\<Map.GeoDictionary 的值 >

此注册表项的值将在前面详细解释。 **Map.GeoDictionary** 的默认值是 “GeoDictionary”，因此 MapX 将使用该默认的注册表项。

Geodictionary Manager 始终读取并更新默认的 Geodictionary 文件 （该文件由注册表项 HKEY_LOCAL_MACHINE\SOFTWARE\MapInfo\MapX 5.0\GeoDictionary 引用）。如果您在用户界面中更改了 Geodictionary 文件的位置，则更新该注册表项以指向这一新位置。

Geodictionary 为每一注册的表 （mapinfo .tab 文件）包含一项。与每一注册表项一起存储的数据是：

- 引用该表或在图层控制对话框中显示的**用户友好的名称**。它在 .tab 文件中默认为 “Description”标记，如果不存在则为文件名。可以通过用户界面对其进行编辑。
- **索引字段的列表**。通过该用户界面，您可以选取哪些索引字段您希望 MapX 在自动数据绑定期间考虑将列与之相比较 （如果可能）。将忽略未选取的字段。在第一次注册一个表时，默认情况下最初选取所有索引的字段。
- **优化表名称**。某些表 （例如 US Counties）包含不唯一的索引列。在这一情况下，需要优化表来确定对数据的确切匹配。如果该表具有不唯一的索引列，将启用优化列组合框，并且用户将可以挑选优化的表。
- **Geoset 文件名列表**。在自动数据绑定期间选择一个表时，将该表的 geoset 列表传递给 ResolveDataBind 事件，以便 MapX 程序从中选择一个 geoset。默认情况下， MapX 将挑选第一个未处理的事件。注：已匹配的表始终装载，即使它不在装载的 geoset 中。此外，在 Geodictionary 中没有为某一表列出任何 geoset 并不是一个错误。在该情况下，只装载该表。最后，只能将与 Geodictionary 文件位于同一目录中的 geoset 文件添加到该列表中。

用户界面

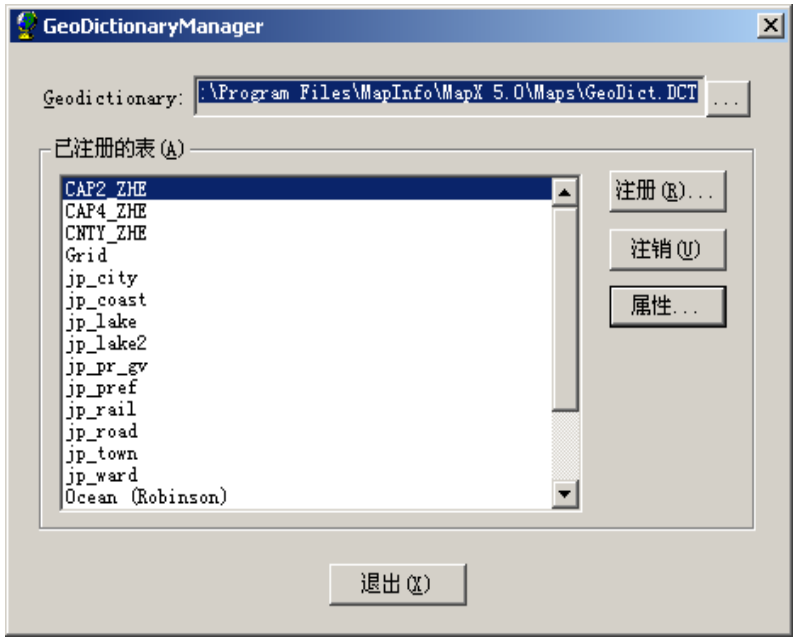
本节描述 Geodictionary Manager 的用户界面。

运行 Geodictionary Manager

要在您希望手动注册图层时运行 GeoDictionary Manager：

1. 开始 -> 程序 -> MapInfo MapX X.0
2. 从 Windows 的 “打开”对话框中选择 “Geoset”。
3. 在 “工具”菜单中，选择 “运行 GeoDictionary Manager”。

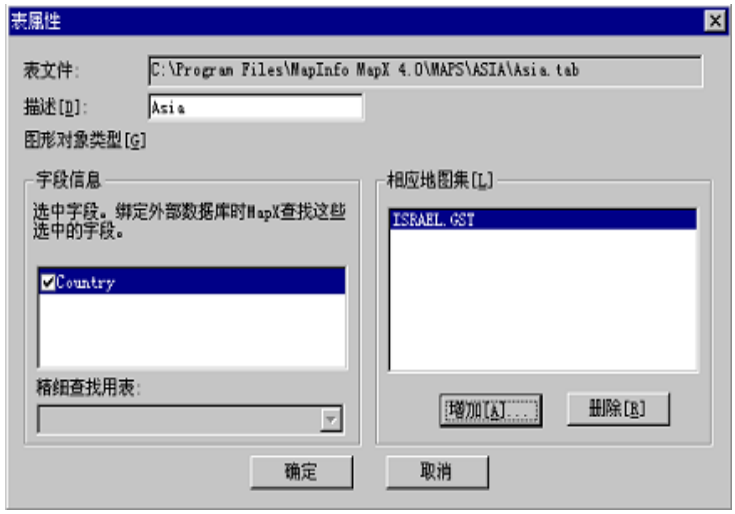
   将显示 “Geodictionary”对话框。



部分	说明
Geodictionary	Geodictionary 编辑框是只读的编辑框，包含指向所管理的 Geodictionary 的完整路径。这一信息可以通过在注册表中查找 MapX 项来获得。通过 Geodictionary 编辑框右侧的按钮，用户可以浏览要管理的其它 Geodictionary。浏览到其它 Geodictionary 将更改 MapX 的注册表项，以便所有随后的 MapX 会话都将使用新的字典。该按钮还可用于更改存储于注册表项 “HKEY_LOCAL_MACHINEOFTWAREapInfoap X.0earchPaths” 中的默认 MapX 搜索路径。
已注册的表	“已注册的表” 列表框包含在 Geodictionary 中注册的所有表的用户友好名称的列表。
注册	这将显示常见的文件打开对话框，而将 “文件类型” 组合框设置为 “MapInfo 表 (*.tab)”。将该表复制到数据目录 （包含 Geodictionary 的目录）中，或者将包含该表的目录添加到数据搜索路径中。在吸收了该表 （通过复制或通过将目录添加到搜索路径）后，将显示 “表属性” 对话框。用户可以通过在单击的同时按住 Ctrl/Shift 键来选择多个要注册的文件。如果选择了多个表，属性按钮将被禁用。请参见 **“表属性” 对话框**。

部分	说明
注销	从 Geodictionary 中删除选定的表。此 “注销”按钮将不会磁盘中删除文件。用户可以通过在单击的同时按住 Ctrl/Shift 键来选择多个要注销的文件。
属性	为选定的表显示 “表属性”对话框。请参见下面的 “表属性”对话框。

使用 “表属性”对话框为给定的表设置 Geodictionary 字段。



MapInfo 表

包含 MapInfo 表 （如果该表与 Geodictionary 位于同一目录中）的文件名或指向该文件的完整路径名 （如果它没有文件名）的只读的编辑框。

说明

提供用来更改表的用户友好名称的机制。该控件在 .TAB 文件中默认为 Description 标记；如果未找到该 Description 标记，则默认为文件名，但用户可以更改该文件名。请注意，在 Geodictionary Manager 中对说明的更改将只存储在 Geodictionary 中，并且将不会在该表中反映出来。这使得 Geodictionary Manager 可以轻松地使用只读数据，例如 CD-ROM 上的数据。

几何

“几何”静态文本控件显示在表中存储的实体的类型。

字段信息

包含表中索引列的列表。如果为给定列选取了该框，则在匹配过程中将搜索该字段。

优化表

某些表 （例如 US Counties）包含不唯一的索引列。在这一情况下，需要优化表来确定对数据的确切匹配。如果该表具有不唯一的索引列，将启用优化列组合框，并且用户将可以挑选优化的表。

候选 Geoset

可能准备进行装载的 geoset 文件名的列表 （如果在数据绑定期间已选择了 MapInfo 表）。可以列出零个或多个 geoset 文件名。在自动数据绑定期间选择该表时，将该表的 geoset 列表传递给 ResolveDataBind 事件，以便 MapX 程序从中选择一个 geoset。默认情况下， MapX 将挑选第一个未处理的事件。

请注意：

- 始终装载该匹配的表，即使它不在装载的 geoset 中。
- 在 Geodictionary 中没有为某一表列出任何 geoset 并不是一个错误。在该情况下，只装载该表。
- 只能将与 Geodictionary 文件位于同一目录中的 geoset 文件添加到该列表中。
- 可以添加或删除多个候选的 geoset

添加

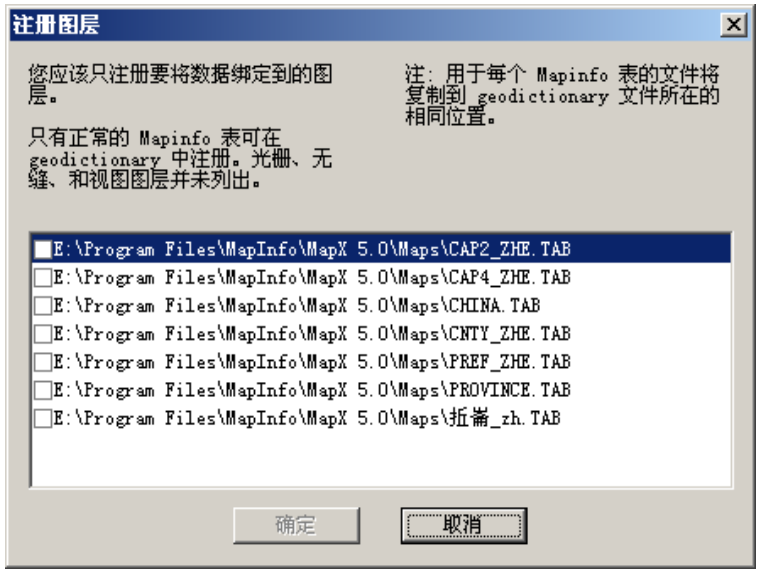
这将显示一个列表框，该列表框包含与 Geodictionary 文件位于同一目录中的 geoset 文件名的列表。选择一个文件名会将该文件名添加到候选 geoset 的列表中。

删除

从候选 geoset 的列表中删除选定的 geoset 文件名。

在 Geodictionary 中注册图层

如果希望 Geoset Manager 自动注册您的图层，则选择 “工具” > “注册图层”。将显示 “注册图层” 对话框。



选取您要注册的图层，然后单击 “确定”。

命令行选项

Geodictonary Manager 命令行提供向 Geodictionary 添加表或从 Geodictionary 删除表的机制。

geosetpath

指向包含表列表的 geoset 文件的路径。通常路径是必需的；仅指定文件名是错误的。

- 使用 **geosetpath** 选项向 Geodictionary 注册该 geoset 中所有有效的 .tab 文件。（geoset 中的 Raster、 Seamless 和 View 表将被忽略。）
- 使用 **geosetpath /remove** 选项从 Geodictionary 注销 geoset 中所有有效的标记文件。

tablepath

指向将添加到 Geodictionary 中的 MapInfo 表的路径 （该选项不具有任何参数）。

- 使用 **tablepath** 选项来向 Geodictionary 注册一个文件。
- 使用 **tablepath /remove** 选项从 Geodictionary 注销一个文件。

geosetfile

只是位于与 Geodictionary 文件相同的目录中的 geoset 文件的文件名。

- 使用 **tablepath /geoset=geosetfile** 选项向 Geodictionary 注册一个文件并将 geosetfile 添加到要匹配的候选 geoset 列表中。
- 使用 **tablepath /geoset=geosetfile /remove** 选项从其要匹配的候选 geoset 列表中删除 geosetfile。

commandfilepath 在文件的每一行包含有效的 migm30 命令行的文本文件的完整路径名。

- 使用 **/file=commandfilepath** 选项处理命令行的文本文件，在该文件的每一行上有一个命令行。不要在文件的每一行上包括 “migm30.exe”。
- 使用 **/file=commandfilepath /remove** 选项处理命令行的文本文件，在该文件的每一行上有一个命令行。 **/remove** 选项将在执行前添加到每一命令行上。不要在文件的每一行上包括 “migm30.exe”。

注：具有空格的路径或文件名必须用双引号引起来。如果可能， Geodictionary Manager 将无提示运行命令行。尝试多次注册一个文件并非错误。注销当前未注册的表并非错误。

附录 E: MapX 事件

普通事件

MapX 支持以下普通事件：

事件	OLE Dispatch ID	说明
Click	-600	用户按下并释放鼠标按键。
DblClick	-601	用户双击。
KeyDown	-602	用户按下一个键。对于构建鼠标活动修饰符非常实用。
KeyPress	-603	用户按下一个按键。
KeyUp	-604	用户释放一个按键。
MouseDown	-605	用户按下鼠标按键。
MouseMove	-606	用户移动鼠标。可用于显示 “状态栏”中的光标坐标。
MouseUp	-607	用户释放鼠标按键。
错误	-608	在异步活动中出现错误，例如重新绘制地图或者用户使用地图工具的时候。

定制事件

MapX 支持以下定制事件：

事件	Dispatch ID	调用 …
AddFeatureToolUsed 事件	16	在使用了 “添加功能”工具时
AnnotationAdded 事件	7	在添加注释时。
AnnotationChanged 事件	8	在添加、修改或删除注释时。
DataMismatch 事件	5	在数据绑定期间行不匹配。
DrawUserLayer 事件	10	在地图窗口需要更新和应用程序已经创建 UserDraw 图层时。
LabelChanged 事件	17	在要添加、选择、取消选择或移动标签时。
MapDraw 事件	15	在开始绘图之前触发一次，在绘图完成之后触发一次。
MapInitialized 事件	13	在地图初始化完成之后。

事件	Dispatch ID	调用 ...
MapViewChanged 事件	6	在地图的最小绑定矩形改变之后 （例如在放大或平移时）。
MouseWheel 事件	12	用户使用滚轮鼠标时。
PolyToolUsed 事件	11	用户使用 polytool 时 （例如绘制多义线）。
RequestData 事件	4	在处理未绑定的数据时从单元容器请求数据。
ResolveDataBind 事件	2	在自动数据绑定期间数据不确定。
ResolveDataBindEx 事件	14	当其具有和 **ResolveDataBind** 事件相同的函数，但是传递 **ResolveObjects** 集合，而不是字符串数组时。
SelectionChanged 事件	1	在选择改变时。
ThemeModifyRequested 事件	9	用户双击图例时。
ToolUsed 事件	3	在用户使用地图上的工具时。

取决于开发环境，您可能还需要使用其他事件。例如， Visual Basic 提供了以下事件支持：DragDrop、DragOver、 GotFocus 和 LostFocus。

AddFeatureToolUsed 事件

此事件在用户使用标准对象创建工具 （miAddPointTool、 miAddLineTool、 miAddPolylineTool 或 miAddRegionTool）时调用。此事件的行为和 PolyToolUsed 事件类似。

Dispatch ID = 16

语法

AddFeatureToolUsed （ToolNum, Flags, Feature, Shift, Ctrl, EnableDefault）

部分	说明
ToolNum	表示添加图元工具的工具数量。
Flags	**ToolFlagConstants** 值，表示用户使用工具是正在开始、正在进行中还是已经完成。
Feature	Feature 对象表示用户目前已经创建的图元。
Shift, Ctrl	表示修饰符键 (SHIFT, CTRL) 在事件触发时是否按下的布尔值， True 表示键已经按下。
EnableDefault	布尔值。此参数控制 MapX 是否允许工具的标准行为生效。默认值为 True。要避免工具的正常行为 （例如取消用户操作），可将此参数设置为 False。此参数只在用户结束工具使用时生效 (miToolEnd)。

注解

此事件在对象创建工具的每次使用时多次触发。

在开始使用工具时触发一次，在用户单击置于第一个图元节点上时。在此调用期间， Flags 变量设置为 miToolBegin (0)。

对于 “添加多义线”和 “添加区域”工具，每次用户单击添加另一节点时触发一次，并且还在用户每次通过按下 BACKSPACE 键删除节点时触发一次。在此调用期间， Flags 变量的值等于 miToolInProgress (3)。

在用户执行以下操作之一时，在结束使用工具时触发一次：

- 通过双击 （多义线和多边形工具）或释放鼠标 （线和点工具）结束图元的绘制，此时 Flags 变量将设置为 miToolEnd (1) ；或者
- 按下 ESC 键结束添加图元工具，此时 Flags 变量将设置为 miToolEndEscaped (2) ；或者
- 使用退格键删除所有节点，此时 Flags 变量将设置为 miToolEndEscaped (2)。

AnnotationAdded 事件

目的

此事件在添加、修改或删除注释时调用。它在注释更改时调用。程序员还有机会避免该操作。

Dispatch ID = 7

语法

AnnotationChanged 事件

目的

此事件在添加、修改或删除注释时调用。它在注释更改时调用。程序员有机会避免该操作。

Dispatch ID = 8

注：该事件只在用户交互做出更改时触发 （例如选择注释、移动注释、重新定义大小等），而非在注释中通过程序进行更改。例如，这不会触发 AnnotationChanged 事件：annotation.X = 100。尽管它正在移动注释。

语法

AnnotationChanged (changetype, EnableDefault)

部分	说明
changetype	指向注释对象的对象指针。

部分	说明
EnableDefault	布尔值。此参数控制 MapX 是否允许注释的标准行为生效。默认值为 True。如果程序员将 EnableDefault 设置为 FALSE，则 MapX 将不会做出任何操作来响应注释。

另请参阅

AnnotationChangedType 常数

Data 事件

目的

这一事件在数据绑定期间，行不匹配地图中的对象时调用。

在 MapX 了解数据集的 geocolumn 之后，它可以随后将其绑定到地图。但是， MapX 可能遇到不匹配地图中的对象的行。在默认情况下，它将会跳过。如果 **DataMismatch** 事件处理程序已经定义，我们将在行不匹配时向其发送事件。

Dispatch ID = 5

语法

DataMismatch (DatasetName, ByValRow, GeoColValue)

部分	说明
DatasetName	此字符串是不匹配的数据集的名称。
ByValrow	这是不匹配的常规数据源中的行号 （长）。
GeoColValue	这是不匹配数据集中的字段的字符串值。

DrawUserLayer 事件

目的

在应用程序使用 Layers 集合的 AddUserDrawLayer 方法创建 UserDraw 图层时，将在窗口需要更新时针对该应用程序触发事件：

Dispatch ID = 10

语法

DrawUserLayer (Layer, hOutputDC, hAttributeDC, RectFull, RectInvalid)

部分	说明
图层	Layer 对象：要绘制的图层。
hOutputDC	OLE_HANDLE：要拖动到的设备上下文。
hAttributeDC	OLE_HANDLE：要拖动到的设备上下文，通常和 hOutputDC 相同，在拖动到元文件设备上下文时除外。
RectFull	Rectangle 对象：完整窗口。
RectInvalid	Rectangle 对象：无效区域。

矩形为 **Rectangle** 类型的对象。矩形中的坐标是屏幕 （客户端）坐标。

Remarks

您还可以通过 Layers.Add 添加 UserDrawLayer。

LabelChanged 事件

MapX 在要添加 （通过普通标签工具）、选择、取消选择或移动 （通过普通标签工具）标签时触发此事件。 MapX 程序员有机会通过将 EnableDefault 的值设置为 False 来避免此操作。

注：此事件只在通过用户交互 （例如选择或移动标签）做出更改时触发。而非通过程序更改。例如 Label.AnchorX = -110 将不会触发该事件，即使它确实移动了标签。

语法

LabelChanged (ChangeType, ChangingLabels, *EnableDefault)

Dispatch ID = 17

部分	说明
changetype	整数：将要发生的更改类型 （miLabelMove、 miLabelAdd 或 miLabelSelectionChange）
ChangingLabels	Labels 对象：表示将要更改的 Label 对象的集合。此对象具有和 Layer 对象的 Labels 集合相同的接口 （例如它具有 Count 和 [Item] 属性）。

部分	说明
*EnableDefault	布尔值：控制是否允许 MapX 的默认行为生效。默认值为 True。如果将此设置在事件处理程序中设置为 false， MapX 将忽略要更改标签的用户交互。

MapDraw 事件

此事件在绘图之前调用一次，在绘图完成之后调用一次。此处有一个参数表明绘图是开始还是结束。

Dispatch ID = 15

语法

MapDraw (Flag)

部分	说明
Flag	miDrawBegin 或 miDrawEnd。

另请参阅

MapDrawConstants

MapInitialized 事件

目的

此事件在地图初始化完成之后立即调用。它还通知容器应用程序进行任何必要的运行时配置的最佳时间。

Dispatch ID = 13

语法

MapInitialized < 无参数 >

MapViewChanged 事件

目的

此事件在控件 （地图）上的距离更改或中点移动时调用。

Dispatch ID = 6

语法

MapViewChanged < 无参数 >

MouseWheel 事件

目的

智能鼠标在 Microsoft 鼠标上提供滚轮作为第三个 （中间）按钮。除了常规的鼠标按键消息之外，滚轮鼠标在转动时发送 MOUSEWHEEL 消息。

MouseWheel 在滚动时将缩放地图，在按下 CTRL 键的同时滚动滚轮将上下滚动地图，在按住中间按钮并远离鼠标点击点移动时 AutoScroll。滚动的速度和鼠标远离鼠标点击点的距离成正比。 MouseWheel 事件允许您覆盖此 MouseWheel 的内置行为。

请注意在 Windows 95 下的 Visual Basic 中没有用于 MouseWheel 的支持，但是借助于已经构建的第三方 ActiveX 控件，即可利用 MouseWheel 的优点。（即将另一控件置于您的表单之上，收集和发送 MOUSEWHEEL 消息。）

Dispatch ID = 12

语法

MouseWheel(long Flags, int zDelta, single X & Y, bool EnableDefault)

部分	说明
Flags	表示各种虚拟键是否已经按下。此参数可以采用以下值的任意组合： **MK_CONTROL**: 如果 CTRL 键按下则设置。 **MK_LBUTTON**：如果鼠标左键按下则设置。 **MK_MBUTTON**：如果鼠标中键按下则设置。 **MK_RBUTTON**：如果鼠标右键按下则设置。 **MK_SHIFT** 如果 SHIFT 键按下则设置。
zDelta	wParam 的高序词的值。. 表示滚轮旋转的距离，以 WHEEL_DELTA 的乘法或除法表示，即 120。正值表示该滚轮向前转动，远离用户；负值表示滚轮向后转动，朝向用户。
X	指定指针的 x 坐标，相对于屏幕的左上角。
Y	指定指针的 y 坐标，相对于屏幕的左上角。
EnableDefault	布尔值。此参数控制 MapX 是否允许鼠标滚轮的标准行为生效。默认值为 True。如果程序员将 EnableDefault 设置为 FALSE，则 MapX 将不会做出任何操作来响应 MouseWheel。

PolyToolUsed 事件

此事件在用户使用标准工具 miPolygonSelectTool (1010) 或定制 miToolTypePoly (4) 类型的工具时调用。

Dispatch ID = 11

语法

PolyToolUsed(ToolNum, Flags, Pts, Shift, Ctrl, EnableDefault)

部分	说明
ToolNum	表示定制工具号。
Flags	一个 **PolyToolFlagConstants** 值，表示用户是开始还是完成工具使用。
Pts	点的集合 （和 **Feature.Parts.Add 中使用的种类相同）。**
Shift, Ctrl	表示修饰符键 (SHIFT, CTRL) 在事件触发时是否按下的布尔值， True 表示键已经按下。对于标准工具 miPolygonSelectTool， MapX 只在用户结束工具时使用这些键 （例如 Flag 参数为 miPolyToolEnd 时）。
EnableDefault	布尔值。如果此事件由标准 miPolygonSelectTool 触发，则此参数控制 MapX 是否允许工具的标准行为生效。默认值为 True。要避免工具的正常行为 （例如取消用户操作），可将此参数设置为 False。此参数只在用户结束工具使用时生效 (miPolyToolEnd)。

Remarks

此事件在每次使用 PolyTool 时多次触发。

在用户单击置于第一个多义线 / 多边形节点上时 , 在开始使用工具时触发一次。在此调用期间， Flags 变量设置为 miPolyToolBegin (0)。

每次用户单击向多义线添加另一节点时触发一次，并且还在用户每次通过按下 BACKSPACE 键删除节点时触发一次。在此调用期间， Flags 变量的值等于 miPolyToolInProgress (3)。

在用户执行以下操作之一时，在结束使用工具时触发一次：

- 双击以结束多边形的绘制，此时 Flags 变量将设置为 miPolyToolEnd (1) ；

  - 或者 -

- 按下 ESC 键以结束多边形的绘制，此时 Flags 变量将设置为 miPolyToolEndEscaped (2) ；

  - 或者 -

- 使用退格键删除所有节点，此时 Flags 变量将设置为 miPolyToolEndEscaped (2) ；或者

RequestData 事件

目的

在处理未绑定的数据时从单元容器请求数据时调用此事件。

Dispatch ID = 4

语法

RequestData (row, column, value, done)

在没有更多数据传递到容器时设置完成标志。

ResolveDataBind 事件

目的

自动数据绑定将遇到不确定的情况。这些情况包括：在用户数据中检测到多个 geocolumn、检测到匹配 geocol 的多个图层、有多个 geoset 可用于已匹配的图层。此时 MapX 将会采用第一个对象，除非已经定义以下 **ResolveDataBind** 事件处理程序。定义之后， MapX 将会调用它，并令处理程序选择要使用的对象。

Dispatch ID = 2

语法

ResolveDataBind (flag, nummatches, matches, choice, cancel)

部分	说明
flag	表示事件是否正在由于存在多个 geocol (0)、多个图层 (1) 或多个 geoset (2) 而调用。
nummatches	表示匹配数组中有多少个条目。字符串的数组匹配包含匹配项目的列表。
matches	**Variant**：表示一个**字符串**数组。
choice	返回的变量。默认为 1。0 表示自动选择和令 MapX 选择最适当的对象。
cancel	允许用户取消数据绑定并令其失败。

Remarks

ResolveDataBind 可触发 3 次。

第一次，它选择源表中的列；然后对于所选的列，如果为 >1 则它将选择表；最后对于相应的表，如果为 >1 则选择 geoset。

geodictinonary 列出了用于各个表的 geoset。

ResolveDataBindEx 事件

目的

此事件具有和 ResolveDataBind 事件相同的函数，但是传递 ResolveObjects 集合，而不是字符串数组。因此，对于类似在用户数据检测到多个 geocolumn、检测到多个匹配 geocol 的多个图层或多个 geoset 可

用于已匹配的图层的情况，MapX 将选择第一个对象，除非 ResolveDataBind 或 ResolveDataBindEx 事件处理程序已作定义。定义之后， MapX 将会调用一个对象，并令处理程序选择要使用的对象。

Dispatch ID = 14

语法

ResolveDataBindEx (flag, nummatches, matches, choice, cancel)

部分	说明
flag	表示正在由于存在多个 geocol 而调用的事件。此标志始终设置为 1。
nummatches	表示匹配集合中有多少个条目。
matches	**Variant**：表示一个 **ResolveObject** 对象集合。
choice	返回的变量。默认为 1。0 表示自动选择和令 MapX 选择最适当的对象。
cancel	允许用户取消数据绑定并令其失败。

Remarks

ResolveDataBindEx 只在存在多个 geocolumn 并且只有在 ResolveDataBind 事件设置为 1 时触发。

SelectionChanged 事件

目的

此事件在选择更改时调用，令容器可以对地图上所做选择做出响应。该选择可以由于用户使用选择工具，或者使用 Layer 对象的 Selection 方法之一而改变

Dispatch ID = 1

语法

SelectionChanged

ThemeModifyRequested 事件

目的

此事件在用户双击主题图例时调用。参数为拥有该图例的 Theme 对象。此事件的目的在于能够编写代码，并允许用户在用户双击图例时修改主题或图例属性。

Dispatch ID = 9

语法

ThemeModifyRequested

Theme 对象

ToolUsed 事件

ToolUsed 事件在用户使用地图上的定制工具时调用。此过程允许您确定如何使用工具，例如，它将告知您用户点击何处的地图坐标。在 **ToolUsed** 过程之内，您编写代码以执行工具的既定功能。

此事件在使用标准工具时也将调用，此时事件在用户交互之后调用，但是在为标准功能执行的操作之前。

Dispatch ID = 3

语法

ToolUsed(ToolNum, x1, y1, x2, y2, distance, shift, control, *EnableDefault)

部分	说明
ToolNum	整数：表示工具号。请注意如果您创建多个自定义工具，此事件将在使用任意这些工具时调用，因此，您需要检查 ToolNum 以确定正在使用的工具。
x1, y1	双击：用户点击的地图坐标。
x2, y2	双击：用户结束工具使用的地图坐标；不适用于点工具，后者不允许拖放。
distance	双击：开始位置和结束位置之间的距离的地图单位。
shift	布尔值：表示用户是否按住了 SHIFT 键。
control	布尔值：表示用户是否按住了 CTRL 键。
*EnableDefault	布尔值：如果使用的是标准工具，则此参数控制 MapX 是否允许工具的标准行为生效。默认值为 True。要避免工具的正常行为 （例如取消用户操作），可将此参数设置为 False。

附录 F: MapX 错误代码

此表列出了编写 MapX 应用程序时遇到的 MapX 错误。有关在 Visual Basic、Delphi 和 Powerbuilder 中处理错误的详细信息，请参阅各自的程序员参考手册和 / 或帮助系统。和程序员有关的 C++ 类似信息，请参阅使用 Visual C++。

错误	说明
1000	内存用尽。
1001	没有实施的属性或方法。
1002	此类型的主题不支持的属性。
1003	对象不再有效。
1004	使用您指定的索引没有找到任何对象。
1005	指定的 DataSet 类型无效。
1006	指定的名称不唯一。（名称已由其他对象使用）。
1007	指定的 GeoField 无效。名称未找到，或索引超出范围。
1008	指定的备用 GeoField 无效。名称未找到，或索引超出范围。
1009	无法自动匹配 GeoField 或 BindLayer。
1010	Fields 参数无效。
1011	MapX 中出现意外错误。
1012	指定的 Theme 类型无效。
1013	意外错误。
1014	指定的数据集中未找到任何关键字列。无法创建主题。
1015	DataBinding 进程已经由于设置 ResolveDataBind 事件中的取消属性而被取消。
1016	指定的聚集函数无效。
1017	没有指定源字段。
1018	指定的字段类型无效。
1019	无法刷新数据集。
1020	无法找到此数据集的 BindLayer。无法创建主题。
1021	无法创建主题。
1022	范围无效。
1023	范围数量无效。
1024	无法移除 Theme 类别。

错误	说明
1025	分发方法无效。
1026	填充模式无效。
1027	墨笔样式无效。
1028	Fields 参数无效 － 字段的名称或索引未找到。
1029	无法创建默认主题。
1030	此属性只在设计阶段可用。
1031	指定的图层不可匹配。
1032	匹配或指定的图层不在任意 GeoSet 之中。
1033	指定的 Name 参数无效。
1034	指定的颜色无效。
1035	按值展开无效。
1036	墨笔宽度无效。
1037	指定的坐标未在地图的数字坐标系范围之内。
1038	指定的缩放无效。缩放必须为 > 0。
1039	指定的工具无效。工具号无效或不存在，或者工具类型或光标无效。
1040	指定的纸张单位无效。
1041	指定的鼠标指针无效。
1042	指定的导出格式无效。
1043	无法创建指定文件。
1044	出现 ODBC 错误：< 错误 >。
1045	指定的 ODBCQueryInfo 对象无效。对象类型不正确，或未指定 SqlQuery。
1046	此类型的注释不支持的属性。
1047	字符代码无效。值必须介于 0 和 255 之间。
1048	无法基于已经按照单值聚集的属性数据列创建此类主题。
1049	Row 参数无效：行超出范围，或取操作超出表的结尾。
1050	Column 参数无效。
1051	无法找到或加载 MIODBC.DLL
1052	初始化 MIODBC.DLL 时出错，检查 DLL 版本。
1053	使用无效句柄调用 MIODBC.DLL。
1054	数据集没有字段。没有从源数据向数据集成功加载任何字段。

错误	说明
1055	无法获取临时目录。
1056	无法创建用于新图层的文件名。
1057	未指定关键字列，无法创建新图层。
1058	指定的参考图层并未安装在 geodictionary 之中。
1059	无法创建新图层。图层文件名已存在。
1060	指定的绑定图层类型无效。
1061	指定的绑定图层无效。
1062	无法使用指定的参考列完成此类绑定。
1063	创建地图时出现未知错误。
1064	指定的搜索时间无效。时间必须介于 1 和 3600 秒之间。
1065	指定的显示单位无效。
1066	指定的转换常数无效。
1067	源缺失或无效，或者源不匹配指定的类型。
1068	指定的参考图层不是点参考图层。
1069	点大小无效。
1070	位置无效。
1071	点主题对象的最大大小为 48。
1072	无法创建 geotable。
1073	无法创建图层。
1074	无法设置或检索图层信息。
1075	无法设置或检索标签信息。
1076	无法设置或检索图层样式信息。
1077	无法设置或检索标签样式信息。
1078	图元不再有效。
1079	无法找到或加载 Helper 对话框 DLL。
1080	初始化 Helper 对话框 DLL 时出错。检查 DLL 版本。
1081	对于指定的 geofield 或必需的自动检测到的备用 geofield，没有指定或自动检测到相应对象。
1082	数据集创建之后不允许进行字段添加或移除。
1083	此类型的图层不支持的属性或方法。
1084	出现 Lotus Notes 错误：<error>。

错误	说明
1085	指定的 NotesViewInfo 对象无效。对象类型不正确，或未指定数据库或视图。
1086	在坐标转换期间出现数字溢出上限或下限。在 Windows95/98 之下，所有坐标均限定在 -32767 至 +32767 的范围之内。实际屏幕或地图坐标可能在该范围之内，但是转换溢出。请参阅 IsPointVisible 或 ClipLine。
1087	尝试将 Animation 图层设置为的对象不是有效图层对象。
1088	Filespec 参数无效。
1089	指定的 DataSet 无效。
1090	图元类型无效或不适用。
1091	在独立图元上不允许进行的操作。
1092	更新图元时出错。
1093	添加图元时出错。
1094	删除图元时出错。
1095	图元类型的图元属性或方法无效。
1096	从图层读取图元时出错。
1097	一条线必须至少有 2 个点。一个区域必须至少有 3 个点。
1098	指定的矩形无效。
1099	无法在 DataSet 中找到指定字段。
1100	图元无效。
1101	指定的搜索类型无效。
1102	指定的搜索精度无效。
1103	多个点对象无效。
1104	一个点对象无效。
1105	样式对象无效。
1106	在没有设置 Point 属性之前尝试使用新的 Symbol 或 Text 图元。
1107	指定的面积单位无效。
1108	Features 集合对象中的各个图元必须源自相同图层。
1109	图层没有找到关键字字段。
1110	Feature 或 Features 对象的来源图层不再有效。
1111	无法将变量转换为适当类型。
1112	指定的参考图层没有可匹配的列，并因此无法用作参考图层。
1113	在地图导出时创建位图图像失败。

错误	说明
1114	指定的 Circle 类型无效。
1115	分辨率必须为 >= 3。
1116	Datum 对象无效。
1117	CoordSys 对象无效。
1118	指定的单位无效。
1119	指定的经度原点无效。
1120	指定的纬度原点无效。
1121	指定的标准并行 #1 无效。
1122	指定的标准并行 #2 无效。
1123	指定的方位角无效。
1124	指定的比例因子无效。
1125	指定的假东向无效。
1126	指定的假北向无效。
1127	指定的范围无效。
1128	指定的边界无效。
1129	AffineInfo 对象无效。
1130	尚未指定 Datum。
1131	尚未指定单位。
1132	尚未指定经度原点。
1133	尚未指定纬度原点。
1134	尚未指定标准并行 #1。
1135	尚未指定标准并行 #2。
1136	尚未指定方位角。
1137	尚未指定比例因子。
1138	尚未指定假东向。
1139	尚未指定假北向。
1140	尚未指定范围。
1141	尚未指定边界。
1142	指定的 CoordSys 类型无效。
1143	AffineTransform 对象无效。

错误	说明
1144	指定的 AffineTransform 退化。
1145	定制数据集错误。
1146	地图中没有图层。 Geoset 未保存。
1147	地图中包含临时或用户绘制的图层。已保存的 Geoset 未包括相应图层。
1148	数据集类型不支持动态列绑定。
1149	Style 对象不支持位图符号。
1150	指定的符号类型无效。
1151	尝试通过没有已保存数据集的名称来恢复数据集。
1152	定制数据集错误。方法失败：IMMapXColumnInfoContainer::GetColumnInfoEnumerator。
1153	定制数据集错误。方法失败：IMEnumMapXColumnInfo::Reset。
1154	定制数据集错误。必需的接口不支持：IMMapXColumnInfoContainer。
1155	定制数据集错误。方法失败：IMMapXColumnInfo::GetColumnName。
1156	定制数据集错误。方法失败：IMMapXColumnInfo::GetColumnNumber。
1157	定制数据集错误。方法失败：IMMapXColumnInfo::GetDataType。
1158	定制数据集错误。列包含不支持的类型。
1159	定制数据集错误。方法失败：IMMapXDataset::GetSample。
1160	定制数据集错误。方法失败：IMMapXStaticDataset::BeginFetch。
1161	定制数据集错误。方法失败：IMMapXStaticDataset::FetchData。
1162	定制数据集错误。方法失败：IMMapXStaticDataset::EndFetch。
1163	定制数据集错误。方法失败：IMMapXDynamicDataset::FetchData。
1164	定制数据集错误。方法失败：IMMapXDataset:Init。
1165	定制数据集错误。无法找到用于指定的定制数据集类型的 CLSID。检查定制数据集是否正确注册。
1166	定制数据集错误。无法创建定制数据集对象。检查定制数据集是否正确注册。
1167	操作由于许可限制而不允许进行。
1168	无法打开一个逐层细化组件表。
1169	无法重设指定层次的 Drilldown 图层。
1170	图元过于复杂。图元的组成部分或点过多。
1171	一个 Line 或 Region 图元必须至少有 1 个组成部分。
1172	独立的图元没有连接到地图。在使用其他图元方法或属性之前先使用 Feature.Attach()。

错误	说明
1173	无法将图元连接到地图。确保 Map 对象有效。
1174	创建缓存时出错。
1175	在空图元集合上不允许进行的操作。
1176	参数组合无效。
1177	指定的 Caption 参数无效。
1178	指定的交叉点标志无效。
1179	指定的交叉测试标志无效。
1180	合并图元时出错
1181	指定的弧线距离无效。
1182	指定的角度无效。
1183	Search 方法中指定的 RefineBoundary 缺失或无效。
1184	XY 和 PointRef 数据集不允许动态绑定。
1185	定制数据集错误。必需的接口不支持。
1186	定制数据集错误。 GetSample 必须返回数组。
1187	定制数据集错误。返回的类型不支持。
1188	保存 geoset 时出错。
1189	无法找到坐标系对话框所需文件 <file>。
1190	Layer.UpdateFeature(Target, Source) – “目标”必须为源自 “图层”的图元。
1191	范围和单值主题在 SpatialWare 图层上不可用。
1192	查找方法在 SpatialWare 图层上不可用。
1193	在没有图层的地图上不允许进行的操作。
1194	不允许文本注释为空。
1195	从逐层细分层 <level> 添加关键字 <key > 失败；可能有其他失败。关键字字段未建立索引?
1196	从逐层细分层 <level > 移除关键字 <key > 失败；可能有其他失败。
1197	指定的缩放值下限无效。缩放值必须大于 0。
1198	指定的缩放值上限无效。缩放值必须大于 0。
1199	除非 ComputeTheme 为 false，否则不允许设置值。
1200	指定的样式单位值无效。
1201	无法添加 miThemeNone 类型的主题。

错误	说明
1202	只有范围和单值主题支持 ShowCount 属性。
1203	地图包括可见光栅图层时，地图显示坐标系无法修改。
1204	此线样式不支持交错样式。
1205	使用您指定的名称没有找到任何图元。
1206	只有范围和单值主题支持 AllOthersText 属性。
1207	指定的匹配阈值无效。百分比必须介于 1 和 100 之间。
1208	ComputeTheme 为 false 时， ShowCount 属性不得设置为 true。
1209	采用覆盖图层样式时，不能更改文本字体大小属性。
1210	缓存选项无效。
1211	无法加载指定的光标。
1212	在图层中搜索时出错。
1213	在更新或绘制选择时出错。
1214	图元 ID 无效。
1215	无法设置或检索主题样式信息。
1216	访问主题对象时出错。
1217	创建主题时出错。
1218	访问主题修饰符时出错。
1219	重新计算主题时出错。
1220	主题范围无效。请验证范围是否重叠。您可能需要在设置定制主题范围时将 AutoRecompute 设置为 FALSE。
1221	只有带有定制范围的主题支持此属性或方法。
1222	指定的主题类别索引类型无效。
1223	所有其他类别不支持此属性或方法。
1224	指定的图层 MBRSearch 选项无效。
1225	指定的主要对象类型无效。
1226	在此数据集集合上不允许进行的操作。
1227	创建的数据集没有 SourceRow 信息。请参阅 Datasets.BuildSourceRows。
1228	只有 ComputeTheme 为 false 时才支持此属性或方法。
1229	设置该主题的类别数量时出错。
1230	名为 <filename> 的文件已存在。

错误	说明
1231	无法打开文件 <filename>。验证该文件存在并且您的访问权限足以打开它。
1232	无法打开文件 <filename>。验证该文件存在于 geodictionary 目录或搜索路径的列表中。还要确保您的访问权限足以打开它。
1233	在导出期间写文件时出错。
1234	BeginAccessType 或 Invalid EndAccessType 无效。
1235	主题值必须为正。
1236	堆叠条形图不支持此属性。
1237	只有堆叠条形图支持此属性。
1238	只有分级堆叠条形图和不是堆叠或独立标记的条形图支持 GraduateSizeBy 属性。
1239	指定的 LayerInfo 参数无效。
1240	添加 LayerInfo 参数时出错。
1241	预期的 LayerInfo 参数 <parameter name> 缺失或无效。
1242	无法打开 <table name>。在 geodictionary 中未找到表。
1243	当前符号类型不支持此属性。符号类型必须匹配正在更改其属性的符号类型。
1244	出现主题反串行化错误。
1245	出现主题串行化错误。
1246	出现主题重建错误。
1247	无法修改图层。表为只读。
1248	指定的 Expression 参数无效。
1249	创建图例时出错。
1250	出现图例反串行化错误。
1251	出现图例串行化错误。
1252	出现图例重建错误。
1253	表不可制图。确保表具有一个对象列。如果表为服务器表，确保 mapinfo mapcatalog 中有用于其的条目。
1254	指定的 closematch 最大值无效。 Closematch 值必须大于等于 0。
1255	指定的圆整值无效。
1256	此属性只有在 RoundRanges 属性设置为 true 时可以设置。
1257	此属性只有在 SpreadBy 属性设置为 miSpreadByColor 时可以设置。
1258	此属性只有在 InflectRanges 属性设置为 true 是可以设置。

错误	说明
1259	指定的偏移范围索引无效。
1260	没有使用偏移的足够范围。至少需要三个范围使用偏移。
1261	指定的应用属性无效。
1262	无法在主要包含边界对象或文本对象的图层上应用大小属性。
1263	无法在主要包含边界对象或文本对象的图层上按照大小展开。
1264	在应用大小属性时无法按照颜色展开。
1265	在应用颜色属性时无法按照大小展开。
1266	无法在单值主题上应用大小属性。
1267	指定的标签最大值无效。标签最大值必须介于 0 和 32767 之间。
1268	尝试将 Insertion 图层设置为的对象不是有效图层对象。
1269	指定的 RowValue 对象无效。
1270	指定的 RowValue 对象无效。
1271	指定的 UpdateFeature 参数无效。
1272	将当前工具设置为标准工具时出错。
1273	没有设置插入图层。在使用对象创建工具之前必须设置插入图层。
1274	将当前工具设置为定制工具时出错。
1275	创建标准工具时出错。
1276	创建定制工具时出错。
1277	无法在无缝表上自动创建主题。
1278	指定的关键字无效。
1279	指定的图层为只读。只读图层不可编辑。
1280	指定的图层为不可编辑。插入图层只能设置为可编辑的图层。
1281	指定的图层不包含任意面积对象。质心只能显示用于包含边界面积对象的图层。
1282	指定的图层不包含任意线对象。线的方向只能显示用于包含线对象的图层。
1283	指定的图层不包含任何点对象或面积对象。节点只能显示用于包含点对象或边界面积对象的图层。
1284	指定的图层无效。无法将插入图层设置为逐层细分、光栅、无缝或用户绘制图层。
1285	指定的字段无效。名称未找到，或索引超出范围。
1286	访问地图的插入工具时出错。
1287	插入图层的可编辑属性不能设置为 false。

附录 G: 常数

本节列出了 MapX 常数及其值

AggregationFunctionConstants	AnnotationChangedTypeConstants
AnnotationTypeConstants	ApplyAttributeConstants
AreaUnitConstants	BindLayerTypeConstants
BindLayerTypeConstants	ColorConstants
ColorSpreadingMethodConstants	ConversionConstants
CoordSysTypeConstants	CursorConstants
DataSetTypeConstants	DistribMethod Constants
DotSizeConstants	ExportFormatConstants
FeatureEditModeConstants	FeatureTypeConstants
FieldTypeConstants	FillPatternConstants
GraduationConstants	IntersectionPointConstants
IntersectionTestConstants	LabelAlongConstants
LayerBeginAccessConstants	LayerEndAccessConstants
LayerInfoTypeConstants	LayerPackConstants
LayerSrvLayerOptions	LayerTypeConstants
LineTypeConstants	MapDrawConstants
MapUnitConstants	MouseWheelSupportConstants
PaperUnitConstants	PenStyleConstants
PolyToolFlagConstants	PositionConstants
ResolveDataBindConstants	SearchTypeConstants
SearchResultTypeConstants	SelectionTypeConstants
SpreadByConstants	StyleUnitConstants
SymbolTypeConstants	ThemeTypeConstants
ToolConstants	ToolFlagConstants
ToolTypeConstants	

AggregationFunctionConstants

miAggregationSum = 0

miAggregationAverage = 1

miAggregationCount = 2

miAggregationIndividual = 4

miAggregationAuto = 5

AnnotationChangedTypeConstants

miAddAnnotation = 0

miDeleteAnnotation = 1

miSelectAnnotation = 2

miEditAnnotation = 3

AnnotationTypeConstants

miSymbolAnnotation = 1

miTextAnnotation = 6

ApplyAttributeConstants

miApplyAttributeAll = 0

miApplyAttributeColor = 1

miApplyAttributeSize = 2

AreaUnitConstants

miUnitSquareMile = 14

miUnitSquareKilometer = 15

miUnitSquareInch = 16

miUnitSquareFoot = 17

miUnitSquareYard = 18

miUnitSquareMillimeter = 19

miUnitSquareCentimeter = 20

miUnitSquareMeter = 21

miUnitSquareSurveyFoot = 22

miUnitSquareNauticalMile = 23

miUnitSquareTwip = 24

miUnitSquarePoint = 25

miUnitSquarePica = 26

miUnitSquareDegree = 27

miUnitAcre = 28

miUnitHectare = 29

miUnitSquareLink = 33

miUnitSquareChain = 34

miUnitSquareRod = 35

miUnitPerch = 36

miUnitRod = 37

BindLayerTypeConstants

miBindLayerTypeNormal = 0

miBindLayerTypeXY = 1

miBindLayerTypePointRef = 2

CircleTypeConstants

常数	值	说明
miCircleTypeScreen	0	CreateCircularRegion 方法创建了在屏幕显示为圆形的区域。但是，圆上不同的点距离圆心的地理距离可能不同。
miCircleTypeMap	1	CreateCircularRegion 方法创建的区域上的点到中心的地理距离均相同。但是，取决于使用的坐标系和投影，区域可能会扭曲，而不是一个 “圆形”。

ColorConstants

注：这些编号表示 **OLE_COLOR 值** (BGR)，这些值可以分配给属性，例如 **Style.LineColor** 属性。指定 geoset 中的颜色时，可使用 RGB 颜色设置来替代。

miColorBlack = 0

miColorRed = 255

miColorGreen = 65280

miColorBlue = 16711680

miColorMagenta = 16711935

miColorCyan = 16776960

miColorWhite = 16777215

miColorLightGray = 12632256

miColorDarkGray = 4210752

miColorGray = 8421504

miColorPaleYellow = 13697023

miColorLightYellow = 8454143

miColorYellow = 65535

miColorLimeGreen = 12639424

miColorTeal = 8421440

miColorDarkGreen = 16384

miColorMaroon = 128

miColorPurple = 8388736

miColorOrange = 33023

miColorKhaki = 7051175

miColorOlive = 32896

miColorBrown = 4210816

miColorNavy = 8404992

miColorScrollBars = 0x80000000

miColorDesktop = 0x80000001

miColorActiveTitleBar = 0x80000002

miColorInactiveTitleBar = 0x80000003

miColorMenuBar = 0x80000004

miColorWindowBackground = 0x80000005

miColorWindowFrame = 0x80000006

miColorMenuText = 0x80000007

miColorWindowText = 0x80000008

miColorTitleBarText = 0x80000009

miColorActiveBorder = 0x8000000A

miColorInactiveBorder = 0x8000000B

miColorApplicationWorkspace = 0x8000000C

miColorHighlight = 0x8000000D

miColorHighlightText = 0x8000000E

miColorButtonFace = 0x8000000F

miColorButtonShadow = 0x80000010

miColorGrayText = 0x80000011

miColorButtonText = 0x80000012

miColorInactiveCaptionText = 0x80000013

miColor3DHighlight = 0x80000014

miColor3DDarkShadow = 0x80000015

miColor3DLight = 0x80000016

miColorInfoText = 0x80000017

miColorInfoBackground = 0x80000018

ColorSpreadingMethodConstants

miColorMethodRGB = 0

miColorMethodHSV = 1

ConversionConstants

miMapToScreen = 0

miScreenToMap = 1

CoordSysTypeConstants

常数	值	说明
miAlbersEqualAreaConic	9	
miAzimuthalEquidistant	5	仅限极点对象
miCylindricalEqualArea	2	
miEckertIV	14	
miEckertVI	15	
miEquidistantConic	6	也称为普通圆锥
miGall	17	
miHotineObliqueMercator	7	
miLambertAzimuthalEqualArea	4	仅限极点对象
miLambertConformalConic	3	
miLambertConformalConicBelgium	19	为比利时 1972 修改
miLongLat	1	经度 / 纬度
miMercator	10	
miMillerCylindrical	11	
miMollweide	13	
miNewZealandMapGrid	18	
miNonEarth	0	未投影，迪卡尔坐标系。
miRobinson	12	
miSinusoidal	16	
miStereographic	20	
miSwissObliqueMercator	25	
miTransverseMercator	8	也称为高斯－克留格尔
miTransverseMercatorDenmarkS34J	21	为丹麦系统 34 Jylland-Fyn 修改
miTransverseMercatorDenmarkS34S	22	为西兰岛修改
miTransverseMercatorDenmarkS45B	23	为丹麦系统 45 博恩荷姆修改
miTransverseMercatorFinland	24	为 Finnish KKJ 修改
miMercatorStdParallel	26	区域墨卡托投影
miPolyconic	27	多圆锥投影
miAzimuthalEquidistantEqualOblique	28	方位等距投影 （所有纬度原点）

常数	值	说明
miLambertAzimuthalEqualArea Oblique	29	朗伯方位等面积投影 （所有纬度原点）
miCassiniSoldner	30	Cassini - Soldner 投影

CursorConstants

miDefaultCursor = 0

miArrowCursor = 1

miCrossCursor = 2

miIBeamCursor = 3

miIconCursor = 4

miSizeCursor = 5

miSizeNESWCursor = 6

miSizeNSCursor = 7

miSizeNWSECursor = 8

miSizeEWCursor = 9

miUpArrowCursor = 10

miHourglassCursor = 11

miNoDropCursor = 12

miArrowHourglassCursor = 13

miArrowQuestionCursor = 14

miSizeAllCursor = 15

miArrowToolCursor = 16

miPanCursor = 17

miCenterCursor = 18

miZoomInCursor = 19

miZoomOutCursor = 20

miSymbolCursor = 21

miTextCursor = 22

miSelectCursor = 23

miRadiusSelectCursor = 24

miRectSelectCursor = 25

miRegionSelectCursor = 26

miInfoCursor = 27

miSelectPlusCursor = 28

miSelectRadiusPlusCursor = 29

miSelectRectPlusCursor = 30

miSelectRegionPlusCursor = 31

miSelectMinusCursor = 32

miSelectRadiusMinusCursor = 33

miSelectRectMinusCursor = 34

miSelectRegionMinusCursor = 35

miLabelCursor = 36

miDrillDownExpandCursor = 37

miDrillDownContractCursor = 38

miInfoCursorOld = 39

miCustomCursor = 40

DataSetTypeConstants

miDataSetDAO = 1

miDataSetODBC = 2

miDataSetUnbound = 3

miDataSetGlobalHandle = 4

miDataSetOLEData = 5

miDataSetLayer = 6

miDataSetNotesView = 7

miDataSetNotesQuery = 8

miDataSetSafeArray = 9

miDataSetOCI = 15

miDataSetDelphi = 1010

miDataSetDelphi4 = 1011

miDatasetDelphi5 = 1012

miDataSetADO = 12

miDataSetRDO = 13

miDatasetXML = 14

DistribMethod Constants

miCustomRanges = 0

miEqualCountPerRange = 1

miEqualRangeSize = 2

miNaturalBreak = 3

miStandardDeviation = 4

DotSizeConstants

miDotSizeSmall = 0

miDotSizeLarge = 1

ExportFormatConstants

miFormatWMF = 0

miFormatBMP = 1

miFormatGIF = 2

miFormatJPEG = 3

miFormatTIF = 4

miFormatPNG = 5

miFormatPSD = 6

FeatureEditModeConstants

miEditModeFeature = 0x1

miEditModeNode = 0x2

miMoveDuplicateNodes = 0x4

miDeleteDuplicateNodes = 0x8

miEditModeAddNode = 0x40

FeatureTypeConstants

miFeatureTypeRegion = 0

miFeatureTypeLine = 1

miFeatureTypeSymbol = 2

miFeatureTypeMixed = 3

miFeatureTypeUnknown = 4

miFeatureTypeText = 5

miFeatureTypeNull = 6

miFeatureTypeMultipoint = 7

miFeatureTypeCollection = 8

FieldTypeConstants

miTypeString = 0

miTypeNumeric = 1

miTypeDate = 2

miTypeInt = 3

miTypeSmallInt = 4

miTypeFloat = 5

miTypeLogical = 6

FillPatternConstants

miPatternNoFill = 0

miPatternHollow = 1

miPatternSolid = 2

miPatternHorizontal = 3

miPatternVertical = 4

miPatternFDiag = 5

miPatternFilBDiag = 6

miPatternCross = 7

miPatternDiagCross = 8

GraduationConstants

miGraduateBySquareRoot = 0

miGraduateByConstant = 1

miGraduateByLogarithm = 2

IntersectionPointConstants

miIntersectCrossings = 9

miIntersectCommon = 10

miIntersectAll = 11

IntersectionTestConstants

miIntersectCentroidWithinFeature = 0

miIntersectFeature = 1

miIntersectEntirelyWithinFeature = 2

LabelAlongConstants

miLabelAlongNone = 0

miLabelAlongParallel = 1

miLabelAlongMultiSegment = 2

LayerBeginAccessConstants

miAccessRead= 0

miAccessReadWrite = 1

LayerEndAccessConstants

miAccessEnd= 0

LayerInfoTypeConstants

miLayerInfoTypeTab = 0

miLayerInfoTypeUserDraw = 1

miLayerInfoTypeRaster = 2

miLayerInfoTypeShape = 3

miLayerInfoTypeServer = 4

miLayerInfoTypeGeodictUserName = 5

miLayerInfoTypeTemp = 6

miLayerInfoTypeNewTable = 7

LayerPackConstants

miPackGraphics = 1

miRebuildGraphics = 2

miPackIndex = 4

miPackRebuildIndex = 8

miPackData = 16

miPackCompactDB = 32

miPackAll = 21 (miPackGraphics & miPackIndex & miPackData)

LayerSrvLayerOptions

miLayerCacheOn = 0

miLayerCacheOff = 1

miLayerMBRSearchOn = 0

miLayerMBRSearchOff = 2

LayerTypeConstants

miLayerTypeNormal = 0

miLayerTypeRaster = 2

miLayerTypeSeamless = 4

miLayerTypeUnknown = 5

miLayerTypeUserDraw = 6

miLayerTypeDrilldown = 7

LineTypeConstants

miLineTypeNone = 0

miLineTypeSimple = 1

miLineTypeArrow = 2

MapDrawConstants

miDrawBegin = 1

miDrawEnd = 2

MapUnitConstants

miUnitMile = 0

miUnitKilometer = 1

miUnitInch = 2

miUnitFoot = 3

miUnitYard = 4

miUnitMillimeter = 5

miUnitCentimeter = 6

miUnitMeter = 7

miUnitSurveyFoot = 8

miUnitNauticalMile = 9

miUnitTwip = 10

miUnitPoint = 11

miUnitPica = 12

miUnitDegree = 13

miUnitLink = 30

miUnitChain = 31

miUnitRod = 32

MouseWheelSupportConstants

miNoMousewheelSupport = 1

miMousewheelNoAutoScroll = 2

miFullMousewheelSupport = 3

PaperUnitConstants

miPaperUnitMile = 0

miPaperUnitKilometer = 1

miPaperUnitInch = 2

miPaperUnitFoot = 3

miPaperUnitYard = 4

miPaperUnitMillimeter = 5

miPaperUnitCentimeter = 6

miPaperUnitMeter = 7

miPaperUnitSurveyFoot = 8

miPaperUnitNauticalMile = 9

miPaperUnitTwip = 10

miPaperUnitPoint = 11

miPaperUnitPica = 12

miPaperUnitDegree = 13

miPaperUnitLink = 30

miPaperUnitChain = 31

miPaperUnitRod = 32

PenStyleConstants

miPenNone = 0

miPenSolid = 1

注：尽管没有为这些样式定义的常数，但是还有其他可用墨笔样式。请参阅 **Style.LineStyle** 属性。

PolyToolFlagConstants

miPolyToolBegin = 0

miPolyToolEnd = 1

miPolyToolEndEscaped = 2

miPolyToolInProgress = 3

PositionConstants

miPositionCC = 0

miPositionTL = 1

miPositionTC = 2

miPositionTR = 3

miPositionCL = 4

miPositionCR = 5

miPositionBL = 6

miPositionBC = 7

miPositionBR = 8

ResolveDataBindConstants

miChooseField = 0

miChooseLayer = 1

miChooseGeoSet = 2

SearchTypeConstants

miSearchTypeCentroidWithin = 0

miSearchTypePartiallyWithin = 1

miSearchTypeEntirelyWithin = 2

SearchResultTypeConstants

miSearchResultDefault = 0x0000

miSearchResultTopmost = 0x0001

miSearchResultRegion = 0x0002

miSearchResultPoint = 0x0004

miSearchResultLine = 0x0008

miSearchResultText = 0x0010

miSearchResultAll = 0xFFFE

miSearchResultMultipoint = 0x0020

miSearchResultCollection = 0x0040

SelectionTypeConstants

miSelectionNew = 0

miSelectionAppend = 1

miSelectionRemove = 2

SpreadByConstants

miSpreadByColor = 1

miSpreadByNone = 0

miSpreadBySize = 2

StyleUnitConstants

miStyleUnitPixel = 0

miStyleUnitTenthsOfPoint = 1

SymbolTypeConstants

miSymbolTypeBitmap = 1

miSymbolTypeTrueTypeFont = 0

miSymbolTypeVector = 2

ThemeTypeConstants

miThemeRanged = 0

miThemeBarChart = 1

miThemePieChart = 2

miThemeGradSymbol = 3

miThemeDotDensity = 4

miThemeIndividualValue = 5

miThemeAuto = 6

miLabelRangedTheme = 7

miLabelIndividualValuesTheme = 8

miThemeNone = 9

ToolConstants

miArrowTool = 1000

miPanTool = 1001

miCenterTool = 1002

miZoomInTool = 1003

miZoomOutTool = 1004

miSymbolTool = 1005

miTextTool = 1006

miSelectTool = 1007

miRadiusSelectTool = 1008

miRectSelectTool = 1009

miPolygonSelectTool = 1010

miLabelTool = 1011

miAddLineTool = 1012

miAddPolylineTool = 1013

miAddRegionTool = 1014

miAddPointTool = 1015

ToolFlagConstants

miToolBegin = 0

miToolEnd = 1

miToolEndEscaped = 2

miToolInProgress = 3

ToolTypeConstants

miToolTypePoint = 0

miToolTypeLine = 1

miToolTypeCircle = 2

miToolTypeMarquee = 3

miToolTypePoly = 4

miToolTypePolygon = 5

附录 H: 创建表达式

表达式在 MapX 中非常实用，其优点不一而足。**Layer.Search** 方法可用于具有表达式选择图层图元的子集。此时，表达式求值应为布尔值，例如 year(Received) = 1990 and month(Received) > 4。

DataSet.AddField 方法使用表达式来通过将表达式应用于现有的列创建新的字段。这些表达式求值，得到结果可以为日期、数字或字符串值，例如 "Female_Pop + Male_Pop" 或 "Proper$(City) + ""、 "" + UCase$(State)"

表达式使用数据集字段名称、常数 （例如特定数据值）或变量代用值以及对列进行操作的函数和运算符构成。

MapX 允许在表达式中使用变量替代值。表达式可以包含对变量名的引用。变量是一个带有名称的变化量。当表达式发现无法解析的标识符时，它将尝试在变量集合中进行解析。变量可以为任意类型 （日期、字符串、数字等，或 MapX Feature 对象。）

表达式中的字符串变量应该使用双引号引起来，以便 MapX 可以将其视为字符串，而不是将其视为数据集字段名或变量。

在表达式中， MapX 将执行从布尔值到用于预期为字符串参数的函数参数字符串的类型转换。

将日期指定为字符串常数时，以下项目适用：

- 由月份、日期和可选年份组成的日期。
- 年份由四位数字指定。
- 整个日期字符串应该使用双引号引起来。
- 日期的组成部分应该通过连字符或短横线间隔开。

目录

- 运算符
- 函数

运算符

以下是在 MapX 中使用的运算符列表和相关主题。

运算符

- 地理运算符
- 数学运算符
- 字符串运算符
- 比较运算符
- 数字比较

- 字符串比较
- 日期比较
- 逻辑比较
- 逻辑运算符

相关主题：

- 字符串子句
- 日期子句
- 关键字
- 运算符优先级

地理运算符

MapX 支持若干种地理运算符：contains、 contains_within、 entirely_within、 intersects 和 within。这些运算符可用于任意 MapX 表达式。所有地理运算符均为中缀运算符 （即在两个对象之间运算，返回布尔值）。

运算符	如果以下条件成立，求值为 true：
object1 **CONTAINS** object2	第二个物体的质心在第一个物体之内。
object1 **CONTAINS ENTIRE** object2	第二个物体完全在第一个物体之内。
object1 **CONTAINS PART** object2	第一个物体接触第二个物体。
object1 **ENTIRELY WITHIN** object2	第一个物体完全在第二个物体之内。
object1 **INTERSECTS** object2	第一个物体接触第二个物体。
object1 **PARTLY WITHIN** object2	第一个物体接触第二个物体。
object1 **WITHIN** object2	第一个物体的质心在第二个物体之内。

数学运算符

A + B	+	加法
A - B	-	减法
-A （负）	-	求反
A * B	*	乘法
A / B	/	除法
A\B	\	整除
A ^ B	^	求幂

注意您可以：

- 向日期添加数字，生成另一个日期。
- 从日期减去一个数字，生成另一个日期。
- 从日期减去一个日期，生成一个数字。

向日期添加数字时，或者从日期减去数字时， MapX 将数字视为指定日期的数字。如果要添加或减去一周，可以使用数字 7。要添加或减去一个月，可使用 30 或 31。当 MapX 从一个日期减去另一个日期时，得到的结果数字表示天数。

字符串运算符

&	“concatenation” — 联接字符串和字符串表达式。
+	“concatenation” — 联接字符串和字符串表达式。

注：切记字符串变量必须使用双引号引起来。

比较运算符

=	等于
<>	不等于
>	大于
<	小于
>=	大于或等于
<=	小于或等于

数字比较

数字比较以表达式的数字值和数字常数为基础。

示例

中间年龄为 42 的所有的行。

- MED_AGE=42

注：此表达式只查询中间年龄为 42 的所有记录。当中间年龄数据包含小数部分时 （例如 MapInfo 提供的人口统计数据），则不太可能有许多区域的中间年龄恰恰为 42。此时，您可能需要使用 **Round** 函数来搜索中间年龄近似等于 42 的记录。

字符串比较

字符串比较以字符串的确切字符内容基础。此时 > 表示 “字母排序大于”（例如在字母表中）， < 表示 “字母顺序小于”。

在表达式中键入字符串常数时，应该使用双引号将其引起来，以便 MapX 可以将其视为字符串，而不是将其视为列名。

示例

供货商为 Acme 的所有的行。

- VENDOR=”Acme”

注：等号运算符是区分大小写的。

日期比较

切记使用引号将日期常数引起来，并使用月日年的顺序。

示例

在 1991 年 10 月 9 日之后接收到的所有信息。

- RECEIVED>stringtodate("10-9-91")

注：此表达式没有查询在 1991 年 10 月 9 日接收的信息。如果需要相应信息，可以使用 >= 运算符。

- RECEIVED>=stringtodate("10-9-91")

示例

在 8 月之前接收到所有记录，无论哪年。

- Month(RECEIVED)<8

逻辑比较

示例

所有已经发货的项目。

- Shipped

注：Shipped 列是一个逻辑列。其中包含 T 用于真， F 用于假或否。在定单发货之后，将会标记为 T。否则为没有发货。对于要发货的顶点，表达式 27 求值为 true。对于没有发货的定单求值为 false。

示例

所有没有发货的项目。

- Str$(Shipped)=”F”
- Not Shipped

逻辑运算符

and	如果 （且只有）其参数 （其联接在一起的表达式）均为 true 则为 true。查询到的记录必须同时满足这两个条件。
or	如果其参数 （其联接在一起的表达式）中的一个或两者为 true 则为 true。查询到的记录必须同时满足其中的一个条件。如果满足两个条件也将会查询到。
not	如果其参数 （其联接在一起的表达式）为 false 则为 true。如果不符合陈述的条件则查询到该记录。

And、 or 和 not 均为逻辑运算符。在使用方法时可以组合使用，类似于 **Layer.Search** 方法。

例如，假定您需要查询哥伦比亚镇中的价值 $250,000 或更高的所有地产。为此需要组合使用两个使用 and 操作符的简单表达式：

- COUNTY = ”Columbia” and VALUE >= 250000

现在假定您需要查询哥伦比亚镇中的价值 $250,000 或更高的所有地产，那么应该怎么办？您可以使用 not 来否定前一表达式的第一个子句：

- not(COUNTY=”Columbia”) and VALUE>=250000

在需要指定备用条件时，您可以使用 or，如下所示：

- COUNTY=”Columbia” or COUNTY=”Greene”

字符串子句

示例

从 N 到 Q 的所有客户。

- LAST_NAME >= ”N” and LAST_NAME <”R”

注：表达式的第一部分检查字母排序为 n 或大于 （在之后）n 的名称。表达式的第二部分检查字母排序小于 （在之前）r 的名称。任意首字母为 n 至 q 的的名称均满足此条件。

示例

所有姓氏首字母为 C 的客户。

- LAST_NAME>=”C” and LAST_NAME<”D”

注：此表达式的逻辑和表达式 47 的逻辑相同。

日期子句

示例

在 1990 年 8 月接收到的所有记录。

- Month(RECEIVED)=8 and Year(RECEIVED)=1990

注：在此表达式中，我们显式指定年份，使用 ear(<somecolumn>)î 函数将其从日期中提取出来。

示例

在 1989 年 7 月或 9 月接收到的所有记录

- month(RECEIVED)=(7, 9) and year(RECEIVED)=89

关键字

MapX 支持使用 like 和 between 关键字。

以下示例展示了 between 关键字的使用：

- PRICE between 50000 and 100000

您还可以对字符串使用 between

关键字 like 用于字符串模式匹配。

- 下划线字符匹配单字符。
- 百分比字符匹配 0 或多个字符。

比较运算符区分大小写。

示例

要确定字符串是否以 South 为起始，请使用如下示例：

- STATE like "South%"

运算符优先级

在 MapX 对表达式求解时，它需要了解先对表达式的哪一部分求解。这就称为优先级。通过转换，特定的运算符将分配不同的优先级。优先级较高先行求解。下表按照其求解顺序列出了 MapX 的运算符。优先级相同的运算符依次从左到右求解，但求幂除外，后者从右侧开始求解。

优先级最高：	括号
	求幂
	求反
	乘、除
	加、减
	地理运算符
	比较运算符
	Not
	And
优先级最低：	Or

例如，表达式 3+4*2 的结果为 11。这是因为乘法比加法的优先级高，进而先行求解的结果。括号必须用于强制 MapX 先进行加法运算：(3+4)*2。

现在考虑此表达式，用于查询 1989 年 7 月或 9 月的所有记录。

- year(RECEIVED)=89 and month(RECEIVED)=7 or month(RECEIVED)=9

尽管 “year(RECEIVED)=89 and month(RECEIVED)=7” 括在括号之内，但由于 and 比 or 具有更高的优先级，MapX 将此表达式处理如下。此例中，89 年 7 月的任意记录或任意年份 9 月的记录均将查询到。这可能和我们的预期相左。但是，通过向第二个表达式添加括号，即可获取所需结果：

- year(RECEIVED)=89 and (month(RECEIVED)=7 or month(RECEIVED)=9)

注：当您不确定 MapX 针对具有若干运算符的表达式如何求解时，您应该使用括号来将所需的元素括起来。

函数

函数取用数据值并对其执行某些运算来得到新的值。 MapX 的大部分函数均采用一或两个参数。参数既可以是列，也可以是另一个表达式。

MapX 对于地理函数使用 obj 或 object 关键字：这些函数有 Area、CentroidX、CentroidY、ObjectLen 和 Perimeter。此关键字告知 MapX 它需要基于表中的地理对象来取值，而不是表中所列的数据。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

MapX 函数

Abs( ) 函数	Acos( ) 函数	Area( ) 函数
Asc( ) 函数	Asin( ) 函数	Atn( ) 函数
Buffer( ) 函数	Centroid( ) 函数	CentroidX( ) 函数
CentroidY( ) 函数	Chr$( ) 函数	Cos( ) 函数
CurDate( ) 函数	Day( ) 函数	DeformatNumber$( ) 函数
Distance( ) 函数	Exp( ) 函数	Fix( ) 函数
Format$( ) 函数	FormatDate$( ) 函数	FormatNumber$( ) 函数
InStr( ) 函数	Int( ) 函数	LCase$( ) 函数
Left$( ) 函数	Log( ) 函数	LTrim$( ) 函数
Maximum( ) 函数	MBR( ) 函数	Mid$( ) 函数
Minimum( ) 函数	Month( ) 函数	ObjectLen( ) 函数
ObjectType( ) 函数	Perimeter( ) 函数	Proper$( ) 函数
Right$( ) 函数	Round( ) 函数	RTrim$( ) 函数
Sgn( ) 函数	Sin( ) 函数	Space$( ) 函数
Sqr( ) 函数	StringCompare( ) 函数	StringCompareIntl( ) 函数
StringToDate( ) 函数	Str$( ) 函数	Tan( ) 函数
UCase$( ) 函数	Val( ) 函数	Weekday( ) 函数
Year( ) 函数		

Abs( ) 函数

目的

此函数返回数值的绝对值。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**Abs**(num_expr)

部分	说明
num_expr	数字表达式。

注解

Abs( ) 函数返回由 num_expr 指定的表达式的绝对值。

如果 num_expr 的值大于或等于 0， Abs( ) 返回的值将等于 num_expr。如果 num_expr 的值为负， Abs( ) 返回的值将等于 num_expr 的值乘以 -1。

Acos( ) 函数

目的

此函数返回数值的反余弦值。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**Acos**(num_expr)

部分	说明
num_expr	介于两个减法对象中的数字表达式，包含。

注解

Acos( ) 函数返回由 num_expr 指定的数值的反余弦值。换而言之， Acos( ) 返回的角度的余弦值等于 num_expr。

从 Acos( ) 返回的结果表示为角度，以弧度表示。这一角度将介于 0 和 π 弧度 （假定 π 约等于 3.141593，并且 π /2 弧度等于 90 度）） 之间。

要将角度值转换为弧度值，需要将该值乘以 π /ISO。要将弧度值转换为角度值，需要将该值乘以 ISO/ π 。

由于余弦值介于 +1 和 -1 之间，因此表达式 num_expr 表示的值不应大于 1，也不应小于 -1。

Area( ) 函数

目的

此函数返回对象的地理面积。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**Area**(obj_expr, unit_name)

部分	说明
obj_expr	位于对象表达式之中。
unit_name	表示面积单位名称的字符串 （例如 sq km）。

注解

Area( ) 函数返回由 obj_expr 指定的地理对象的面积。

该函数返回使用 unit_name 参数指定的单位的测量面积值。例如，要获取以英亩为单位的面积值，可以将 unit_name 参数指定为 acre。请参阅 “设置面积单位”语句，以获取有关可用单位名称列表。

只有地区、椭圆、矩形和圆整矩形可以具有一定面积。按照定义，任意点、弧、文本、线、多义线对象的 Area( ) 均为 0。用于圆整矩形时，Area( ) 函数返回近似值。MapX 将圆整矩形视为常规矩形计算其面积。

Asc( ) 函数

目的

此函数返回字符串表达式中第一个字符的字符代码。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[int=]**Asc**(string_expr)

部分	说明
string_expr	字符串表达式

注解

Asc( ) 函数返回表示由 string_expr 指定的字符串的第一个字符的字符代码。

如果 string_expr 为空字符串，则 Asc( ) 函数返回的值为 0。

请注意 Windows 和 Macintosh 操作环境使用不同的字符集。因此，由 Asc( ) 函数返回的值部分取决于运行时使用的操作环境。但是，所有 MapInfo 环境在 32 （空格）至 126 （波浪线）之间的范围内具有公共的字符代码。

在支持双字节字符集的系统上 （例如 Windows 日文版）：如果 string_expr 的第一个字符是单字节字符，Asc( ) 将返回 0 - 255 之间的数字；如果 string_expr 的第一个字符是双字节字符，则 Asc( ) 将返回 256 - 65,535 之间的值。

在不支持双字节字符集的系统上， Asc( ) 将返回 0 - 255 之间的数字。

Asin( ) 函数

目的

此函数返回数值的反正弦值。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]Asin (num_expr)

部分	说明
num_expr	用于两个减法对象中的数字表达式，包含。

注解

Asin( ) 函数返回由 num_expr 指定的数值的反正弦值。换而言之， Asin( ) 返回的角度的正弦值等于 num_expr。

从 Asin( ) 返回的结果表示为角度，以弧度表示。这一角度将介于 - π /2 和 π /2 弧度 （假定 π 约等于 3.141593，并且 π /2 弧度等于 90 度））之间。

要将角度值转换为弧度值，需要将该值乘以 π /ISO。要将弧度值转换为角度值，需要将该值乘以 ISO/ π 。

由于正弦值介于 +1 和 -1 之间，因此表达式 num_expr 表示的值不应大于 1，也不应小于 -1。

Atn( ) 函数

目的

此函数返回数值的反正切值。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**Atn**(num_expr)

部分	说明
num_expr	数字表达式。

注解

Atn( ) 函数返回由 num_expr 指定的数值的反正切值。换而言之， Atn( ) 返回的角度的正切值等于 num_expr。 num_expr 表达式可以等于任意数值。

从 Atn( ) 返回的结果表示为角度，以弧度表示，取值范围介于 - π /2 和 π /2 弧度之间。

要将角度值转换为弧度值，需要将该值乘以 π /ISO。要将弧度值转换为角度值，需要将该值乘以 ISO/ π 。

Buffer( ) 函数

目的

此函数返回一个表示缓冲对象的区域对象 （现有对象的指定缓冲距离中的区域）

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[region object=]**Buffer**(inputobject, resolution, width, unit_name)

部分	说明
inputobject	对象表达式
resolution	SmallInt 值表示每个角落处的每个圆的节点数
width	是一个浮点值，表示缓冲半径；如果 width 为负，并且如果 inputobject 是闭合的对象，则返回的对象将表示小于原始对象的对象
unit_name	是宽度使用的距离单位的名称 （例如 mi 代表英里， km 代表公里）

注解

Buffer( ) 函数返回代表缓冲的区域。

Centroid( ) 函数

目的

此函数返回对象的质心 （中点）。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[Point object=]**Centroid**(obj_expr)

部分	说明
obj_expr	一个对象表达式。

注解

Centroid( ) 函数返回一个点对象，该对象位于指定的 obj_expr 对象的质心。

注：在 MapInfo 中，区域质心表示的不是质量的中心。取而代之的是，质心表示用于自动标记、地理编码和主题饼图和的位置。如果您在重新造型模式下编辑地图，您可以通过拖放来重新定位区域质心。

如果 obj_expr 参数表示点对象， Centroid( ) 函数返回点的位置。

如果 obj_expr 参数表示线对象， Centroid( ) 函数返回线的两端之间的中点。

如果 obj_expr 参数表示多义线对象，则 Centroid( ) 函数返回位于多义线中段的中点。

如果 obj_expr 参数表示任意其他类型对象，则 Centroid( ) 函数返回原始对象的真正质心处的点。对于矩形、弧、文本和椭圆对象，则质心位置介于对象的上、下两半部分之间。但是对于区域对象，质心位置则始终位于有问题的对象 “之上”，不在对象范围的中间部分。

CentroidX( ) 函数

目的

此函数返回对象的质心的 X 坐标。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**CentroidX**(obj_expr)

部分	说明
obj_expr	一个对象表达式

注解

CentroidX( ) 函数返回指定对象的质心的 X 坐标 （例如经度）部分。请参阅 Centroid( ) 函数，查看有关对于不同图形对象类型而言 （线和区域等），质心位置的概念意味着什么的讨论。

坐标信息返回到 Map.NumericCoordSys 中。默认情况下， MapX 使用经度、纬度坐标系。

CentroidY( ) 函数

目的

此函数返回对象的质心的 Y 坐标。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**CentroidY**(obj _expr)

部分	说明
obj_expr	一个对象表达式

注解

CentroidY( ) 函数返回指定对象的质心的 Y 坐标 （例如纬度）部分。请参阅 Centroid( ) 函数，获取有关对于不同的图形对象类型而言 （线和区域等），质心位置的概念意味着什么的讨论。

坐标信息返回到 Map.NumericCoordSys 中。默认情况下， MapX 使用经度、纬度坐标系。

Chr$( ) 函数

目的

此函数返回相应于指定字符代码的单字符字符串。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[string=]**Chr$**(num_expr)

部分	说明
num_expr	从 0 至 255 的整数值 （或者，如果使用双字节字符集，即从 0 至 65,535），包含

注解

Chr$( ) 函数返回字符串，单字节，基于在 num_expr 参数中指定的字符代码。在大多数系统上，num_expr 应该为 0 和 255 之间的正整数。在支持双字节字符集的系统上 （例如 Windows 日文版），num_expr 可以具有从 0 至 65,535 之间的值。

如果 num_expr 参数是小数， MapX 将圆整至最接近的整数。

字符 12 是换页字符。这样，您就可以声明 Print Chr$(12)，清空消息窗口。字符 10 是换行字符，参阅以下示例。

字符 34 是双引号 (")。如果字符串表达式包含函数调用 Chr$(34)， MapX 在字符串中嵌入双引号字符。

Cos( ) 函数

目的

此函数返回数值的余弦值。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**Cos** (num_expr)

部分	说明
num_expr	以弧度表示角度的数字表达式

注解

Cos( ) 函数返回 num_expr 数字值的余弦值，是以弧度表示的角度。从 Cos( ) 返回的结果将介于 1 和 -1 之间。

要将角度值转换为弧度值，需要将该值乘以 π /ISO。要将弧度值转换为角度值，需要将该值乘以 ISO/ π 。

CurDate( ) 函数

目的

使用 YYYYMMDD 格式返回当前日期。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

CurDate( )

返回值

日期

注解

Curdate( ) 函数返回表示当前日期的 Date 值。其格式将始终为 YYYYMMDD。要将该值更改为本地系统格式的字符串，可使用 FormatDate$( ) 或 Srt$( ) 函数。

Day( ) 函数

目的

从 Date 表达式返回日期部分。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

Day(date_expr)

date_expr 是 Date 表达式

返回值

从 1 至 31 的 SmallInt

注解

Day( ) 函数返回从 1 至 31 的整数值，表示指定日期的当月日期。例如，如果指定日期为 12/17/93，则 Day( ) 函数返回的值为 17。

DeformatNumber$( ) 函数

目的

从表示数字的字符串移除格式。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[string=]**DeformatNumber$** (numeric_string)

部分	说明
numeric_string	表示数值的字符串，例如 12,345,678

注解

此函数返回表示数字的字符串。返回值不包括千分位分隔符，无论 numeric_string 参数是否包含千分位分隔符。返回的值使用句点作为小数分隔符，无论用户的计算机是否设置其他字符作为小数分隔符。

Distance( ) 函数

目的

此函数将返回两个位置之间的距离。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**Distance**(x1 , x2 , y1 , y2 , unit_name)

部分	说明
x1 和 x2	x 坐标 （例如经度）
y1 和 y2	y 坐标 （例如纬度）
unit_name	表示距离单位名称的字符串 （例如 km）

注解

Distance( ) 函数计算两个位置之间的距离。

该函数返回使用 unit_name 参数指定单位的距离测量值。例如，要获取以英里为单位的距离值，可以将 unit_name 参数指定为 mi。请参阅 “设置距离单位”语句，以获取有关可用单位名称列表。

X 和 Y 坐标参数必须使用在 Map.NumericCoordSys 中设置的坐标系。默认情况下， MapInfo 预期的坐标使用经、纬度坐标系。

如果 Map.NumericCoordSys 是地球坐标系，则 Distance( ) 将返回两点之间的大圆距离。大圆距离是球体上两点之间的最短距离。（大圆是环绕地球的圆，其圆心位于地球中心，两点之间的大圆距离是连接两点的大圆上的距离。）

如果 Map.NumericCoordSys 不是地球坐标系，则 Distance( ) 将返回迪卡尔坐标距离。

Exp( ) 函数

目的

此函数返回指定指数的数值 e。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**Exp**(num_expr)

部分	说明
num_expr	数字表达式

注解

Exp( ) 函数将数值 e 自乘由 num_expr 指定的幂次， e 的值约等于 2.7182818。

注：MapX 通过脱字运算符 (^) 支持常规幂指数。

Fix( ) 函数

目的

此函数返回整数值，通过去除小数值的小数部分获取。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[Int=]**Fix**(num_expr)

部分	说明
num_expr	数字表达式

注解

Fix( ) 函数去除数字的小数部分，返回相应的整数值。

Fix( ) 函数和 Int( ) 函数类似但不完全相同。两个函数的差异在于其处理负小数值的方式。在传递负小数时， Fix( ) 将返回大于或等于原值的最接近整数值，此时函数将调用：

```
Fix(-2.3)
```

返回的值为 -2。但是 Int( ) 函数传递负小数时，它将返回小于或等于原值的最接近整数值，此时函数将调用：

```
Int(-2.3)
```

返回的值为 ñ3。

Format$( ) 函数

目的

此函数将返回表示定制格式数字的字符串。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[string=]**Format$(**value , pattern**)**

部分	说明
value	数字表达式
pattern	指定如何格式化结果的字符串

注解

Format$( ) 函数返回表示格式化数字的字符串。例如给定的值是 **12345.67**， **Format$( )** 可以产生格式化的结果，例如 **$12,345.67**。

value 参数指定要格式化的数字值。

pattern 参数是代码字符字符串，选择用于产生特定的格式类型。pattern 字符串应该包括一个或多个特殊格式字符，例如 **#**、**0**、**%**、逗号字符、句点或分号，这些字符控制结果的外观。下表列出了格式字符。

pattern 字符	格式化结果的作用：
#	该结果将包括来自于值的一个或多个数字。 如果在 pattern 字符串中，小数点位置左侧包含一个或多个 # 字符，并且如果该值介于 0 和 1 之间，格式化的结果字符串在小数点位置之前将不会有 0。
0	数字占位符，类似于 # 字符。如果在 pattern 字符串中，小数点位置左侧包含一个或多个 **0** 字符，并且该值介于 0 和 1 之间，格式化的结果字符串在小数点位置之前将不会有 0。请参阅以下示例。
. （句点）	如果您希望结果字符串包括 “小数点分隔符”，则 pattern 字符串必须包括句点。结果字符串将包括当前在用户计算机中使用的小数点分隔符。要强制小数点分隔符为句点，可使用**设置格式**语句。
, （逗号）	如果您希望结果字符串包括 “千分位分隔符”，则 pattern 字符串必须包括逗号。结果字符串将包括当前在用户计算机中使用的千分位分隔符。要强制千分位分隔符为逗号，可使用**设置格式**语句。
%	结果将表示为该值乘以 100，例如值为 0.75 时得到的结果字符串将为 75%。如果您希望在结果中包含百分比符号，但是不希望 MapX 将该值乘以 100，则在百分比符号之前放置 \ （反斜线）字符 （如下所示）。

pattern 字符	格式化结果的作用：
E+	使用科学计数法格式化结果。例如，值 1234 得到结果为 1.234e+03。如果指数为正，则在 e 之后将会出现加号。如果指数为负 （例如值为小数时），则结果将在 e 之后包括一个减号。
E-	除了结果在 e 之后不会显示加号之外，控制字符函数的字符串和 E+ 字符串相似。
; （分号）	通过在 pattern 字符串中包含分号，您可以指定一个用于正数的格式和另一个用于负数的格式。将分号置于第一组格式字符之后，并置于第二组格式字符之前。第二组格式字符适用于负数。如果希望显示负数带有减号，则可在第二组格式字符中包含 -。
\	如果反斜线字符出现在 pattern 字符之后，则 MapX 对于反斜线之后的字符不会做出任何特殊处理。这将允许您在结果中包括特殊字符 （例如 **%** ），不会导致上述特殊格式化操作。

FormatDate$( ) 函数

目的

返回由控制面板指定的简短日期样式格式化的日期。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[string=]**FormatDate$ (** value **)**

部分	说明
value	以 YYYMMDD 格式表示日期的数字。

注解

FormatDate$( ) 函数返回的字符串以控制面板指定的本地系统格式表示日期。

Visual Basic 示例

假定控制面板设置 d/m/y 用于日期顺序，设置 ñ 用于日期分隔符，设置 dd-MMM-yyyy 用于简短日期格式：

```
Dim lyrUSA As MapXLib.Layer
Dim ds As MapXLib.Dataset
Dim ftrs As MapXLib.Features
Set lyrUSA = Map1.Layers.Add("USA")
Set ds = Map1.Datasets.Add(miDataSetDAO, Data1.Recordset, "TestDataset",
"GEOABBR", , lyrUSA, , False)

' 查找由 FormatDate$ 返回的 Oct。应该成功用于数据集中的每一记录。
Set ftrs = lyrUSA.Search("InStr(1, FormatDate$(20001011, ""Oct"") <> 0")
If ftrs.Count <> 51 Then
 Debug.print "FormatDate$ failed"
End If
```

FormatNumber$( ) 函数

目的

此函数返回的字符串表示数字，包括千分位分隔符和小数位分隔符，匹配用户系统配置。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[string=]**FormatNumber$(** num **)**

部分	说明
num	表示数值的数值或字符串，例如 1234.56

注解

此函数返回表示数字的字符串。如果数字较大，足以使用千分位分隔符，则此函数将会插入千分位分隔符。 MapInfo 读取用户的系统配置，确定将哪些字符用作千分位分隔符和小数位分隔符。

下表展示了如果用户的计算机设置将逗号用作千分位分隔符，将句点用作小数位分隔符 （美国默认设置）， **FormatNumber$( )** 函数如何工作。

函数调用	返回结果
FormatNumber$("12345.67")	"12,345.67" （插入千分位分隔符）
FormatNumber$("12,345.67")	"12,345.67" （没有改变）

如果用户的计算机设置将句点用作千分位分隔符，将逗号用作小数位分隔符，下表显示了相应的结果：

函数调用	返回结果
FormatNumber$("12345.67")	"12.345,67" （插入千分位分隔符，将小数位分隔符更改为符合用户设置）
FormatNumber$("12,345.67")	"12.345,67" （更改两个字符以匹配用户的设置）

InStr( ) 函数

目的

此函数返回字符位置，表示子字符串初次显示在另一字符串中时的位置。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[int=]**InStr**(position, string, substring)

部分	说明
position	一个正整数，表示搜索的起始位置
string	字符串表达式
substring	我们尝试在字符串中确定其位置的字符串表达式

注解

InStr( ) 函数测试字符串表达式字符串是否包含字符串表达式子字符串。 MapX 搜索字符串表达式，从 position 参数表达的位置开始，此时如果 position 参数的值为 1，则 MapX 将从字符串参数的起始位置开始搜索。

如果字符串不包含子字符串，则 InStr( ) 函数将返回的值为 0。

如果字符串包含子字符串，则 InStr( ) 函数将返回子字符串出现的字符位置。例如，如果子字符串出现在字符串的起始位置，则 InStr( ) 函数返回的值为 1。

如果子字符串参数为空字符串，则 InStr( ) 函数返回 0。

InStr( ) 函数区分大小写。换而言之， InStr( ) 函数将无法在较大的字符串 abcde 中确定子字符串 BC 的位置，因为 BC 为大写。

Int( ) 函数

目的

此函数返回整数值，通过去除小数值的小数部分获取。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[Int=]**Int**(num_expr)

部分	说明
num_expr	数字表达式

注解

Int( ) 函数返回一个最接近的整数值，小于或等于指定的 num_expr 表达式。

注意 Fix( ) 函数和 Int( ) 函数类似但不完全相同。两个函数的差异在于其处理负小数值的方式。在传递负小数时， Fix( ) 将返回大于或等于原值的最接近整数值，此时函数将调用：

```
Fix(-2.3)
```

返回的值为 -2。但是 Int( ) 函数传递负小数时，它将返回小于或等于原值的最接近整数值，此时函数将调用：

```
Int(-2.3)
```

返回的值为 -3。

LCase$( ) 函数

目的

此函数返回字符串的小写等同字符。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[string=]**LCase$**(string_expr)

部分	说明
string_expr	字符串表达式

注解

LCase$( ) 函数返回字符串表达式 string_expr 的小写等同字符。

从大写字符转换为小写字符只影响字母字符 （从 A 至 Z），数字字符和标点标记不受影响。此时函数将调用：

```
LCase$( "A#12a" )
```

返回字符串值 a#12a。

Left$( ) 函数

目的

此函数返回全部字符串的一部分，在字符串的左端开始。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[string=]**Left$**(string_expr, num_expr)

部分	说明
string_expr	字符串表达式
num_expr	数字表达式， 0 或更大的值

注解

Left$( ) 函数返回的字符串包含字符串表达式 string_expr 最左侧的 num_expr 字符。

num_expr 参数应该为整数值、 0 或更大的值。如果 num_expr 为小数，则 MapX 将圆整至最接近的整数。如果 num_expr 为 0，则 Left$( ) 将返回空字符串。如果 num_expr 参数大于 string_expr 字符串的字符数，则 Left$( ) 将返回整个 string_expr 字符串的复本。

Log( ) 函数

目的

此函数返回数值的自然对数。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**Log**(num_expr)

部分	说明
num_expr	数字表达式。

注解

Log( ) 函数返回由 num_expr 参数指定的数字表达式的自然对数。

自然对数表示为了令数字值获取 num_expr 约等于的 2.7182818， e 所需的幂级数。

对数只定义用于正数，相应的，如果 num_expr 为负，则 Log( ) 函数将会生成一个错误。

您可以使用其他底数 （例如以 10 为底）计算对数值来使用自然对数。要获取数 n 以 10 为底的对数，可以将 n 的自然对数 Log(n) 除以 10 的自然对数 Log(10)。

LTrim$( ) 函数

目的

去除字符串开始处的空格字符，并返回结果。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[string=]**LTrim**$(string_expr)

部分	说明
string_expr	字符串表达式

注解

LTrim$( ) 函数从 string_expr 字符串的起始处去除所有空格，然后返回相应的结果字符串。

Maximum( ) 函数

目的

此函数返回两个数中较大的值。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]Maximum(num_expr , num_expr)

部分	说明
num_expr	是一个数字表达式
num_expr	是一个数字表达式

注解

Maximum( ) 函数返回两个数字表达式中较大的一个。

MBR( ) 函数

目的

此函数返回矩形对象，表示其他对象的最小边界限定矩形。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[Rectangle object=]**MBR**(obj_expr)

部分	说明
obj_expr	对象表达式

注解

MBR( ) 函数计算最小边界限定矩形 （或 MBR），其中包含 obj_expr 对象。

最小边界限定矩形定义为最小矩形，其大小足以包含特定的对象。换而言之，美国的 MBR 向东延伸至缅因州的最东端，向南延伸至夏威夷最南端，向西延伸至阿拉斯加最西端，向北延伸到阿拉斯加的最北端。

点对象的 MBR 宽度和高度为 0。

Mid$( ) 函数

目的

此函数返回了取自另一字符串中部的字符串。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[string=]**Mid$**(string_expr, position, length)

部分	说明
string_expr	字符串表达式
position	数字表达式，表示字符串的起点
length	数字表达式，表示要提取的字符数

注解

Mid$( ) 函数返回源自 string_expr 字符串的子字符串表达式。

Mid$( ) 从 string_expr 字符串由 position 参数表示的字符位置复制长度字符。如果是小于等于 1 的正数，则 MapX 将从字符 string_expr string 的最开始处开始复制。

如果 string_expr 字符串的长度不够，则可能没有要复制的长度字符。此时，取决于所有参数， Mid$( ) 有可能会返回字符串长度字符长。如果位置 position 表示的数字大于 string_expr 中的字符数，Mid$( ) 将返回空字符串。如果 length 参数为 0，则 Mid$( ) 将返回空字符串。如果 length 或 position 参数为小数，MapX 将圆整至最接近的整数。

Minimum( ) 函数

目的

此函数返回两个数中较小的值。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**Minimum**(num_expr , num_expr)

部分	说明
num_expr	是一个数字表达式
num_expr	是一个数字表达式

注解

Minimum( ) 函数返回两个数字表达式中较小的一个。

Month( ) 函数

目的

此函数将返回日期数字的月份部分 (1 - 12)。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[SmallInt value from 1 to 12, inclusive=]**Month**(date_expr)

注解

Month( ) 函数返回一个整数，代表指定日期的月份部分 (1 - 12)。

ObjectLen( ) 函数

目的

此函数返回线或多义线对象的地理长度。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**ObjectLen**(expr , unit_name)

部分	说明
expr	一个对象表达式
unit_name	表示距离单位名称的字符串 （例如 mi 代表英里）。

注解

ObjectLen( ) 函数返回对象表达式的长度。请注意只有线或多义线对象的长度值大于 0，要测量矩形、椭圆或地区周长，可使用 Perimeter( ) 函数。

ObjectLen( ) 函数返回使用 unit_name 参数指定单位的长度测量值。例如，要获取以英里为单位的长度，可以将 unit_name 参数指定为 mi。请参阅 “设置距离单位”语句，以获取有效单位名称列表。

ObjectType( ) 函数

目的

返回对象 (GeoObject.h) 的类型。

- POINT= 1
- LINE = 2
- ARC = 3
- POLYLINE = 4
- REGION = 5
- TEXT= 6
- RECT= 7
- ROUNDRECT = 8
- ELLIPSE = 9

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

ObjectType(object)

部分	说明
object	对象表达式。

返回值

正数

Perimeter( ) 函数

目的

此函数返回图形对象的周长。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**Perimeter**(obj_expr , unit_name)

部分	说明
obj_expr	一个对象表达式

部分	说明
unit_name	表示距离单位名称的字符串 （例如 km）

注解

Perimeter( ) 函数计算 obj_expr 对象的周长。 Perimeter( ) 函数定义用于以下对象类型：椭圆、矩形、圆整矩形和多边形。其他类型的对象的周长测量值为 0。

Perimeter( ) 函数返回使用 unit_name 参数指定单位的长度测量值。例如，要获取以英里为单位的长度，可以将 unit_name 参数指定为 mi。请参阅 “设置距离单位”语句，以获取有效单位名称列表。

用于圆整矩形时， Perimeter( ) 函数返回近似值。 MapX 将圆整矩形视为常规矩形对象计算其周长。

Proper$( ) 函数

目的

此函数将返回混合大小写的字符串，其中只有每个单词的第一个字母是大写的。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[string=]**Proper$**(string_expr)

部分	说明
string_expr	字符串表达式

注解

Proper$( ) 函数先将整个 string_expr 字符串转换为小写，然后将字符串的每个单词的第一个字符大写，即可产生带有 “正确”结果的字符串。这一大写样式适用于固有名称。

Right$( ) 函数

目的

此函数返回全部字符串的一部分，在字符串的右端开始。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[string=]**Right$**(string_expr, num_expr)

部分	说明
string_expr	字符串表达式
num_expr	数字表达式

注解

Left$( ) 函数返回的字符串包含字符串表达式 string_expr 最右侧的 num_expr 字符。

num_expr 参数应该为整数值、 0 或更大的值。如果 num_expr 为小数，则 MapX 将圆整至最接近的整数。如果 num_expr 为 0，则 Right$( ) 将返回空字符串。如果 num_expr 参数大于 string_expr 字符串的字符数，则 Right$( ) 将返回整个 string_expr 字符串的复本。

Round( ) 函数

目的

此函数将返回通过圆整另一个数字获取的值。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**Round**(num_expr, round_to)

部分	说明
num_expr	数字表达式。
round_to	num_expr 应该圆整到的值。

注解

Round( ) 函数返回数字 num_expr 表达式的圆整版本。

结果的精度取决于 round_to 参数。 Round( ) 函数将 num_expr 值圆整至 round_to 参数的最接近倍数。如果 round_to 为 0.01， MapInfo 将圆整至最接近的百分数；如果 round_to 为 5，则 MapInfo 将圆整至 5 的最接近倍数等。

RTrim$( ) 函数

目的

去除字符串结束处的空格字符，并返回结果。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[string=]**RTrim$**(string_expr)

部分	说明
string_expr	字符串表达式

注解

RTrim$( ) 函数从 string_expr 字符串的结尾处去除所有空格，然后返回相应的结果字符串。

Sgn( ) 函数

目的

此函数返回 -1、 0 或 1，表示指定数字是负数、 0 或正数 （分别）。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**Sgn**(num_expr)

部分	说明
num_expr	数字表达式

注解

如果 num_expr 小于 0，则 Sgn( ) 函数返回的值为 -1 ；如果 num_expr 为 0，则 Sgn( ) 函数返回的值为 0 ；如果 num_expr 大于 0，则 Sgn( ) 函数返回的值为 1。

Sin( ) 函数

目的

此函数返回数值的正弦值。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**Sin**(num_expr)

部分	说明
num_expr	数字表达式

注解

Sin( ) 函数返回 num_expr 数字值的正弦值，是以弧度表示的角度。从 Sin( ) 返回的结果将介于 1 和 -1 之间。

要将角度值转换为弧度值，需要将该值乘以 π /ISO。要将弧度值转换为角度值，需要将该值乘以 ISO/ π 。

Space$( ) 函数

目的

此函数返回只包含空格的字符串。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[string=]**Space$**(num_expr)

部分	说明
num_expr	数字表达式

注解

Space$( ) 函数返回字符串 num_expr 字符长度，完全由空格字符构成。

如果 num_expr 的值小于或等于 0， Space$( ) 函数将返回空字符串。

Sqr( ) 函数

目的

此函数将返回数字的平方根。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**Sqr**(num_expr)

部分	说明
num_expr	数字表达式

注解

Sqr( ) 函数返回由 num_expr 指定的数字表达式的平方根。由于平方根运算不适用于负实数，因此 num_expr 应该表示大于或等于 0 的值。

取一个数字的平方根等于求数字的 0.5 次幂。相应的， Sqr(n) 等同于表达式 n ^ 0.5 ；但是 Sqr( ) 函数计算的平方根更快。

StringCompare( ) 函数

目的

执行区分大小写的字符串比较

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

StringCompare(string1, string2 **)**

string1 和 string2 是字符串表达式

返回值

SmallInt: 如果第一个字符串超出第二个字符串则为 -1 ；如果第一个字符串在第二个字符串之后则为 1 ；如果两个字符相等则为 0

注解

StringCompare( ) 函数执行区分大小写的字符串比较。 MapX 字符串比较使用 = 运算符区分大小写。比较表达式如下所示：

```
If  "ABC" = "abc" Then
```

求解为 TRUE，因为字符串比较是区分大小写的。

StringCompare( ) 函数执行区分大小写的字符串比较，返回如何比较字符串的表示。

返回值：	当：
-1	第一个字符串的字母排序在第二个字符串之前
0	这两个字符串相等
1	第一个字符串的字母排序在第二个字符串之后

StringCompareIntl( ) 函数

目的

执行区分语言的字符串比较。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

StringCompareIntl(string1 , string2 **)**

string1 和 string2 是正在比较的字符串表达式

返回值

SmallInt: 如果第一个字符串超出第二个字符串则为 -1 ；如果第一个字符串在第二个字符串之后则为 1 ；如果两个字符相等则为 0

注解

StringCompareIntl( ) 函数执行区分语言的字符串比较。如果需要确定两个字符串的字母顺序，以及包含普通美国字符集之外 （例如日耳曼语）的字符时，可调用此函数。

比较使用当前在用户计算机中使用的语言设置。例如， Windows 用户可以通过控制面板来控制语言设置。

返回值：	当：
-1	使用当前的语言的设置，第一个字符串在第二个字符串之前
0	这两个字符串相等
1	使用当前的语言的设置，第一个字符串在第二个字符串之后

StringToDate( ) 函数

目的

StringToDate( ) 函数返回 Date 值，给定表示日期的字符串。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

StringToDate (datestring)

datestring 是表示日期的 String 表达式，例如 10/17/96

返回值

日期

注解

MapX 根据用户的计算机上设置的日期格式选项来解释日期字符串。美国计算机通常配置的格式日期为“月 / 日 / 年”，但其他国家的计算机常常会使用其他顺序 （例如 “日 / 月 / 年”），或其他分隔符字符（例如使用句点来替代 /）。字符串中的 m/d/y 序列需要匹配 Windows 系统的简短日期格式设置。如果设置为 MM/DD/YY，则以上示例 51/01/01 将会无效，因为不存在第 51 月。

要强制 StringToDate( ) 函数应用美国格式规范，可使用设置格式语句。

注：当应用美国格式规范时， datestring 参数必须使用正斜线字符 (/) 来间隔日期的日、月和年部分。

datestring 参数必须表示月份 （1 - 12，表示为一个或两个数字），以及每月的日期 （1 - 31，表示为一个或两个数字）。

您可以指定四位数字的年份，也可以完全省略年份。如果没有指定年份， Mapx 将使用当前年份。

注：MapInfo Corporation 建议使用 4 位数字格式的年份用于日期变量或日期字符串。@ 位的年份将返回意外的结果。

如果您使用带有日期类型的 MapX 变量对象，而且您提供了 2 位数字的表示，则开发环境 （而非 MapX）将确定世纪。 StringToDate 只将字符串变量作为参数。它将 MapX 日期类型变量。

Str$( ) 函数

目的

此函数将返回表示表达式的字符串 （例如数字的打印输出）。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[string=]**Str$**(expression)

部分	说明
expression	数字、日期、墨笔、画笔、符号、字体、逻辑或对象表达式。

注解

Str$( ) 函数返回表示指定表达式的值的字符串。

如果表达式为负数，则返回的字符串的第一个字符为减号 (-)。如果表达式为正数，则字符串的第一个字符为空格。

取决于指定的表达式中的数字精度位数，并且取决于小数点左侧的位数， Str$( ) 函数可能返回表示圆整值的字符串。如果需要控制字符串中显示的数字精度位数，可使用 Format$( ) 函数。

如果表达式是对象表达式，则 Str$( ) 函数将返回字符串，表示对象类型：弧、椭圆、框架、线、点、多义线、矩形、区域、圆整矩形或文本。

如果表达式是表单 tablename.obj 的对象表达式，并且源自该表的当前行没有附属的图形对象， Str$( ) 返回空字符串。注：向 Str$( ) 函数传递未初始化的 Object 变量将会产生错误。

如果表达式为日期，则 Str$( ) 的输出将取决于用户的计算机配置。如果表达式是数字，则 Str$( ) 函数将使用句点作为小数分隔符，即使用户的计算机设置其他字符作为小数分隔符。 Str$( ) 函数在返回字符串中从不包括千分位分隔符。要生成使用用户指定的千分位分隔符和小数分隔符的字符串，可使用 FormatNumber$( ) 函数。

Tan( ) 函数

目的

此函数返回数值的正切值。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]**Tan**(num_expr)

部分	说明
num_expr	数字表达式

注解

Tan( ) 函数返回 num_expr 数字值的正切值，是以弧度表示的角度。

要将角度值转换为弧度值，需要将该值乘以 π /ISO。要将弧度值转换为角度值，需要将该值乘以 ISO/ π 。

UCase$( ) 函数

目的

此函数返回转换为大写字符的字符串。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[string=]**UCase$**(string_expr)

部分	说明
string_expr	字符表达式

注解

UCase$( ) 函数返回字符串表达式 string_expr 的大写等同字符。

从小写字符转换为大写字符只影响字母字符 （从 A 至 Z），数字字符和标点标记不受影响。此时函数将调用：

UCase$("A#12a")

返回字符串值 A#12A。

Val( ) 函数

目的

返回由字符串表示的数字值。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[float=]Val(string_expr)

部分	说明
string_expr	字符表达式

注解

Val( ) 函数基于 string_expr 字符串表达式返回一个数字。 Val( ) 忽略 string_expr 字符串开始部分的空白（制表符、空格、回行），然后尝试将第一个字符解释为数字值。 Val( ) 函数在发现字符不是数字的组成部分时，将停止处理字符串。

如果字符串中的第一个非空白字符不是句点、数字、减号或 “和”字符 (&)， Val( ) 将返回 0。（“和”符号用于十六进制符号，请参阅以下示例。）

注：如果字符串包含小数分隔符，则必须为句点，无论用户的计算机是否设置其他字符作为小数分隔符。此外，字符串不能包含千分位分隔符。要从数字字符串移除千分位分隔符，可调用 DeformatNumber$( ) 函数。

Weekday( ) 函数

目的

此函数返回从 1 至 7 的正数，表示指定日期是星期几。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[SmallInt value from 1 to 7, inclusive; 1 represents Sunday=]**Weekday**(date_expr)

部分	说明
date_expr	日期表达式

注解

Weekday( ) 函数返回表示指定日期是星期几的部分 （从 1 至 7）。

Weekday( ) 函数只适用于 100 年中 1 月 1 日以及其后的日期。如果 date_expr 指定 100 年前的日期，Weekday( ) 函数返回的值为 0。

Year( ) 函数

目的

此函数将返回日期值的年份部分。

注：MapX 中的函数用于创建表达式并传递为参数。相应参数只能传递到 **Layer.Search** 和 **DataSet.AddField** 方法。

语法

[SmallInt value from 1 to 7, inclusive; 1 represents Sunday=]**Year**(date_expr)

部分	说明
date_expr	日期表达式

注解

Year( ) 函数返回一个整数，表示指定日期的年份部分。例如，如果指定日期为 12/17/93，则 Year( ) 函数返回的值为 1993。

附录 I: Geoset 关键字

Geoset 是包含有关一组图层信息的文件 （扩展名为 .GST），这些文件可以一次加载。 geoset 在设计阶段（和属性相同）执行加载，或使用 Layer 对象的 **AddGeosetLayers 方法**加载。 geoset 文件是一个包含字符串的 ASCII 文件。这些字符串包含关键字和值。这些关键字相应于 MapX 中的属性 － 主 Map 对象以及 Layer 对象的属性。

关键字自然继承，通过引号引起来的字符串指定。关键字值也是引号引起来的值，即使是数字也要使用引号。以下显示部分关键字和值的示例：

附录 J: 示例 Geoset

```
!GEOSET
!VERSION 100
begin_metadata
"\GEOSET" = ""
"\GEOSET\NAME" = "SAMPLE GEOSET"
"\GEOSET\PROJECTION" = "3,62,7,-96,23,20,60,0,0"
"\GEOSET\CENTER" = "-54851.35483414936,1844196.997419479"
"\GEOSET\MBR" = ""
"\GEOSET\MBR\LOWERLEFT" = "-3093309.705881681,-450646.671927353"
"\GEOSET\MBR\UPPERRIGHT" = "2983606.996213382,4139040.666766311"
"\GEOSET\ZOOMLEVEL" = "4019.82"
"\GEOSET\AUTOLAYER" = "FALSE"
"\GEOSET\MAPUNIT" = "0"
"\GEOSET\ROTATION" = "0"
"\TABLE" = ""
"\TABLE\1" = ""
"\TABLE\1\FILE" = "usa_caps.tab"
"\TABLE\1\DESCRIPTION" = "US Capitals"
"\TABLE\1\ISVISIBLE" = "TRUE"
"\TABLE\1\AUTOLABEL" = "FALSE"
"\TABLE\1\DRAWLABELSAFTER" = "FALSE"
"\TABLE\1\SHOWLINEDIRECTION" = "FALSE"
"\TABLE\1\SHOWNODES" = "FALSE"
"\TABLE\1\SHOWCENTROIDS" = "FALSE"
"\TABLE\1\EDITABLE" = "FALSE"
"\TABLE\1\SELECTABLE" = "TRUE"
"\TABLE\1\REGISTERINGEOODICT" = "TRUE"
"\TABLE\1\DISPLAY" = ""
"\TABLE\1\DISPLAY\BRUSH" = ""
"\TABLE\1\DISPLAY\BRUSH\Pattern" = "2"
"\TABLE\1\DISPLAY\BRUSH\Forecolor" = "16777215"
```

```
"\TABLE\1\DISPLAY\BRUSH\Backcolor" = "16777215"
"\TABLE\1\DISPLAY\BRUSH\Transparent" = "FALSE"
"\TABLE\1\DISPLAY\PEN" = ""
"\TABLE\1\DISPLAY\PEN\LineWidth" = "1"
"\TABLE\1\DISPLAY\PEN\LineStyle" = "1"
"\TABLE\1\DISPLAY\PEN\Pattern" = "2"
"\TABLE\1\DISPLAY\PEN\Color" = "0"
"\TABLE\1\DISPLAY\LINEPEN" = ""
"\TABLE\1\DISPLAY\LINEPEN\LineWidth" = "1"
"\TABLE\1\DISPLAY\LINEPEN\LineStyle" = "1"
"\TABLE\1\DISPLAY\LINEPEN\Pattern" = "2"
"\TABLE\1\DISPLAY\LINEPEN\Color" = "0"
"\TABLE\1\DISPLAY\FONT" = ""
"\TABLE\1\DISPLAY\FONT\Style" = "3"
"\TABLE\1\DISPLAY\FONT\ExtStyle" = "1"
"\TABLE\1\DISPLAY\FONT\Description" = "Arial"
"\TABLE\1\DISPLAY\FONT\Size" = "36"
"\TABLE\1\DISPLAY\FONT\Forecolor" = "0"
"\TABLE\1\DISPLAY\FONT\Backcolor" = "16777215"
"\TABLE\1\DISPLAY\FONT\Opaque" = "FALSE"
"\TABLE\1\DISPLAY\SYMBOL" = ""
"\TABLE\1\DISPLAY\SYMBOL\Type" = "1"
"\TABLE\1\DISPLAY\SYMBOL\Code" = "35"
"\TABLE\1\DISPLAY\SYMBOL\Color" = "255"
"\TABLE\1\DISPLAY\SYMBOL\Pointsize" = "14"
"\TABLE\1\DISPLAY\SYMBOL\Font" = ""
"\TABLE\1\DISPLAY\SYMBOL\Font\Style" = "0"
"\TABLE\1\DISPLAY\SYMBOL\Font\ExtStyle" = "0"
"\TABLE\1\DISPLAY\SYMBOL\Font\Description" = "Map Symbols"
"\TABLE\1\DISPLAY\SYMBOL\Font\Size" = "14"
"\TABLE\1\DISPLAY\SYMBOL\Font\Forecolor" = "255"
"\TABLE\1\DISPLAY\SYMBOL\Font\Backcolor" = "16777215"
"\TABLE\1\DISPLAY\SYMBOL\Font\Opaque" = "FALSE"
"\TABLE\1\DISPLAY\SYMBOL\Font\Rotation" = "0"
```

```
"\TABLE\1\LABEL" = ""
"\TABLE\1\LABEL\FONT" = ""
"\TABLE\1\LABEL\FONT\Style" = "0"
"\TABLE\1\LABEL\FONT\ExtStyle" = "0"
"\TABLE\1\LABEL\FONT\Description" = "Arial"
"\TABLE\1\LABEL\FONT\Size" = "9"
"\TABLE\1\LABEL\FONT\Forecolor" = "0"
"\TABLE\1\LABEL\FONT\Backcolor" = "16777215"
"\TABLE\1\LABEL\FONT\Opaque" = "FALSE"
"\TABLE\1\LABEL\DUPLICATE" = "TRUE"
"\TABLE\1\LABEL\PARALLEL" = "TRUE"
"\TABLE\1\LABEL\OVERLAP" = "FALSE"
"\TABLE\1\LABEL\PARTIALSEGMENTS" = "FALSE"
"\TABLE\1\LABEL\LINETYPE" = "2"
"\TABLE\1\LABEL\OFFSET" = "2"
"\TABLE\1\LABEL\POSITION" = "5"
end_metadata
```

前 11 行显示由 GEOSET 为起始字的关键字 － 这些关键字设置整个地图的属性，或 Map 对象的属性。请注意某些关键字，例如 GEOSET\MBR 是多级关键字，有 LOWERLEFT 和 UPPERRIGHT 两级。

接下来的 13 行显示了 usa_caps 这一图层的关键字设置属性。这些关键字的起始词为 TABLE。紧接该起始词的词是指特定图层的属性集的任意关键字，此时它们采用了和表文件名相同的词。接下来是用于指定图层的层次关键字值。

附录 K: 支持的关键字

geoset 文件中支持的关键字列表如下：

\GEOSET\	
NAME	geoset 的友好名称
PROJECTION	投影子句
CENTER	数字 , 数字 – 地图中心点经纬度。
\MBR\LOWERLEFT	数字 , 数字 – 左下角的经纬度。
\MBR\UPPERRIGHT	数字 , 数字 – 右上角的经纬度。
ZOOMLEVEL	数字 – 地图的缩放级别
AUTOLAYER	TRUE 或 FALSE （1 或 0） – 图层自动排序
MAPUNIT	数字 – 相应于 Map.MapUnit。当 geoset 加载到地图控件中时，如果没有已经加载的图层，则 MapUnit 将设置为该 geoset 的 MapUnit。但是，如果已有加载的图层， MapX 将加载 geoset 并使用当前的地图单位。
ROTATION	数字 – 地图的旋转角度相应于 Map.Rotation

\TABLE\<number>\	
FILE	字符串 – .TAB 文件的位置
DESCRIPTION	字符串 – 表的说明
ISVISIBLE	TRUE 或 FALSE （1 或 0） – 图层是否可见
AUTOLABEL	TRUE 或 FALSE （1 或 0） – 图层是否自动标记
DRAWLABELSAFTER	TRUE 或 FALSE （1 或 0） – 在绘制图层之后是否绘制标签
SHOWLINEDIRECTION	TRUE 或 FALSE （1 或 0） – 相应于 Layer.ShowLineDirection
SHOWNODES	TRUE 或 FALSE （1 或 0） – 相应于 Layer.ShowNodes
SHOWCENTROIDS	TRUE 或 FALSE （1 或 0） – 相应于 Layer.ShowCentroids
EDITABLE	TRUE 或 FALSE （1 或 0） – 相应于 Layer.Editable
SELECTABLE	TRUE 或 FALSE （1 或 0） – 相应于 Layer.Selectable
REGISTERINGEODICT	TRUE 或 FALSE （1 或 0） – 是否在 geodictionary 之中注册 geoset

\TABLE\<number>\	
ZOOM\MIN	数字 – 将缩放值最小化以显示图层 (Layer.ZoomMin)
ZOOM\MAX	数字 – 将缩放值最大化以显示图层 (Layer.ZoomMax)

DISPLAY\BRUSH\	
PATTERN	数字 – 相应于 Style.RegionPattern
FORECOLOR	数字 – 相应于 Style.RegionColor
BACKCOLOR	数字 – 相应于 Style.RegionBackColor
TRANSPARENT	数字 – 相应于 Style.RegionTransparent

DISPLAY\PEN\	
LINEWIDTH	数字 – 相应于 Style.RegionBorderWidth
LINESTYLE	数字 – 相应于 Style.RegionBorderStyle
COLOR	数字 – 相应于 Style.RegionBorderColor

DISPLAY\LINEPEN\	
LINEWIDTH	数字 – 相应于 Style.LineWidth
LINESTYLE	数字 – 相应于 Style.LineStyle
COLOR	数字 – 相应于 Style.LineColor

DISPLAY\FONT\	
STYLE	数值 – 位掩码 (BOLD 0x01, ITALIC 0x02, UNDER 0x04, STRIKEOUT 0x08, OUTLINE 0x10, SHADOW 0x20, INVERSE 0x40, BLINK 0x80)
EXTSTYLE	数字 – 位掩码 (HALO 0x01, ALLCAPS 0x02, DBLSPACE 0x04)
DESCRIPTION	字符串 – 字体名称
SIZE	数字 – 字体大小 （磅）
FORECOLOR	数字 – 相应于 Style.TextFontColor
BACKCOLOR	数字 – 相应于 Style.TextFontBackColor

DISPLAY\FONT\	
OPAQUE	TRUE 或 FALSE － 相应于 Style.TextFontOpaque

DISPLAY\SYMBOL\	
TYPE	数字 － 相应于 Style.SymbolType
CODE	数字 － 相应于 Style.SymbolCharacter
COLOR	数字 － 相应于 Style.SymbolFontColor
POINTSIZE	数字 － 符号大小 （磅）

TrueType 字体符号关键字

FONT	符号所用的字体。如果符号类型为 miSymbolTypeTrueTypeFont，则将会出现此关键字。它具有一个相同的子关键字 DISPLAY\FONT
FONT\ROTATION	字体符号的旋转角度。相应于 Style.SymbolFontRotation。

BitmapSymbol 关键字

BITMAP\NAME	字符串 － 位图符号的名称。相应于 Style.SymbolBitmapName。
BITMAP\OVERRIDECOLOR	TRUE 或 FALSE － 是否对符号应用颜色。相应于 Style.SymbolBitmapOverrideColor
BITMAP\TRANSPARENT	TRUE 或 FALSE － 位图是否透明。相应于 Style.SymbolBitmapTransparent。

LABEL\	
ZOOM\MIN	数字 － 将缩放值最小化至标签图层
ZOOM\MAX	数字 － 将缩放值最大化至标签图层
FONT	这是标签字体。它具有一个相同的子关键字 DISPLAY\FONT （如上所示）
DUPLICATE	TRUE 或 FALSE － 允许标签重复
PARALLEL	TRUE 或 FALSE － 相应于 LabelProperties.Parallel
OVERLAP	TRUE 或 FALSE － 相应于 LabelProperties.Overlap
PARTIALSEGMENTS	TRUE 或 FALSE － 相应于 LabelProperties.PartialSegments

LABEL\	
MAXLABELS	数字 – 相应于 LabelProperties.LabelMax
LINETYPE	数字 – 相应于 LabelProperties.LineType
OFFSET	数字 – 相应于 LabelProperties.Offset
POSITION	数字 – 相应于 LabelProperties.Position

附录 L: OLE_COLOR 值

OLE_COLOR 值是 BGR （蓝、绿、红）值。要确定 BGR 值，请使用以下公式指定蓝色、绿色或红色的值 （每种颜色取值从 0 至 255）：

```
BGR 值 = (蓝色 * 65536) + (绿色 * 256) + 红色
```

注：如果您已经使用 MapBasic 编程语言，请注意 MapBasic 使用以下公式计算 RGB 值：

```
BGR 值 = (红色 * 65536) + (绿色 * 256) + 蓝色
```

如果您正在使用 RGB 颜色值 （例如用于 MapBasic 应用程序），在 MapX 中使用这些值之前，您需要将颜色转换为 BGR。

另请参阅

ColorConstants

附录 M: Idispatch 表

此表列出了 MapX 的 IDispatch。

方法 / 属性	IDispatch
AffineTransform.A	2
AffineTransform.B	3
AffineTransform.C	4
AffineTransform.D	5
AffineTransform.E	6
AffineTransform.F	7
AffineTransform.Set	8
AffineTransform.Units	1
Annotation.Graphic	2
Annotation.Type	1
Annotations.ActiveAnnotation	8
Annotations.AddSymbol	3
Annotations.AddText	6
Annotations.Count	2
Annotations.Editable	1
Annotations.Item	4
Annotations.Remove	5
Annotations.RemoveAll	7
BindLayer.CoordSys	6
BindLayer.Filespec	7
BindLayer.KeyLength	8
BindLayer.LayerName	1
BindLayer.LayerType	4
BindLayer.RefColumn1	2
BindLayer.RefColumn2	3
BindLayer.ReferenceLayer	5
BindLayer.ReferenceLayerField	9

方法 / 属性	IDispatch
BitmapSymbol.Name	1
BitmapSymbols.Count	1
BitmapSymbols.Item	2
BitmapSymbols.Refresh	3
BitmapSymbols.Unload	4
CoordSys.AffineTransform	14
CoordSys.Azimuth	9
CoordSys.Bounds	4
CoordSys.Clone	17
CoordSys.Datum	2
CoordSys.FalseEasting	11
CoordSys.FalseNorthing	12
CoordSys.OriginLatitude	6
CoordSys.OriginLongitude	5
CoordSys.PickCoordSys	16
CoordSys.Range	13
CoordSys.ScaleFactor	10
CoordSys.Set	15
CoordSys.StandardParallelOne	7
CoordSys.StandardParallelTwo	8
CoordSys.Type	1
CoordSys.Units	3
DataSet.AddField	14
DataSet.Fields	3
DataSet.GeoField	5
DataSet.Layer	7
DataSet.Name	1
DataSet.ReadOnly	12
DataSet.Refresh	8
DataSet.RowCount	2
DataSet.RowValues	13

方法 / 属性	IDispatch
DataSet.SecondaryGeoField	6
DataSet.SourceRows	10
DataSet.Themes	4
DataSet.Type	11
DataSet.Value	9
DataSets.Add	2
DataSets.BuildSourceRows	7
DataSets.Count	1
DataSets.Item	3
DataSets.Remove	4
DataSets.RemoveAll	6
DataSets.Restore	5
Datum.Eccentricity	13
Datum.Ellipsoid	1
Datum.Flattening	12
Datum.PrimeMeridian	9
Datum.RotateX	5
Datum.RotateY	6
Datum.RotateZ	7
Datum.ScaleAdjust	8
Datum.SemiMajorAxis	10
Datum.SemiMinorAxis	11
Datum.Set	14
Datum.SetFromList	15
Datum.ShiftX	2
Datum.ShiftY	3
Datum.ShiftZ	4
Feature.Area	10
Feature.Attach	20
Feature.Bounds	7
Feature.Caption	8

方法 / 属性	IDispatch
Feature.CenterX	4
Feature.CenterY	5
Feature.Clone	21
Feature.FeatureID	1
Feature.FeatureKey	23
Feature.HasMultipoint	29
Feature.HasPolyline	28
Feature.HasRegion	27
Feature.Multipoint	26
Feature.Polyline	25
Feature.Region	24
Feature.KeyValue	17
Feature.LabelPoint	13
Feature.Layer	11
Feature.Length	3
Feature.Name	16
Feature.Nodes	22
Feature.Offset	19
Feature.Parts	9
Feature.Perimeter	14
Feature.Point	12
Feature.Smooth	15
Feature.Style	6
Feature.Type	2
Feature.Update	18
FeatureFactory.BufferFeatures	1
FeatureFactory.CombineFeatures	2
FeatureFactory.CreateArc	9
FeatureFactory.CreateCircularRegion	11
FeatureFactory.CreateEllipticalRegion	10
FeatureFactory.CreateLine	7

方法 / 属性	IDispatch
FeatureFactory.CreateCollectionFeature	14
FeatureFactory.CreateMultipoint	15
FeatureFactory.CreateRegion	6
FeatureFactory.CreateSymbol	5
FeatureFactory.CreateText	8
FeatureFactory.EraseFeature	4
FeatureFactory.IntersectFeatures	3
FeatureFactory.IntersectionPoints	13
FeatureFactory.IntersectionTest	12
Features.Add	4
Features.AddByID	9
Features.Bounds	8
Features.Clone	3
Features.Common	5
Features.Count	1
Features.Item	2
Features.Remove	6
Features.RemoveByID	10
Features.Replace	7
Field.AggregationFunction	3
Field.Name	2
Field.Type	1
Fields.Add	2
Fields.Count	1
Fields.Item	3
Fields.Remove	4
Fields.RemoveAll	5
Find.Abbreviations	2
Find.CloseMatchMax	10
Find.ClosestAddr	3
Find.FindDataSet	7

方法 / 属性	IDispatch
Find.FindField	5
Find.OtherBoundary	4
Find.RefineDataSet	8
Find.RefineField	6
Find.RefineLayer	1
Find.Search	9
Find.SearchEx	11
FindFeature.Area	65546
FindFeature.Bounds	65543
FindFeature.Caption	65544
FindFeature.CenterX	65540
FindFeature.CenterY	65541
FindFeature.FeatureID	65537
FindFeature.FindRC	1
FindFeature.KeyValue	65553
FindFeature.LabelPoint	65549
FindFeature.Layer	65547
FindFeature.Length	65539
FindFeature.Name	65552
FindFeature.Offset	65555
FindFeature.Parts	65545
FindFeature.Perimeter	65550
FindFeature.Point	65548
FindFeature.Smooth	65551
FindFeature.Style	65542
FindFeature.Type	65538
FindFeature.Update	65554
FindMatch.FeatureID	1
FindMatch.FeatureKey	4
FindMatch.Name	2
FindMatch.Score	3

方法 / 属性	IDispatch
FindMatches.Count	1
FindMatches.Item	2
FindResult.AddressOutOfRange	3
FindResult.ExactMatch	1
FindResult.FindRC	7
FindResult.IntersectionNotFound	4
FindResult.MatchedFeature	9
FindResult.Matches	8
FindResult.MultipleMatches	5
FindResult.RefineRegion	6
FindResult.Substitute	2
Geoset.Centroid	1
Geoset.PathName	2
Geoset.UserName	3
Geosets.Count	1
Geosets.Item	2
Graphic.Caption	1
Graphic.Position	4
Graphic.Style	5
Graphic.X	2
Graphic.Y	3
IndividualValueCategories.AllOthersCategory	3
IndividualValueCategories.Count	1
IndividualValueCategories.Item	2
IndividualValueCategory.NumItems	2
IndividualValueCategory.Style	1
IndividualValueCategory.Value	3
LabelProperties.DataField	13
LabelProperties.DataSet	9
LabelProperties.Duplicate	10
LabelProperties.LabelMax	12

方法 / 属性	IDispatch
LabelProperties.LabelZoom	3
LabelProperties.LabelZoomMax	5
LabelProperties.LabelZoomMin	4
LabelProperties.LineType	6
LabelProperties.Offset	7
LabelProperties.Overlap	11
LabelProperties.Parallel	8
LabelProperties.PartialSegments	15
LabelProperties.Position	14
LabelProperties.Style	1
LabelProperties.Visible	2
Layer.AddFeature	21
Layer.AllFeatures	26
Layer.AutoLabel	6
Layer.BeginAccess	41
Layer.Bounds	32
Layer.ClearCustomLabels	19
Layer.CoordSys	17
Layer.DataSets	39
Layer.DeleteFeature	22
Layer.DrawLabelsAfter	37
Layer.DrillDownAddFeatures	31
Layer.DrillDownRemoveFeatures	30
Layer.DrilldownReset	34
Layer.Editable	46
Layer.EndAccess	42
Layer.FeatureIDFromFeatureName	38
Layer.FeatureKeyFromFeatureName	48
Layer.FileSpec	4
Layer.Find	14
Layer.GetDrilldownFeaturesByID	33

方法 / 属性	IDispatch
Layer.GetFeatureByID	35
Layer.GetFeatureByKey	47
Layer.Invalidate	23
Layer.KeyField	16
Layer.LabelAtPoint	18
Layer.LabelProperties	5
Layer.Name	1
Layer.NoFeatures	27
Layer.OverrideStyle	8
Layer.PredominantFeatureType	13
Layer.Refresh	36
Layer.Search	40
Layer.SearchAtPoint	29
Layer.SearchWithinDistance	24
Layer.SearchWithinFeature	28
Layer.SearchWithinRectangle	25
Layer.Selectable	3
Layer.Selection	12
Layer.ShowCentroids	44
Layer.ShowLineDirection	45
Layer.ShowNodes	43
Layer.Style	9
Layer.Type	15
Layer.UpdateFeature	20
Layer.Visible	2
Layer.ZoomLayer	7
Layer.ZoomMax	11
Layer.ZoomMin	10
LayerInfo.AddParameter	2
LayerInfo.Type	1
Layers.Add	4

方法 / 属性	IDispatch
Layers.AddGeoSetLayers	5
Layers.AddServerLayer	13
Layers.AddUserDrawLayer	8
Layers.AnimationLayer	9
Layers.Bounds	12
Layers.ClearSelection	2
Layers.Count	1
Layers.CreateLayer	10
Layers.InsertionLayer	15
Layers.Item	6
Layers.LayersDlg	11
Layers.Move	3
Layers.Remove	7
Layers.RemoveAll	14
Legend.BodyTextStyle	6
Legend.Compact	1
Legend.CompactTitle	11
Legend.CompactTitleStyle	5
Legend.CurrencyFormat	8
Legend.ExportLegend	18
Legend.Height	15
Legend.Left	12
Legend.LegendDlg	16
Legend.LegendTexts	7
Legend.PaperHeight	21
Legend.PaperWidth	22
Legend.PrintLegend	20
Legend.ShowCount	19
Legend.ShowEmptyRanges	17
Legend.SubTitle	10
Legend.SubTitleStyle	4

方法 / 属性	IDispatch
Legend.Title	9
Legend.TitleStyle	3
Legend.Top	13
Legend.Visible	2
Legend.Width	14
LegendText.Text	1
LegendText.Visible	2
LegendTexts.AllOthersText	4
LegendTexts.AutoGenerate	1
LegendTexts.Count	2
LegendTexts.Item	3
Map.AboutBox	-552
Map.Annotations	3
Map.AreaUnit	28
Map.AutoRedraw	6
Map.BackColor	-501
Map.Bounds	29
Map.CenterX	8
Map.CenterY	9
Map.ClipLine	44
Map.ClipLineV	45
Map.ConvertCoord	34
Map.ConvertCoordV	41
Map.CreateCustomTool	36
Map.CurrentTool	20
Map.DataSet	4
Map.DataSetGeoField	1
Map.DataSetTheme	19
Map.DataSets	22
Map.DefaultConversionResolution	50
Map.DefaultStyle	17

方法 / 属性	IDispatch
Map.DisplayCoordSys	30
Map.Distance	37
Map.ExportMap	39
Map.ExportSelection	32
Map.FeatureFactory	49
Map.GeoDictionary	18
Map.GeoSet	10
Map.GeoSetWidth	11
Map.Geosets	27
Map.hWnd	-515
Map.InfotipPopupDelay	61
Map.InfotipSupport	63
Map.IsPointVisible	43
Map.Layers	5
Map.MapPaperHeight	12
Map.MapPaperWidth	13
Map.MapScreenHeight	54
Map.MapScreenWidth	53
Map.MapUnit	25
Map.MatchNumericFields	52
Map.MatchThreshold	57
Map.MaxSearchTime	14
Map.MouseIcon	62
Map.MousePointer	21
Map.MousewheelSupport	51
Map.NumericCoordSys	31
Map.Pan	64
Map.PanAnimationLayer	60
Map.PaperUnit	15
Map.PreferCompactLegends	16
Map.PrintMap	38

方法 / 属性	IDispatch
Map.PropertyPage	40
Map.RedrawInterval	55
Map.Refresh	-550
Map.Rotation	26
Map.SaveMapAsGeoset	47
Map.SearchPath	56
Map.SelectionStyle	33
Map.SetSize	42
Map.Title	23
Map.TitleText	24
Map.Version	2
Map.WaitCursorEnabled	58
Map.Zoom	7
Map.ZoomTo	35
MultivarCategories.Count	1
MultivarCategories.Item	2
MultivarCategory.Style	1
NotesQueryInfo.BeginDate	4
NotesQueryInfo.Database	2
NotesQueryInfo.DefaultNumericValue	8
NotesQueryInfo.DefaultStringValue	7
NotesQueryInfo.EndDate	5
NotesQueryInfo.FullTextSearch	6
NotesQueryInfo.MaxNumDocs	9
NotesQueryInfo.Query	3
NotesQueryInfo.Server	1
NotesViewInfo.Database	2
NotesViewInfo.Server	1
NotesViewInfo.View	3
ODBCQueryInfo.ConnectString	3
ODBCQueryInfo.DataSource	1

方法 / 属性	IDispatch
ODBCQueryInfo.SqlQuery	2
Parts.Add	4
Parts.Count	1
Parts.Item	2
Parts.Remove	3
Parts.RemoveAll	5
Point.Offset	4
Point.Set	3
Point.X	1
Point.Y	2
Points.Add	4
Points.AddXY	5
Points.Count	1
Points.Item	2
Points.Remove	3
Points.RemoveAll	6
RangeCategories.AllOthersCategory	3
RangeCategories.Count	1
RangeCategories.Item	2
RangeCategory.Max	2
RangeCategory.Min	1
RangeCategory.NumItems	3
RangeCategory.Style	4
Rectangle.Height	5
Rectangle.Offset	8
Rectangle.Set	7
Rectangle.Width	6
Rectangle.XMax	3
Rectangle.XMin	1
Rectangle.YMax	4
Rectangle.YMin	2

方法 / 属性	IDispatch
ResolveObject.SourceMatch	2
ResolveObject.TableMatch	3
ResolveObject.TableName	1
ResolveObjects.Add	4
ResolveObjects.Count	1
ResolveObjects.Item	2
ResolveObjects.Remove	3
ResolveObjects.RemoveAll	5
RowValue.DataSet	2
RowValue.Field	3
RowValue.ReadOnly	1
RowValue.Value	4
RowValues.Add	4
RowValues.Clone	6
RowValues.Count	1
RowValues.Item	2
RowValues.ReadOnly	7
RowValues.Remove	3
RowValues.RemoveAll	5
Selection.Add	65540
Selection.ClearSelection	2
Selection.Clone	65539
Selection.Common	65541
Selection.Count	65537
Selection.Item	65538
Selection.Remove	65542
Selection.Replace	65543
Selection.SelectAll	6
Selection.SelectByID	7
Selection.SelectByPoint	3
Selection.SelectByRadius	4

方法 / 属性	IDispatch
Selection.SelectByRectangle	5
Selection.SelectByRegion	1
SourceRow.Row	1
SourceRows.Count	1
SourceRows.Item	2
Style.Clone	41
Style.DrawLineSample	32
Style.DrawRegionSample	31
Style.DrawSymbolSample	30
Style.DrawTextSample	29
Style.ExportLineSample	51
Style.ExportRegionSample	50
Style.ExportSymbolSample	49
Style.ExportTextSample	52
Style.LineColor	2
Style.LineInterleaved	44
Style.LineStyle	3
Style.LineStyleCount	42
Style.LineSupportsInterleave	47
Style.LineWidth	4
Style.LineWidthUnit	45
Style.MaxVectorSymbolCharacter	56
Style.MinVectorSymbolCharacter	55
Style.PickLine	26
Style.PickRegion	25
Style.PickSymbol	28
Style.PickText	27
Style.RegionBackColor	7
Style.RegionBorderColor	9
Style.RegionBorderStyle	8
Style.RegionBorderWidth	10

方法 / 属性	IDispatch
Style.RegionBorderWidthUnit	46
Style.RegionColor	5
Style.RegionPattern	6
Style.RegionTransparent	43
Style.SupportsBitmapSymbols	39
Style.SymbolBitmapColor	35
Style.SymbolBitmapName	36
Style.SymbolBitmapOverrideColor	34
Style.SymbolBitmapSize	38
Style.SymbolBitmapTransparent	33
Style.SymbolCharacter	1
Style.SymbolFont	12
Style.SymbolFontBackColor	16
Style.SymbolFontColor	15
Style.SymbolFontHalo	19
Style.SymbolFontOpaque	17
Style.SymbolFontRotation	40
Style.SymbolFontShadow	20
Style.SymbolType	37
Style.SymbolVectorColor	53
Style.SymbolVectorSize	54
Style.TextFont	11
Style.TextFontAllCaps	23
Style.TextFontBackColor	14
Style.TextFontColor	13
Style.TextFontDblSpace	24
Style.TextFontHalo	21
Style.TextFontOpaque	18
Style.TextFontRotation	48
Style.TextFontShadow	22
Theme.AutoRecompute	5

方法 / 属性	IDispatch
Theme.ComputeTheme	13
Theme.DataMax	12
Theme.DataMin	11
Theme.Fields	10
Theme.Layer	9
Theme.Legend	4
Theme.Name	6
Theme.Properties	3
Theme.ThemeDlg	8
Theme.ThemeProperties	7
Theme.Type	2
Theme.Visible	1
ThemeProperties.AllowEmptyRanges	15
ThemeProperties.ApplyAttribute	31
ThemeProperties.BarFrameStyle	28
ThemeProperties.BarFramed	38
ThemeProperties.BarGraduatedStack	30
ThemeProperties.BarIndependentScale	26
ThemeProperties.BarStacked	20
ThemeProperties.BarWidth	25
ThemeProperties.BorderStyle	24
ThemeProperties.ColorMethod	34
ThemeProperties.DataValue	10
ThemeProperties.DistMethod	1
ThemeProperties.DotColor	16
ThemeProperties.DotSize	5
ThemeProperties.GraduateSizeBy	23
ThemeProperties.Graduated	6
ThemeProperties.Independent	8
ThemeProperties.IndividualValueCategories	4
ThemeProperties.InflectRanges	35

方法 / 属性	IDispatch
ThemeProperties.InflectionColor	37
ThemeProperties.InflectionRange	36
ThemeProperties.MultivarCategories	7
ThemeProperties.NegativeSymbolStyle	21
ThemeProperties.NumRanges	2
ThemeProperties.PieClockwise	17
ThemeProperties.PieGraduated	29
ThemeProperties.PieHalfPies	18
ThemeProperties.PieStartAngle	19
ThemeProperties.PositiveSymbolStyle	27
ThemeProperties.RangeCategories	3
ThemeProperties.RoundBy	33
ThemeProperties.RoundRanges	32
ThemeProperties.ShowNegativeValues	22
ThemeProperties.Size	11
ThemeProperties.SpreadBy	14
ThemeProperties.SymbolStyle	13
ThemeProperties.ValuePerDot	9
ThemeProperties.Width	12
Themes.Add	2
Themes.Count	1
Themes.Item	3
Themes.Remove	4
Themes.RemoveAll	5
Title.Border	8
Title.Caption	1
Title.Editable	6
Title.Position	7
Title.TextStyle	2
Title.Visible	3
Title.X	4

方法 / 属性	IDispatch
Title.Y	5

附录 N: 管理 MapX 数据

除了 MapX 软件之外， MapX 安装程序还在您的计算机上安装数据产品 （地图和人口统计数据），以便您可以即刻体验 MapX 的功能。

注：数据产品的使用收到 MapInfo 标准许可协议的约束 – 软件和数据。

默认情况下，地图和 geoset 均安装在以下目录：

- C:\Program Files\MapInfo\MapX 5.0\Maps

人口统计数据，采用 Microsoft Access 表的形式 (MAPSTATS.MDB)，安装在以下目录中：

- C:\Program Files\MapInfo\MapX 5.0\Data

安装在计算机上的确切文件列表部分取决于在安装时所选择的选项。（默认情况下，所有数据产品均安装，但是安装程序允许您取消选择任意或所有 geoset，以便避免安装相应文件。）安装文件的列表还取决于安装的软件版本， MapX 安装 CD 提供了比下载评估版软件更全面的数据产品。

使用 MapX Geodictionary Manager 来注册有关与 MapInfo 表有关的信息，这些 MapX 表可以在自动数据绑定期间匹配。默认情况下，这一实用程序位于以下路径：

- C:\Program Files\Common Files\MapInfo Shared\MapX Common\GeoDictionaryManager40.EXE

使用 MapX Geoset Manager 来创建和管理 geoset。默认情况下，这一实用程序位于以下路径：

- C:\Program Files\Common Files\MapInfo Shared\MapX Common\GeosetManager40.EXE

有关 Geodictionary Manager 的详细信息，请参阅**使用 GeoDictionary Manager**。有关 Geoset Manager 的详细信息，请参阅**使用 Geoset Manager**。

附录 O: 数据源

以下来源提供了 MapInfo MapX 发布的数据。有关 MapInfo 产品的详细信息，请参阅 “MapInfo 数据产品目录”或与 MapInfo Corporation 联系。

地图名称	版权所有
日本 Geoset	
日本城区公路	GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
日本城市	MapInfo 取自 Digital Chart of the World
日本国家背景	GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
日本高速公路	GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
日本主要城市	MapInfo 取自 Digital Chart of the World
日本河流和湖泊	GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
世界海洋 （纬度 / 经度）	MapInfo
澳大利亚 Geoset	
澳大利亚城市	MapInfo 取自 Digital Chart of the World
澳大利亚高速公路	MapInfo 取自 Digital Chart of the World
澳大利亚主要城市	MapInfo 取自 Digital Chart of the World
澳大利亚州界	MapInfo
澳大利亚州府城市	MapInfo 取自 Digital Chart of the World
世界海洋 （纬度 / 经度）	MapInfo
欧洲 Geoset	
亚洲	MapInfo 取自 Digital Chart of the World
欧洲首都	MapInfo 取自 Digital Chart of the World
欧洲城市	MapInfo 取自 Digital Chart of the World
欧洲国界	URPI 1998 制作和数据版权所有， GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993

地图名称	版权所有
欧洲高速公路	GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
欧洲主要城市	MapInfo 取自 Digital Chart of the World
欧洲 NUTS 1 级边界	版权所有 URPI 1998
欧洲 NUTS 2 级边界	版权所有 URPI 1998
世界海洋 （纬度 / 经度）	MapInfo
法国 Geoset	
欧洲国界	URPI 1998 制作和数据版权所有， GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
法国城市	MapInfo 取自 Digital Chart of the World
法国高速公路地图	GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
法国主要城市	MapInfo 取自 Digital Chart of the World
法国 NUTS 2 级行政边界	版权所有 URPI 1998
世界海洋 （纬度 / 经度）	MapInfo
英国 Geoset	
欧洲国界	URPI 1998 制作和数据版权所有， GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
英国城市	MapInfo 取自 Digital Chart of the World
英国 A 级公路	GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
英国高速公路	GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
英国标准区域	版权所有 URPI 1998
世界海洋 （纬度 / 经度）	MapInfo
加拿大 Geoset	
加拿大城市	MapInfo 取自 Digital Chart of the World
加拿大高速公路	MapInfo 取自 Digital Chart of the World
加拿大主要城市	MapInfo 取自 Digital Chart of the World
加拿大省界	MapInfo 取自 Digital Chart of the World

地图名称	版权所有
加拿大省会城市	MapInfo 取自 Digital Chart of the World
美国州界	MapInfo 取自 Digital Chart of the World
世界海洋 （纬度 / 经度）	MapInfo
美国 Geoset	
加拿大省界	MapInfo 取自 Digital Chart of the World
墨西哥州界	MapInfo 取自运输统计局
美国城市	MapInfo 取自 Digital Chart of the World
美国城镇边界	MapInfo 取自人口统计局
美国高速公路	MapInfo 取自运输统计局
美国主要城市	MapInfo 取自 Digital Chart of the World
美国州界	MapInfo 取自 Digital Chart of the World
美国州府	MapInfo 取自 Digital Chart of the World
美国 20 大城市	MapInfo 取自 Digital Chart of the World
世界海洋 （纬度 / 经度）	MapInfo
世界 Geoset	
各国首都	MapInfo 取自 Digital Chart of the World
世界各国	MapInfo 取自 Digital Chart of the World
世界网格	MapInfo
世界海洋 (Robinson)	MapInfo
世界 25 大城市	MapInfo 取自 Digital Chart of the World
哥伦比亚特区 Geoset	
哥伦比亚特区界标	MapInfo
哥伦比亚特区城市边界	MapInfo
哥伦比亚特区高速公路	MapInfo 取自运输统计局
哥伦比亚特区区内公路	MapInfo 取自运输统计局
哥伦比亚特区界标	MapInfo
哥伦比亚特区公路	MapInfo 取自运输统计局
哥伦比亚特区州际公路	MapInfo 取自运输统计局

地图名称	版权所有
哥伦比亚特区水域图层	MapInfo
哥伦比亚特区邮政区域	© 1995 Geographic Data Technology, Inc.
亚洲 Geoset	
亚洲	MapInfo 取自 Digital Chart of the World
亚洲首都	MapInfo 取自 Digital Chart of the World
亚洲主要城市	MapInfo 取自 Digital Chart of the World
欧洲国界	URPI 1998 制作和数据版权所有， GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
世界海洋 （亚洲）	MapInfo
中大西洋地区 Geoset	
中大西洋地区首都	MapInfo 取自 Digital Chart of the World
中大西洋地区城市	MapInfo 取自 Digital Chart of the World
中大西洋地区城镇	MapInfo
中大西洋地区高速公路	MapInfo 取自运输统计局
中大西洋地区主要城市	MapInfo 取自 Digital Chart of the World
中大西洋地区的州	MapInfo 取自 Digital Chart of the World
世界海洋 （纬度 / 经度）	MapInfo
美国州界	MapInfo 取自 Digital Chart of the World
中国 Geoset	
亚洲	MapInfo 取自 Digital Chart of the World
亚洲主要城市	MapInfo 取自 Digital Chart of the World
中国城市	MapInfo 取自 Digital Chart of the World
中国国界	MapInfo 取自 Digital Chart of the World
中国高速公路	MapInfo 取自 Digital Chart of the World
中国主要城市	MapInfo 取自 Digital Chart of the World
世界海洋 （亚洲）	MapInfo
墨西哥 Geoset	
墨西哥城市	MapInfo 取自 Digital Chart of the World
墨西哥高速公路	MapInfo 取自 Digital Chart of the World

地图名称	版权所有
墨西哥主要城市	MapInfo 取自 Digital Chart of the World
墨西哥州界	MapInfo 取自运输统计局
墨西哥州府城市	MapInfo 取自 Digital Chart of the World
美国州界	MapInfo 取自 Digital Chart of the World
世界海洋 （纬度 / 经度）	MapInfo
北美 Geoset	
加拿大城市	MapInfo 取自 Digital Chart of the World
加拿大高速公路	MapInfo 取自 Digital Chart of the World
加拿大主要城市	MapInfo 取自 Digital Chart of the World
加拿大省界	MapInfo 取自 Digital Chart of the World
加拿大省会城市	MapInfo 取自 Digital Chart of the World
墨西哥城市	MapInfo 取自 Digital Chart of the World
墨西哥高速公路	MapInfo 取自 Digital Chart of the World
墨西哥主要城市	MapInfo 取自 Digital Chart of the World
墨西哥州界	MapInfo 取自运输统计局
墨西哥州府城市	MapInfo 取自 Digital Chart of the World
美国城市	MapInfo 取自 Digital Chart of the World
美国高速公路	MapInfo 取自运输统计局
美国主要城市	MapInfo 取自 Digital Chart of the World
美国州界	MapInfo 取自 Digital Chart of the World
美国州府	MapInfo 取自 Digital Chart of the World
世界海洋 （纬度 / 经度）	MapInfo
德国 Geoset	
欧洲国界	URPI 1998 制作和数据版权所有， GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
德国城市	MapInfo 取自 Digital Chart of the World
德国高速公路	GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
德国主要城市	MapInfo 取自 Digital Chart of the World

地图名称	版权所有
德国 NUTS 2 级边界	MapInfo 取自 Digital Chart of the World
世界海洋 （纬度 / 经度）	MapInfo
意大利 Geoset	
欧洲国界	URPI 1998 制作和数据版权所有， GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
意大利城市	MapInfo 取自 Digital Chart of the World
意大利高速公路	GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
意大利主要城市	MapInfo 取自 Digital Chart of the World
意大利 NUTS 2 级边界	版权所有 URPI 1998
世界海洋 （纬度 / 经度）	MapInfo
葡萄牙 Geoset	
欧洲国界	URPI 1998 制作和数据版权所有， GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
葡萄牙高速公路	GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
葡萄牙主要城市	MapInfo 取自 Digital Chart of the World
葡萄牙 NUTS 2 级边界	版权所有 URPI 1998
世界海洋 （纬度 / 经度）	MapInfo
阿根廷 Geoset	
阿根廷城市	MapInfo 取自 Digital Chart of the World
阿根廷国界	MapInfo 取自 Digital Chart of the World
阿根廷主要城市	MapInfo 取自 Digital Chart of the World
南美国界	MapInfo 取自 Digital Chart of the World
世界海洋 （纬度 / 经度）	MapInfo
巴西 Geoset	
巴西主要城市	MapInfo 取自 Digital Chart of the World
巴西城市	MapInfo 取自 Digital Chart of the World
巴西国界	MapInfo 取自 Digital Chart of the World

地图名称	版权所有
南美国界	MapInfo 取自 Digital Chart of the World
世界海洋 （纬度 / 经度）	MapInfo
以色列 Geoset	
非洲国界	MapInfo 取自 Digital Chart of the World
亚洲	MapInfo 取自 Digital Chart of the World
以色列城市	MapInfo 取自 Digital Chart of the World
以色列国界	MapInfo 取自 Digital Chart of the World
以色列主要城市	MapInfo 取自 Digital Chart of the World
世界海洋 （纬度 / 经度）	MapInfo
韩国 Geoset	
亚洲	MapInfo 取自 Digital Chart of the World
韩国城市	MapInfo 取自 Digital Chart of the World
韩国国界	MapInfo 取自 Digital Chart of the World
韩国主要城市	MapInfo 取自 Digital Chart of the World
世界海洋 （亚洲）	MapInfo
西班牙 Geoset	
欧洲国界	URPI 1998 制作和数据版权所有， GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
西班牙城市	MapInfo 取自 Digital Chart of the World
西班牙高速公路	GISdata Limited 授权 MapInfo 使用 GisNET 数据。 © GDC Ltd 1993
西班牙主要城市	MapInfo 取自 Digital Chart of the World
西班牙 NUTS 2 级边界	版权所有 URPI 1998
世界海洋 （纬度 / 经度）	MapInfo
印度 Geoset	
亚洲	MapInfo 取自 Digital Chart of the World
印度首都城市	MapInfo 取自 Digital Chart of the World
印度区界	Risk Management Solutions, Inc
印度主要城市	MapInfo 取自 Digital Chart of the World

地图名称	版权所有
印度小城市	MapInfo 取自 Digital Chart of the World
印度省界	Risk Management Solutions, Inc
世界海洋 （亚洲）	MapInfo
美国详细 Geoset	
加拿大省界	MapInfo 取自 Digital Chart of the World
界标地图	MapInfo 取自 USGS
墨西哥州界	MapInfo 取自运输统计局
美国城市	MapInfo 取自 Digital Chart of the World
美国城镇边界	MapInfo 取自人口统计局
美国高速公路	MapInfo 取自运输统计局
美国主要城市	MapInfo 取自 Digital Chart of the World
美国州界	MapInfo 取自 Digital Chart of the World
美国州府	MapInfo 取自 Digital Chart of the World
美国 20 大城市	MapInfo 取自 Digital Chart of the World
美国邮政编码边界	© 1997 Geographic Data Technology, Inc.
世界海洋 （纬度 / 经度）	MapInfo
世界详细 Geoset	
各国首都	MapInfo 取自 Digital Chart of the World
世界各国	MapInfo 取自 Digital Chart of the World
世界网格	MapInfo
世界主要城市	MapInfo 取自 Digital Chart of the World
世界小城市	MapInfo 取自 Digital Chart of the World
世界海洋 (Robinson)	MapInfo
世界 25 大城市	MapInfo 取自 Digital Chart of the World
德克萨斯达拉斯 Geoset	
犯罪统计城镇地图	MapInfo
犯罪统计地图	MapInfo
达拉斯市界 A	MapInfo 取自 USGS

地图名称	版权所有
达拉斯市界 B	MapInfo 取自 USGS
达拉斯位置	MapInfo
达拉斯主要高速公路 A	MapInfo 取自 USGS
达拉斯主要高速公路 B	MapInfo 取自 USGS
达拉斯街道 A	MapInfo 取自 USGS
达拉斯街道 B	MapInfo 取自 USGS
达拉斯街道 C	MapInfo 取自 USGS
达拉斯水域	MapInfo 取自 USGS
达拉斯水域河流	MapInfo 取自 USGS
德克萨斯达拉斯光栅	版权所有 © 1997, Horizons Technology, Inc
其他文件	
达拉斯市界取自 MapXsite Streets 1.0	MapInfo 取自 USGS
达拉斯主要高速公路取自 MapXsite Streets 1.0	MapInfo 取自 USGS
达拉斯街道取自 MapXsite Streets 1.0	MapInfo 取自 USGS
达拉斯水域取自 MapXsite Streets 1.0	MapInfo 取自 USGS
达拉斯水域取自 MapXsite Streets 1.0	MapInfo 取自 USGS
美国 5 位数字邮政编码点	(c) 1999 Geographic Data Technology, Inc.
美国 5 位数字邮政编码点 （压缩点文件）	(c) 1995 Geographic Data Technology, Inc.
MapStats.mdb	
亚洲人口统计状况	美国人口统计年鉴 1994 版权所有
澳大利亚人口统计状况	MapInfo 澳大利亚 :
哥伦比亚特区人口统计状况	版权所有 1998, The Polk Company, All Rights Reserved
中大西洋地区的州的人口统计状况	版权所有 1998, The Polk Company, All Rights Reserved
世界人口统计状况	美国人口统计年鉴 1994 版权所有
美国人口统计状况	版权所有 1998, The Polk Company, All Rights Reserved
美国城镇人口年龄统计状况	版权所有 1998, The Polk Company, All Rights Reserved
美国城镇人口统计年龄状况 （性别）	版权所有 1998, The Polk Company, All Rights Reserved
美国城镇家庭统计状况	版权所有 1998, The Polk Company, All Rights Reserved

地图名称	版权所有
美国城镇家庭统计状况 （年龄、收入）	版权所有 1998, The Polk Company, All Rights Reserved
美国城镇家庭收入	版权所有 1998, The Polk Company, All Rights Reserved
美国城镇住房价值	版权所有 1998, The Polk Company, All Rights Reserved
美国城镇人口统计状况	版权所有 1998, The Polk Company, All Rights Reserved
美国客户数据库	MapInfo

附录 P: 使用 Geoset Manager

使用 Geoset Manager 可以很容易地保留供您使用的地图图层及其设置的集合。Geoset 可以帮助您避免在每次要处理图层时要分别打开和显示这些图层的耗时的工作。

计算机地图被组织成多个图层。将图层视作在顶部彼此堆叠的透明层。通过打开地图来开始建立一个 geoset。每一地图图层都包含不同的地图对象，例如区域、点、线条和文本。

例如，一个图层可以包含州边界，第二个图层可以具有表示首都的符号，第三个图层可以包含文本标签。通过将这些图层堆叠到其它图层的顶部，就可以开始构建完整的地图。

一旦您创建了自己的 geoset 后，就可以定制图层显示的方式，并且可以添加、删除图层或对它们进行重新排序。

从 Geoset Manager，使用示例 geoset，或者创建您自己的 geoset。

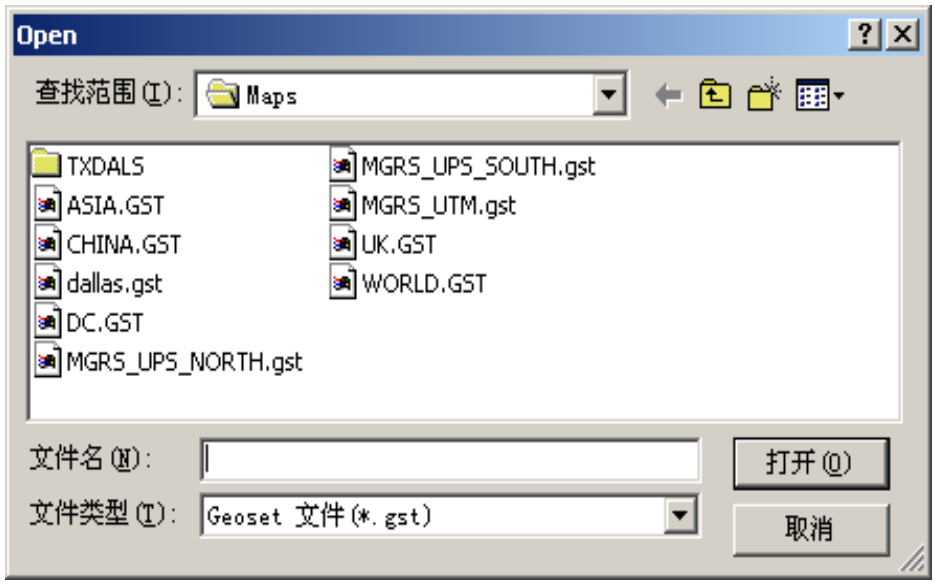
若要启动 Geoset Manager，从 “开始”菜单上的 “MapX 程序组”中选择 “Geoset Manager”。

注：在 MapX 的试用版中， Geoset Manager 将只能在开发人员的计算机上使用。

打开现有的 Geoset

MapX 包括一些示例 geoset。您可以在它们的基础上建立自己的 geoset，也可以照原样使用它们。要显示现有 geoset：

1. 选择 “文件” > “打开 Geoset”。将显示 “打开”对话框。

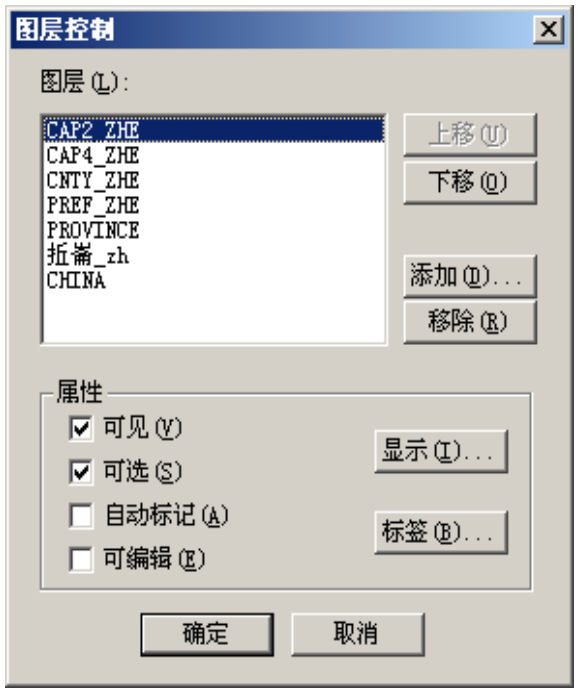


2. 选择您所需的 geoset 并单击 “打开”。将显示该 geoset。

从显示的对话框中，您可以使用 “地图” 菜单或 “视图” 菜单中的项更改各种图层设置。您还可以插入其它 geoset 以随当前 geoset 一起显示。然后，将这些新设置保存到当前 geoset 或通过 “另存为” 命令创建新的 geoset。我们将在随后的几节中详细介绍这些项。

创建 Geoset

1. 选择 “文件” > “新建 Geoset”。将显示 “图层控制” 对话框。
2. 单击 **“添加”** 将显示 “打开” 对话框。
3. 选择您要用作 geoset 的一部分的图层。按住 Ctrl 键的同时进行选择可以标记多个图层。这些选定的图层将显示在 “图层控制” 对话框中。



从该对话框中，您可以设置显示和标签属性，重新排序图层显示的方式，删除或添加其它图层并设置图层是否可见，包含自动标签，或者将图层设为可选的。请参见下一节中的图层控制的说明。您可以在显示初始 geoset 后在以后编辑这些属性。

4. 单击 “确定” 可显示该 geoset。将显示您新建的 geoset。

使用后面几节中提供的信息可以控制和定制 geoset 中的图层。在您创建好 geoset 后，需要保存它。

要保存 geoset：

1. 选择 “文件” > “保存”。将显示 “另存为” 对话框。
2. 输入名称，然后单击 “保存”，将用您指定的属性保存这些图层。

使用 “地图”菜单控制图层

使用 “地图”菜单中提供的选项可以控制您的地图显示的方式。

放大

使用 “放大”工具来获得地图或图层的更近的区域视图。

1. 选择 “地图” > “放大”或单击菜单栏上的 “放大”按钮，或者在 Geoset Manager 窗口中单击鼠标右键并选择 “放大”。

   将显示 “放大”鼠标图标。

2. 在您要放大的区域的中心单击 “放大”光标，通过两倍的线性因子放大该区域。该点在放大的视图中将是地图的中心。重复此步骤，直到您具有适当的放大级别为止。

要放大矩形区域：

1. 选择 “地图” > “放大”或单击菜单栏上的 “放大”按钮，或者在 Geoset Manager 窗口中单击鼠标右键并选择 “放大”。

   将显示 “放大”鼠标图标。

2. 通过对角拖动 “放大”鼠标图标在地图或布局中拖出一个矩形。该矩形内的区域将被放大。

缩小

使用 “缩小”工具来获得地图或图层的更宽的区域视图。

1. 选择 “地图” > “缩小”或单击 “缩小”按钮，或者在 Geoset Manager 窗口中单击鼠标右键并选择 “缩小”。

   将显示 “缩小”鼠标图标。

2. 在您要缩小的区域的中心单击 “缩小”鼠标图标，通过两倍的线性因子扩大该区域。该点在缩小的视图中将是地图的中心。重复此步骤，直到您具有适当的缩放级别为止。

要缩小矩形区域：

1. 选择 “地图” > “缩小”或单击 “缩小”按钮，或者在 Geoset Manager 窗口中单击鼠标右键并选择 “缩小”。

   将显示 “缩小”鼠标图标。

2. 通过对角拖动 “缩小”鼠标图标在地图或布局中拖出一个矩形。矩形内的区域被缩小，这样可以显示地图的更多内容。

平移

使用 “平移”来重新确定地图在其窗口中的位置。

要移动或调整地图显示：

1. 选择 “地图” > “平移”或单击 “平移”按钮，或者在 Geoset Manager 窗口中单击鼠标右键并选择 “平移”。

   将显示 “平移”鼠标图标。

2. 单击该地图的一个区域。

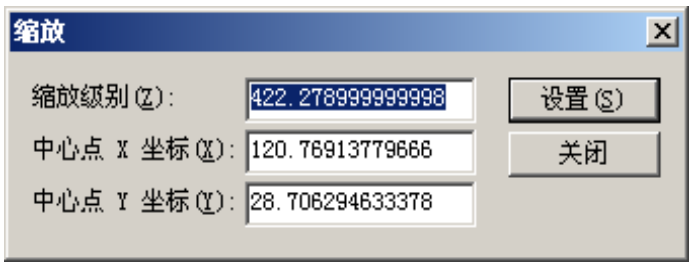
3. 在按住鼠标左键的同时，沿适当的方向拖动该地图。在松开鼠标按键后， Geoset Manager 将在其新位置中重新绘制该地图。

缩放为

在地图上缩放为特定的 X 和 Y 坐标并设置缩放级别。

选择 “地图” > “缩放为”。

将显示 “缩放为”对话框。



输入您要缩放为的坐标。

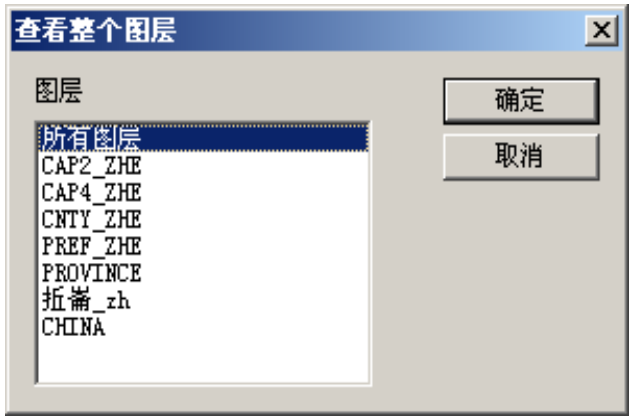
查看整个图层

使用 “查看整个图层”可以缩放并显示地图中的整个图层或所有图层。

如果地图包含覆盖不同地域范围的图层，则使用此选项。例如，您可以具有包含纽约郡、高速公路、ZIP 编码和尤蒂卡城的街道的地图。如果您选择 “所有图层”，将会缩小视图以显示整个地图。但是，如果您只是对查看尤蒂卡城的街道感兴趣，则选择尤蒂卡城的街道图层。则视图将会放大以显示这些街道。

要显示整个地图或地图图层：

1. 选择 “地图” > “查看整个图层”，或者使用鼠标右键在 “地图”窗口上右击并选择 “查看整个图层”。将显示 “查看整个图层”对话框。



2. 选择要显示的具体图层或所有图层。单击 “确定”将查看图层。

图层控制

使用 “图层控制”对话框可以：

- 更改地图图层在活动窗口中的显示
- 确定显示、删除、添加、成为可选择、缩放分层、加标签和设置哪些图层
- 更改地图图层的顺序。

要访问 “图层控制”：

- 选择 “地图” > “图层控制”，或者单击 “图层控制”按钮，或者使用鼠标右键在地图上单击并选择 “图层控制”或创建新 geoset。

“图层控制”对话框控制地图的显示方式。一些重要信息包括：

- “图层控制”对话框在当前地图窗口中显示图层的列表，并指示每一图层是否显示、可选择或自动加标签。
- 图层包括数据表、光栅图或在地图窗口中堆叠的主题地图。
- 图层在 “图层控制”对话框中的顺序就是图层在地图窗口中的顺序。例如，在边界图层放置于点图层之下时，这些点保持可见。
- 若要使用一个图层，在该图层上单击来选择它。通过在适当的框中选取或清除可以控制其设置，以显示图层、使图层可选择或自动给图层加标签。

部分	说明
上移	上移一个或多个图层。
下移	下移一个或多个图层。
添加	向地图添加一个或多个图层。从表中显示的列表框中选择。
删除	从地图中删除一个或多个图层。
可见	指示图层是否可见。为您要在地图中保持可见的一个或多个图层选取此复选框。有时候，您可能要在 geoset 中包括多个图层以用于计算目的，并且不希望它们在 geoset 中实际显示。
可选择	指示图层是否可选择。为您要成为可选择的一个或多个图层选取此复选框。如果您要选择多个对象或给多个对象加标签，则图层必须是可选择的。要想使一个图层成为可选择的，必须显示该图层。可以同时使多个图层成为可选择的。但您只能一次从一个图层上选择。
自动加标签	选取此选项可以自动给地图加标签。这些标签取自在 “标签选项”对话框的 “标签”部分中指定的表列。
显示	这将显示 “显示选项”对话框。使用该对话框可为地图图层指定显示属性。
标签	这将显示 “标签属性”对话框。使用该对话框可为地图图层指定标签属性。

若要为地图图层指定显示属性，从 “图层控制”对话框中单击 “显示”按钮。

显示模式

部分	说明
覆盖样式	选取该选项后将覆盖图层的默认样式。
样式按钮	在选取了 “样式覆盖”选项后启用。这将显示适合于所选图层中包含的对象类型的按钮：线条　　符号　　文本 î°£。单击可以显示适当的对话框。

缩放分层

部分	说明
在缩放内显示	选取该选项可以激活缩放分层。通过缩放分层，您可以设置选定的图层将可见的最小和最大距离。例如，如果您想在近于 3 英里时看到地图上的特定街道，则将最小缩放比例设置为 0 并将最大缩放比例设置为 3。
最小缩放比例	指定选定图层可见的最小距离。
最大缩放比例	指定选定图层可见的最大距离。

自动加标签

要使用该图层中的信息给地图中的图层自动加标签：

1. 从 “图层控制”对话框中，选择您要加标签的图层；然后选取 “自动加标签”复选框。
2. 单击 “确定”。

如果装载了数据集并且指定了该数据集的字段，则地图通过来自 “标签选项”对话框的 “标签”部分中指定的表列的标签重新显示。

一次每个表只显示一列。

在 “标签属性”对话框中，可以更改标签的可见性、内容、字体、文本颜色、线条样式和位置。

“标签属性”对话框

“标签属性”对话框确定标签的可见性、内容、字体、文本颜色、线条样式和锚点。这些设置用于使用 “标签”工具自动加标签和以交互的方式加标签。在使用 “标签”工具时，标签放置于您通过单击鼠标按键指定的位置。

要访问 “标签属性”对话框：

1. 选择 “地图” > “图层控制”。
2. 单击 “标签”按钮。将显示 “标签属性”对话框。

部分	说明
标签用于：	选择您要在标签中反映的数据集名和字段名。
可见性：	
启用	取消该选项将允许显示标签。

部分	说明
禁用	选取该选项将禁止显示标签。
在范围内显示	选取该选项后可以激活缩放加标签。通过缩放加标签，您可以设置标签可见的最小和最大距离。例如，如果您只想在近于 3 英里时看到地图上的特定标签，则将最小缩放比例设置为 0 并将最大缩放比例设置为 3。标签大小不会随着缩放或比例的更改而更改。
最小缩放比例	指定标签可见的最小距离。
最大缩放比例	指定标签可见的最大距离。
允许重复文本	选取该选项可以多次在地图上放置相同的标签。
允许覆盖文本	允许显示具有相同文本的多个对象。
最大标签	输入将显示的最大标签数；标签是从指定的表中按它们在该表中输入的顺序选择的。例如，如果指定的表是 States 表并且您输入 10，则在该表中 （在视图中）列出的前 10 个州将加标签。
样式：	
“文本样式”按钮	单击 “文本样式”按钮将显示 “文本样式”对话框。请参见 “文本样式”对话框。
标签线条	选择要将标签添加到锚点的线条类型，或者没有任何线条类型。
无	不显示具有标签的线条。
简单	通过使用将标签连接到对象的矩心的简单线条来创建标注。标签线条将在您从最初创建的地方移动标签后显示。
箭头	通过使用将标签连接到对象的矩心的箭头线条来创建标注。标签线条将在您从最初创建的地方移动标签后显示。
位置：	
锚点	单击一个图标可以选择相对于标签锚点的标签位置。菱形字符表示标签锚点；矩形表示标签。选定框的边框是粗体。
使用线条旋转标签	选取该选项可以旋转文本以保持与线段平行。对于点区域将忽略此设置。
标签偏移量	指定标签应距锚点放置的点数 （文本大小的度量单位）。

投影

如何展平地球的曲面以便您可以在扁平的纸张或 （接近）扁平的计算机屏幕上绘制地图？您可以使用投影。投影是定义如何使对象扁平的系统。您可以以多种不同的投影来显示地图。

在您将对象从球形世界转换为相对扁平的计算机屏幕时，注定会有一些扭曲。

当来自球面的对象在平面上显示时，投影是一种减少扭曲发生的方法。有多种不同类型的投影，每一类型均设计为减少给定区域的扭曲量。

要选择一种投影：

选择 “地图” > “投影”，或者在地图上单击鼠标右键并选择 “投影”。

使用 “视图” 菜单

使用 “视图” 菜单中的 “选项” 可以指定状态栏或工具栏是否显示并更改地图距离单位或坐标。

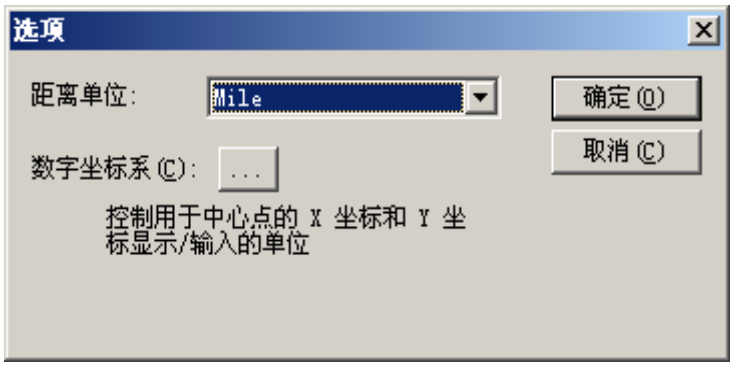
工具栏显示

若要显示或隐藏 “地图” 工具栏、“Geoset 名称” 工具栏或状态栏，应选取或清除适当的菜单选项。

选项

使用 “选项” 对话框设置地图所使用的距离单位和数字坐标。

要设置选项，选择 “视图” > “选项”。将显示 “选项” 对话框。



部分	说明
距离单位	选择用于地图的距离单位。
数字坐标	选择 Geoset Manager 将用来查找 X 和 Y 坐标的数字坐标。坐标系是一组参数，指示如何为对象解释位置坐标。几何对象中的每一点均由一对数字表示。这些数字是用于这些点的坐标。

工具和 GeoDictionary Manager

使用 “工具”菜单中的选项来访问 GeoDictionary Manager。通过 GeoDictionary Manager，您可以在 geodictionary 中注册各图层。在 geodictionary 中注册的图层可在其它 MapX 会话中用于数据绑定。

有关 Geodictionary Manager 的详细信息，请参阅**使用 GeoDictionary Manager**。

索 引

A

A\B\C\D\E 属性 （Affine Transform 对象）234
Abbreviations 属性 （Find 对象）311
AboutBox 方法 （MapObject）390
Abs( ) 函数 584
Acos( ) 函数 584
ActiveAnnotation 方法 （Annotations 集合）239
Add 方法 （DataSets 集合）263
Add 方法 （Features 集合）281
Add 方法 （Fields 集合）305
Add 方法 （Layers 集合）365
Add 方法 （Parts 集合）427
Add 方法 （Point 对象）430
Add 方法 （RowValues 集合）444
Add 方法 （Selection 集合）447
Add 方法 （Themes 集合）485
Add 方法 （Variables 集合）503
AddByID 方法 （Features 集合）281
AddDateField method (Fields collection) 306
AddFeature 方法 （Layer 对象）342
AddFeatureToolUsed 事件 536
AddField 方法 （Dataset 对象）258
AddFloatField （Fields 集合）306
AddGeosetLayers 方法 （Layers 集合）365
AddIntegerField （Fields 集合）306
AddLogicalField 方法 （Fields 集合）307
AddNumericField 方法 （Fields 集合）307
AddParameter 方法 （LayerInfo 对象）377
AddressOutOfRange 属性 （FindResult 对象）319
AddServerLayer 方法 （Layers 集合）366
AddSmallIntField 方法 （Fields 集合）308
AddStringField 方法 （Fields 集合）308
AddSymbol 方法 （Annotations 集合）239
AddText 方法 （Annotations 集合）239
AddUserDrawLayer 方法 （Layers 集合）367
AddXY 方法 （Point 对象）431
AffineTransform 对象 234
AffineTransform 属性 （CoordSys 对象）252
AggregationFunction 属性 （Field 对象）303
AggregationFunctionConstants 557
AllFeatures 方法 （Layer 对象）343
AllOthersCategory 属性
　（IndividualValueCategories）326
AllOthersCategory 属性 （RangeCategories 集合）434
AllOthersText 属性 （LegendTexts 集合）386
AllowEmptyRanges 属性 （ThemeProperties 对象）489
AnchorX 属性 （Label 对象）329
AnchorY 属性 （Label 对象）329
Angle 属性 （Label 对象）329
AnimationLayer 属性 （Layers 集合）368
Annotation 对象 238
AnnotationAdded 事件 537
AnnotationChanged 事件 537
AnnotationChangedTypeConstants 558
Annotations 集合 238
　属性和方法 42–43
Annotations 属性 （MapObject) 390
AnnotationTypeConstants 558
ApplyAttribute 属性 （ThemeProperties 对象）499
ApplyAttributeConstants 558
Area 属性 （Feature 对象）272
Area( ) 函数 585
AreaUnit 属性 （MapObject) 390
AreaUnitConstants 558
Asc( ) 函数 586
ASCII 文件
　定义 519
Asin( ) 函数 587
Atn( ) 函数 587
Attach 方法 （Feature 对象）272
AutoGenerate 属性 （LegendText 对象）387
AutoLabel 属性 （Layer 对象）343
AutoRecompute 属性 （Theme 对象）481
AutoRedraw 属性 （MapObject) 390
Azimuth 属性 （CoordSys 对象）253
安装
　步骤 53–54
　产品组件 52–53
　系统要求 52

B

BackColor 属性 （MapObject) 391
BarFramed 属性 （ThemeProperties 对象）495
BarFrameStyle (ThemeProperties 对象）495
BarGraduatedStack 属性 （ThemeProperties 对象）494
BarIndependentScale 属性 （ThemeProperties 对象）494
BarStacked 属性 （ThemeProperties 对象）494
BarWidth 属性 （ThemeProperties 对象）495
BeginAccess 方法 （Layer 对象）344
BeginDate 属性 （NotesQueryInfo 对象）421

BindLayer 对象 242
BindLayer 数据绑定
    BindLayer 参数 91
    BindLayer 对象 94
    将数据显示为点 92–95
BitmapSymbol 对象 247
BitmapSymbol 关键字 629
BitmapSymbols 集合 247
BMP 文件
    导出格式 218
BodyTextStyle 属性 （Legend 对象）379
Border 属性 （Title 对象）500
BorderStyle 属性 （ThemeProperties 对象）495
Bounds 属性 （BoundsConstraint 对象）249
Bounds 属性 （CoordSys 对象）253
Bounds 属性 （Feature 对象）273
Bounds 属性 （Layer 对象）344
Bounds 属性 （Layers 集合）368
Bounds 属性 （MapObject) 391
BoundsConstraint 对象 249
BufferFeatures 方法 （FeatureFactory 对象）288
BuildSourceRows 属性 （Datasets 集合）259
绑定数据 49
保存
    geosets 138, 666
比较运算符 578
编辑
    地图图元 109–110
边界区域
    定义 520
便携网络图形文件
    导出格式 218
表达式 575
标记图像文件格式文件
    导出格式 218
标签
    控制显示 79
    生成用于图层 78–80
    移除 79
    在 Geoset Manager 中 142–143
    在 Geoset Manager 中设置 670–671
标签工具 79
“标签属性”对话框 142
标签位置 79
标签主题 123
标准工具
    可用类型 208
饼状图地图
    主题地图类型 121
不绑定数据集
    数据源类型 90, 99

C

Caption 属性 （Feature 对象）273
Caption 属性 （Graphic 对象）324
Caption 属性 （Label 对象）329
Caption 属性 （Title 对象）500
CenterX 属性、 CenterY 属性 （Feature 对象）274
CenterX 属性 （MapObject) 391
CenterY 属性 （MapObject) 391
Centroid 属性 （Geoset 对象）322
Centroid( ) 函数 588
CentroidX( ) 函数 590
CentroidY( ) 函数 590
Chr$( ) 函数 591
CircleTypeConstants 560
ClearCustomLabels 方法 （Layer 对象）347
ClearSelection 方法 （Layers 集合）369
ClearSelection 方法 （Selection 集合）447
ClipLine 方法 （MapObject）392
CliplineV 方法 （MapObject）393
ClippedBounds 属性 （Layer 对象）346
ClippedBounds 属性 （Layer 集合）368
Clone 方法 （CoordSys 对象）253
Clone 方法 （Feature 对象）273
Clone 方法 （Features 集合）282
Clone 方法 （RowValues 对象）445
Clone 方法 （Selection 集合）448
Clone 方法 （Style 对象）464
Clone 方法 （Variables 集合）504
CloseMatchMax 属性 （Find 对象）311
ClosestAddr 属性 （Find 对象）312
ColorConstants 560
ColorMethod 属性 （ThemeProperties 对象）499
COM 对象
    请参阅静态数据集对象
CombineFeatures 方法 （FeatureFactory 对象）289
Common 方法 （Features 集合）282
Common 方法 （Selection 集合）448
Compact 属性 （Legend 对象）380
CompactTitle 属性 （Legend 对象）380
CompactTitleStyle 属性 （Legend 对象）380
ComputeTheme 属性 （Theme 对象）482
ConnectString 属性 （OCIQueryInfo 对象）425
ConnectString 属性 （ODBC QueryInfo 对象）426
Contains 方法 （Dataset 对象）265
ConversionConstants 562
ConvertCoord 方法 （MapObject）393
ConvertCoordP 方法 （Map 对象）394

ConvertCoordV 方法 （MapObject）394
CoordSys 对象 252
查询属性 178
获取 178
CoordSys 属性 （BindLayer 对象）242
CoordSys 属性 （Layer 对象）346
CoordSysTypeConstants 563
Cos( ) 函数 592
Count 属性 （Annotations 集合）240
Count 属性 （BitmapSymbol 对象）247
Count 属性 （DataSets 集合）265
Count 属性 （Features 集合）282
Count 属性 （Fields 集合）309
Count 属性 （Geoset 集合）322
Count 属性 （Labels 集合）333
Count 属性 （Layers 集合）369
Count 属性 （LegendTexts 对象）387
Count 属性 （MultivarCategories 集合）420
Count 属性 （Parts 集合）427
Count 属性 （Point 对象）431
Count 属性 （RangeCategories 集合）434
Count 属性 （ResolveObjects 集合）440
Count 属性 （RowValues 集合）443
Count 属性 （Selection 集合）446
Count 属性 （SourceRow 对象）454
Count 属性 （Theme 对象）486
Count 属性 （Variables 集合）503
CreateAllFeaturesConstraint 方法 344
CreateArc 方法 （FeatureFactory 对象）290
CreateBoundsConstraint method (Layer Object) 345
CreateCircularRegion 方法 （FeatureFactory 对象）293
CreateCollectionFeature 方法
（FeatureFactory 对象）294
CreateCustomTool 方法 （MapObject）395
CreateEllipticalRegion 方法 （FeatureFactory 对象）294
CreateFeaturesConstraint 方法 （Layer 对象）345
CreateLayer 方法 （Layers 集合）369
CreateLine 方法 （FeatureFactory 对象）295
CreateMultipoint 方法 （FeatureFactory 对象）295
CreateRegion 方法 （FeatureFactory 对象）296
CreateSymbol 方法 （FeatureFactory 对象）296
CreateText 方法 （FeatureFactory 对象）296
CurrencyFormat 属性 （Legend 对象）380
CurrentTool 属性 （MapObject) 396
CursorConstants 564
菜单项
用 Visual C++ 创建 194–195
产品功能 34–35
常数 557
范围分布 117–118
工具类型 213
合计函数 92
数据集 89
主题类型 113
超文本
链接 520
传输协议 (HTTP) 520
重新通过 MrSID 分发您的 MapX 应用程序 230
创建
Features 集合 106–107
geosets 137–138, 666
MapInfo 地图目录 169–170
Visual C++ 对话框中的地图 203–205
Visual C++ 中的 MapX 控件 192
Visual C++ 中的菜单项 194–195
标签的标注 79
地图图元 109
定制工具 212–214
多边形绘制工具 214–215
汇总工具 154
特定点的 Selection 集合 109
永久点图层 95
逐层细化工具 152
主题 112–114
错误代码 547

D

Database 属性 （NotesQueryInfo 对象）421
Database 属性 （NotesViewInfo 对象）424
DataField 属性 （LabelProperties 对象）335
DataMax 属性 （Theme 对象）482
DataMin 属性 （Theme 对象）482
DataMismatch 事件 538
Dataset 42
插入 Visual Basic 示例 60
Dataset 对象 257
DataSet 属性 （LabelProperties 对象）336
DataSet 属性 （MapObject) 396
Dataset 属性 （RowValue 对象）442
DataSetGeoField 属性 （MapObject) 397
DataSets 集合 257
处理 DataSet 对象 87
DataSets 属性 （Layer 集合）347
DataSets 属性 （MapObject) 397
DataSetTheme 属性 （MapObject) 397
DatasetTypeConstants 565
DataSource 属性 （ODBC QueryInfo 对象）426
DataValue 属性 （ThemeProperties 对象）489

Datum 对象 267
Datum 属性 （CoordSys 对象）253
DBMS 服务器
    访问远程空间数据 158
Decimals 属性 （Field 对象）303
DefaultConversionResolution 属性 （MapObject) 398
DefaultNumericValue 属性 （NotesQueryInfo 对象）421
DefaultStringValue 属性 （NotesQueryInfo 对象）422
DefaultStyle 属性 （Map 对象）399
DeformatNumber$( ) 函数 593
DeleteFeature 方法 （Layer 对象）347
Delphi
    安装 MapX 61
Delphi Native
    数据源类型 89
Disable 方法 （FeaturesConstraint 对象）285
Disable 属性 （BoundsConstraint 对象）250
DisplayCoordSys 属性 （Map 对象）399
Distance 方法 （Map 对象）399
Distance( ) 函数 593
DistMethod 属性 （ThemeProperties 对象）489
DistribMethod 常数 566
DotColor 属性 （ThemeProperties 对象）497
DotSize 属性 （ThemeProperties 对象）490
DotSizeConstants 566
DrawLabelsAfter 属性 （Layer 对象）348
DrawLineSample 方法 （Style 对象）465
Drawn 属性 （Label 对象）329
DrawRegionSample 方法 （Style 对象）465
DrawSymbolSample 方法 （Style 对象）465
DrawTextSample 方法 （Style 对象）466
DrawUserLayer 事件 539
DrilldownAddFeatures 方法 （Layer 对象）348
DrilldownRemoveFeatures 方法 （Layer 对象）349
DrilldownReset 方法 （Layer 对象）349
Duplicate 属性 （LabelProperties 对象）336
Dynamic 参数
    动态数据绑定 91
DynamicSelection 属性 （Map 对象）400
打开
    geosets 136–137, 665–666
打印地图 219
导出
    地图 218
    选择模式 219
地产产权
    定义 521
地理信息系统 (Geographic Information System, GIS)
    定义 521
迪卡尔坐标 521
地图
    打印 219
    导出 218
地图菜单
    使用 Geoset Manager 138–143, 667–672
地图创建
    geoset 概览 41
    Map 对象 38
    地图属性 38–39
    可创建对象 43–44
    图层概览 40
地图控件
    添加 55
地图图像
    定义 523
地图图元
    编辑 109–110
    创建 109
    独立 108
    获取要编辑的 110
    删除 110
    说明 106
    修改 110
点参考绑定 95
点密度地图
    主题地图类型 119
点样式
    远程空间表 174
调度 ID 表 633
定制工具
    Visual C++ 示例 197–198
    编写工具处理程序 213–214
    创建 212–214
    多边形绘制工具 214–215
    类型常数 213
定制数据集支持 509, 509–515
动画图层 81
独立图元 108
对象
    可创建 43–44
对象编辑工具 209
多边形绘制工具 214–215

E

Eccentricity 属性 （Datum 对象）267
Editable 属性 （Annotations 集合）240
Editable 属性 （Title 对象）500
EditableLabels 属性 （Map 对象）400
Edited 属性 （Label 对象）329
EditedAnchor 属性 （Label 对象）330

EditedAngle 属性 （Label 对象）330
EditedCaption 属性 （Label 对象）330
EditedLineStyle 属性 （Label 对象）330
EditedLineType 属性 （Label 对象）330
EditedMultiSegment 属性 （Label 对象）331
EditedOffset 属性 （Label 对象）331
EditedPosition 属性 （Label 对象）331
EditedTextStyle 属性 （Label 对象）331
EditedVisibile 属性 （Label 对象）332
Ellipsoid 属性 （Datum 对象）267
Enabled 方法 （AllFeaturesConstraint 对象）237
Enabled 方法 （BoundsConstraint 对象）250
Enabled 方法 （FeaturesConstraint 对象）286
Enabled 属性 （AllFeaturesConstraint 对象）236, 237
Enabled 属性 （BoundsConstraint 对象）251
Enabled 属性 （FeaturesConstraint 对象）286
EndAccess 方法 （Layer 对象）350
EndDate 属性 （NotesQueryInfo 对象）422
EraseFeature 方法 （FeatureFactory 对象）297
ExactMatch 属性 （FindResult 对象）319
Exp( ) 函数 594
ExportFormatConstants 566
ExportLegend 方法 （Legend 对象）380
ExportLineSample 方法 （Style 对象）466
ExportMap 方法 （Map 对象）401
ExportRegionSample 方法 （Style 对象）467
ExportSelection 属性 （Map 对象）402
ExportSymbolSample 方法 （Style 对象）467
ExportTextSample 方法 （Style 对象）468
二元主题地图 122

F

FalseEasting、 FalseNorthing 属性 （CoordSys 对象）
253
Feature 对象 271
属性 108
FeatureEditMode 属性 （Map 对象）402
FeatureEditModeConstants 566
FeatureFactory
方法 109
FeatureFactory 对象 288
FeatureFactory 属性 （Map 对象）402
FeatureID 属性 （Feature 对象）274
FeatureID 属性 （FindMatch 对象）317
FeatureIDFromFeatureName 方法 （Layer 对象）350
FeatureKey 属性 （Feature 对象）274
FeatureKey 属性 （FindMatch 对象）317
FeatureKeyFromFeatureName 方法 （Layer 对象）351
FeatureKey （Label 对象）332
Features 集合 271
创建 106–107
说明 106
修改 107
FeaturesConstraint 对象 285
Features （FeaturesConstraint 对象）286
FeatureTypeConstants 566
Field 对象 302
Field 属性 （RowValue 对象）443
Fields 参数
确定数据源字段 91
Fields 集合 302
数据绑定 91
Fields 属性 （Dataset 对象）259
Fields 属性 （Theme 对象）483
FieldTypeConstants 567
Filespec 属性 （BindLayer 对象）242
FileSpec 属性 （Layer 对象）351
FillPatternConstants 567
Find 对象
结果码 126–127
确定地图图元的位置 126
Find 对象 , 使用
确定地图图元的位置 126
Find 对象。311
Find 属性 （Layer 对象）351
FindDataset 属性 （Find 对象）312
FindFeature 对象 315
存储图元对象属性 126–127
FindField 属性 （Find 对象）312
FindMatch 对象 317
FindMatches 集合 317
FindRC 属性 （FindFeature 对象）315
FindRC 属性 （FindResult 对象）320
FindResult 对象 319
Fix( ) 函数 595
Format$( ) 函数 596
FormatDate$( ) 函数 598
FormatNumber$( ) 函数 599
FullTextSearch 属性 （NotesQueryInfo 对象）422
范围地图
范围分布类型 117–118
主题地图类型 116–118

方法
DataSets 集合 87
FeatureFactory 109
Fields 集合 91
Layer 集合 70–74
Selection 集合 108
Themes 集合 112–114
在 Visual C++ 中访问 191
仿射转换 185
访问远程空间表
MapInfo 地图目录 168–173
变成可以进行地图绘制 172–173
指定样式 174–175
访问远程空间数据
DBMS 服务器 158
LayerInfo 参数 162
ODBC 连接字符串 162–164
缓存 166–168
属性数据 165–166
通过 X/Y 列进行地图绘制 159
性能问题 166
非地球地图
定义 521
分级符号地图
主题地图类型 118
符号样式
远程空间表 174
服务器端图像地图
定义 521
服务器 （NotesViewInfo 对象）424

G

Geocoding
定义 519
Geodictionary
geodictionary 文件 130–131, 527–528
数据绑定 96–97
注册图层 134, 531
Geodictionary Manager
命令行选项 135–136, 532–533
实用程序位置 653
运行 131–134, 528–531
注册 MapInfo 表 527
GeoDictionary 属性 （Map 对象）402
Geofield 参数
数据源 90
GeoField 属性 （Dataset 对象）259
GeoSet
概览 41
Geoset 对象 322
Geoset 关键字 621
Geoset Manager
标签 142–143, 670–671
地图菜单选项 138–143, 667–672
平移工具 139, 667
设置缩放级别 138–140, 667–668
设置投影 143, 671–672
视图菜单选项 144, 672
实用程序位置 653
图层控制选项 140–143, 669–671
GeoSet 属性 （Map 对象）403
Geosets
保存 138, 666
创建 137–138, 666
打开 136–137, 665–666
Geosets 集合 322
GeoSets 属性 （Map 对象）404
GeoSetWidth 属性 （Map 对象）404
GetDrilldownFeaturesByID 方法 （Layer 对象）351
GetFeatureByID 方法 （Layer 对象）352
GetFeatureByKey 方法 （Layer 对象）352
GetXY 方法 （Points 集合）431
GIF 文件
导出格式 218
Graduated 属性 （ThemeProperties 对象）490
GraduateSizeBy 属性 （ThemeProperties 对象）497
Graduation Constants 567
Graphic 对象 324
Graphic 属性 （Annotation 对象）238
个别值地图
主题地图类型 114–116
公共网关接口 (Common Gateway Interface, CGI)
定义 521
工具
Visual Basic 6.0 示例 59
标准类型 208
对象编辑 209
多边形绘制工具 214–215
指定当前工具 208
工具处理程序 213–214
光栅图像
显示 80

H

HasMultipoint 属性 （Feature 对象）275
HasPolyline 属性 （Feature 对象）276
HasRegion 属性 （Feature 对象）276
Height 属性 （Legend 对象）381
Height 属性 （Rectangle 对象）437

HTML
定义 520
客户端图像地图 (CSIM) 522
hWnd 属性 （Map 对象）404
函数 582
画笔样式
远程空间表 174
缓存
访问远程空间数据 166–168
绘图法
定义 521
绘制图层 82
汇总工具
创建 154

I

Independent 属性 （ThemeProperties 对象）490
Indexed 属性 （Field 对象）303
IndividualValueCategories 集合 326
IndividualValueCategories 属性
（ThemeProperties 对象）491
IndividualValueCategory 对象 326
InflectionColor 属性 （ThemeProperties 对象）498
InflectionRange 属性 （ThemeProperties 对象）498
InflectRanges 属性 （ThemeProperties 对象）498
InfotipPopupDelay 属性 （Map 对象）405
InfotipSupport 属性 （Map 对象）404
InsertionLayer (Layers 对象）370
InStr( ) 函数 600
Int( ) 函数 601
Internet 服务器应用程序编程接口 (Internet Server
Application Programming Interface, ISAPI)
定义 519
IntersectFeatures 方法 （FeatureFactory 对象）299
IntersectionNotFound 属性 （FindResult 对象）320
IntersectionPointConstants 568
IntersectionPoints 方法 （FeatureFactory 对象）300
IntersectionTest 方法 （FeatureFactory 对象）301
IntersectionTestConstants 568
Invalidate 方法 （Layer 对象）353
IsPointVisible 方法 （Map 对象）405
Item 方法 （Variables 集合）504
Item 属性 （Annotations 集合）240
Item 属性 （BitmapSymbol 对象）248
Item 属性 （DataSets 集合）265
Item 属性 （Features 集合）283, 447
Item 属性 （Fields 集合）309
Item 属性 （Geoset 集合）323
Item 属性 （IndividualValueCategories 集合）326
Item 属性 （Labels 集合）334
Item 属性 （Layers 集合）370
Item 属性 （LegendTexts 对象）387
Item 属性 （MultivarCategories 集合）420
Item 属性 （Parts 集合）427
Item 属性 （Points 集合）431
Item 属性 （RangeCategories 集合）434
Item 属性 （ResolveObjects 集合）441
Item 属性 （RowValues 集合）443
Item 属性 （SourceRow 对象）454
Item 属性 （Theme 对象）486

J

JPEG 文件
导出格式 218
集成开发环境
(Integrated Development Environment, IDE)
定义 521
交互式标记 79
节点选择和编辑 209
结果码
Find 对象 126–127
经度
定义 521
经线
定义 522
经线 / 纬线
定义 522
精度 （Field 对象）303
静态数据集对象
接口 509–515
注册 509
静态页
定义 522

K

KeyField 属性 （Layer 对象）353
KeyLength 属性 （BindLayer 对象）243
KeyValue 属性 （Feature 对象）275
可编辑 （Layer 对象）350
可创建
对象 43–44
客户端脚本
定义 522
客户端图像地图 (CSIM)
定义 522
快捷菜单
在 Visual C++ 中添加 200
宽度 （Field 对象）303
框架 522

L

Label 对象和 Labels 集合 328
Label 集合 328
LabelAlong (LabelProperties 对象）336
LabelAlongConstants 568
LabelAtPoint 方法 （Layer 对象）354
LabelChanged Event 539
LabelMax 属性 （LabelProperties 对象）336
LabelPartialObjects 属性 （LabelProperties 对象）337
LabelPoint 属性 （Feature 对象）276
LabelProperties 对象 335
    控制标签显示 79–80
LabelProperties 属性 （Layer 对象）354
Labels 属性 （Layer 对象）354
LabelZoom 属性 （LabelProperties 对象）337
LabelZoomMax 属性 （LabelProperties 对象）337
LabelZoomMin 属性 （LabelProperties 对象）337
Layer 对象 340
    图层属性 74
Layer 集合 340
    方法 70–74
    获取 66–68
    检查图层类型 75
    检查图层特征对象类型 76
    属性 68–70
Layer 属性 （Dataset 对象）259
Layer 属性 （Feature 对象）276
Layer 属性 （Label 对象）332
Layer 属性 （Theme 对象）483
LayerBeginAccessConstants 568
LayerEndAccessConstants 568
LayerInfo 对象 373
    参数 162
LayerInfoTypeConstants 568
LayerName 属性 （BindLayer 对象）243
LayerPackConstants 569
Layers 属性 （Map 对象）405
LayersDlg 方法 （Layers 集合）370
LayerSrvLayerOptions 569
LayerType 属性 （BindLayer 对象）243
LayerTypeConstants 569
LCase$( ) 函数 601
Left 属性 （Legend 对象）381
Left$( ) 函数 602
Legend 对象 379
    对于 Theme 对象 123
Legend 属性 （Theme 对象）483
LegendDlg 方法 （Legend 对象）381
LegendText 对象 386
LegendTexts 386
LegendTexts 集合 386
LegendTexts 属性 （Legend 对象）382
Length 属性 （Feature 对象）276
LineColor 属性 （Style 对象）469
LineInterleaved 属性 （Style 对象）469
LineStyle 属性 （Style 对象）469
LineStyleCount 属性 （Style 对象）470
LineSupportsInterleave 属性 （Style 对象）471
LineType 属性 （Label 对象）332
LineType 属性 （LabelProperties 对象）338
LineTypeConstants 569
LineWidth 属性 （Style 对象）471
LineWidthUnit 属性 （Style 对象）471
Log( ) 函数 603
Lotus Notes
    数据源类型 90, 98
LTrim$( ) 函数 603
链接
    超文本 520
连接字符串示例 163
逻辑比较 579
逻辑运算符 579

M

Map 对象 388
    属性 38
    图元类型 49
MapDraw 事件 540
MapDrawConstants 569
MapInfo 表
    导入 SpatialWare 169
    数据源类型 89, 98
    在 Geodictionary 中注册 527
MapInfo 地图目录
    创建 169–170
    使远程表成为可以进行地图绘制的表 168–173
MapInfo Professional
    创建地图目录 168
MAPINFOW.PRJ 文件
    参数 183
    定制 184
    使用设置 183–184
MapInitialized 事件 540
MapPaperHeight 属性 （Map 对象）406
MapPaperWidth 属性 （Map 对象）406
MapScreenHeight 属性 （Map 对象）406
MapScreenWidth 属性 （Map 对象）406
MapViewChanged 事件 540
MapUnit 属性 （Map 对象）405
MapUnitConstants 570

MapX
安装步骤 53–54
产品组件 52–53
文档集说明 61–62
系统要求 52
主要功能 34–35
MapX 控件
用 Visual C++ 创建 192
MapX 事件 535
MapX 数据
安装声明 653
来源 655–664
MapX 异常错误
在 Visual C++ 中 202–203
MapX 应用程序
从早期的版本升级 190
MapX 中的可创建对象 43
MapXtreme
术语表 136
术语词汇表 519–525
MatchedFeature 属性 （FindResult 对象）320
Matches 属性 （FindResult 对象）320
MatchThreshhold 属性 （Map 对象）406
Max 属性 （RangeCategories 集合）435
Maximum( ) 函数 604
MaxNumDocs 属性 （NotesQueryInfo 对象）422
MaxSearchTime 属性 （Map 对象）407
MaxVectorSymbolCharacter 属性 （Style 对象）478
MBR( ) 函数 604
Mid$( ) 函数 605
MilitaryGridReferenceToPoint 方法 （Map 对象）409
MilitaryGridReferenceToPointV 方法 （Map 对象）408
MilitraryGridReferenceFromPoint 方法
（Map 对象）407
Min 属性 （RangeCategories 集合）435
Minimum( ) 函数 605
MinVectorSymbolCharacter 属性 （Style 对象）478
Month( ) 函数 607
Move 方法 （Layers 集合）371
MouseIcon 属性 （Map 对象）410
MousePointer 属性 （Map 对象）411
MouseWheel 事件 541
MouseWheelSupportConstants 570
MrSID 230
MultipleMatches 属性 （FindResult 对象）320
Multipoint 属性 （Feature 对象）277
MultiSegment 属性 （Label 对象）332
MultivarCategories 集合 420
MultivarCategories 属性 （ThemeProperties 对象）491
MultivarCategory 对象 420

N

Name 参数
数据集 90
Name property (DataSet object) 259
Name property (Feature object) 277
Name 属性 （BitmapSymbol 对象）247
Name 属性 （Field 对象）303
Name 属性 （FindMatch 对象）317
Name 属性 （Layer 对象）355
Name 属性 （Theme 对象）483
NegativeSymbolStyle 属性
（ThemeProperties 对象）496
Netscape 服务器应用程序编程接口 (Netscape Server
Application Programming Interface, NSAPI)
定义 519
Nodes 属性 （Feature 对象）277
NoFeatures 方法 （Layer 对象）355
NotesQueryInfo 对象 421
NotesViewInfo 对象 424
NumericCoordSys 属性 （Map 对象）411
NumItems 属性 （IndividualValueCategories 集合）327
NumRanges 属性 （ThemeProperties 对象）491
内置帮助程序对话框
Visual C++ 201–202

O

ObjectInfo( ) 函数 607
ObjectLen( ) 函数 607
OCI 数据源类型 90
OCIQueryInfo 对象 425
ODBC 连接字符串 162–164
ODBC 连接字符串格式 162
ODBC 驱动程序
定义 519
ODBC 数据源类型 90, 98
ODBCQueryInfo 对象 426
Offset 方法 （Feature 对象）278
Offset 方法 （Point 对象）429
Offset 方法 （Rectangle 对象）437
Offset 属性 （Label 对象）333
Offset 属性 （LabelProperties 对象）338
OLE_COLOR 值 631
Oracle Spatial 参考支持 161
Oracle Spatial 连接字符串格式 163
OriginLatitude、 OriginLongitude 属性
（CoordSys 对象）254
OtherBoundary 属性 （Find 对象）313
Overlap 属性 （LabelProperties 对象）338
OverrideStyle 属性 （Layer 对象）355
OverwriteFile 属性 （BindLayer 对象）244

P

Pack 方法 （Layer 对象）355
Pan 方法 （Map 对象）411
PanAnimationLayer 属性 （Map 对象）412
PaperHeight 属性 （Legend 对象）382
PaperWidth 属性 （Legend 对象）382
PaperUnit 属性 （Map 对象）412
PaperUnitConstants 570
Parallel 属性 （LabelProperties 对象）338
Parameter 属性 （LayerInfo 对象）378
PartialSegments 属性 （LabelProperties 对象）339
Parts 集合 427
Parts 属性 （Feature 对象）278
PathName 属性 （Geoset 对象）322
PenStyleConstants 571
Perimeter 属性 （Feature 对象）278
Perimeter( ) 函数 608
Photoshop 文件
    导出格式 218
PickCoordSys 方法 （CoordSys 对象）254
PickLine 方法 （Style 对象）471
PickRegion 方法 （Style 对象）472
PickSymbol 方法 （Style 对象）472
PickText 方法 （Style 对象）472
PieClockwise 属性 493
PieGraduated 属性 （ThemeProperties 对象）494
PieHalfPies 属性 （ThemeProperties 对象）493
PieStartAngle 属性 493
PNG 文件
    导出格式 218
Point 对象 429
Point 属性 （Feature 对象）279
Points 集合 429
Polyline 属性 （Feature 对象）279
PolyToolFlagConstants 571
PolyToolUsed 事件 542
Position 属性 （Graphic 对象）324
Position 属性 （Label 对象）333
Position 属性 （LabelProperties 对象）339
Position 属性 （Title 对象）500
PositionConstants 571
PositiveSymbolStyle 属性 （ThemeProperties 对象）496
PowerBuilder
    数据源类型 90, 98
PredominantFeatureType 属性 （Layer 对象）356
PreferCompactLegends 属性 （Map 对象）412
PrimeMeridian 属性 （Datum 对象）268
PrintLegend 方法 （Legend 对象）383
PrintMap 方法 （Map 对象）412
Proper$( ) 函数 609
Properties 属性 （Theme 对象）484
PropertyPage 方法 （Map 对象）413
PSD 文件
    导出格式 218
平移工具
    Geoset Manager 139, 667

Q

Query 属性 （NotesQueryInfo 对象）422
Query 属性 （OCIQueryInfo 对象）425
确定地图图元的位置 126

R

Range 属性 （CoordSys 对象）254
RangeCategories 集合 434
RangeCategories 属性 （ThemeProperties 对象）491
RangeCategory 对象 434
ReadFromFile 方法 （State 对象）458
ReadOnly property (Dataset object) 261
ReadOnly 属性 （RowValue 对象）442
ReadOnly 属性 （RowValues 集合）443
Rectangle 对象 437
RedrawInterval 属性 （Map 对象）413
RefColumn1 属性 （BindLayer 对象）244
RefColumn2 属性 （BindLayer 对象）245
ReferenceLayer 属性 （BindLayer 对象）245
ReferenceLayerfield 属性 （BindLayer 对象）245
RefineDataset 属性 （Find 对象）313
RefineField 属性 （Find 对象）313
RefineLayer 属性 （Find 对象）313
RefineRegion 属性 （FindResult 对象）321
Refresh 方法 （BitmapSymbol 集合）248
Refresh 方法 （DataSet 对象）260
Refresh 方法 （Layer 对象）357
Refresh 方法 （Map 对象）414
Region 属性 （Feature 对象）279
RegionBackColor 属性 （Style 对象）472
RegionBorderColor 属性 （Style 对象）473
RegionBorderStyle 属性 （Style 对象）473
RegionBorderWidth 属性 （Style 对象）473
RegionBorderWidthUnit 属性 （Style 对象）473
RegionColor 属性 （Style 对象）473
RegionPattern 属性 （Style 对象）473
RegionTransparent 属性 （Style 对象）474
Remove 方法 （Annotations 集合）241
Remove 方法 （DataSets 集合）266
Remove 方法 （Features 集合）283
Remove 方法 （Fields 集合）309
Remove 方法 （Layers 集合）372
Remove 方法 （Parts 集合）428

Remove 方法 （Point 对象）433
Remove 方法 （RowValues 对象）444
Remove 方法 （Selection 集合）448
Remove 方法 （Theme 对象）487
Remove 方法 （Variables 集合）504
RemoveAll 方法 （Annotations 集合）241
RemoveAll 方法 （DataSets 集合）266
RemoveAll 方法 （Fields 集合）310
RemoveAll 方法 （Layers 集合）372
RemoveAll 方法 （Parts 集合）428
RemoveAll 方法 （Point 对象）433
RemoveAll 方法 （RowValues 集合）444
RemoveAll 方法 （Theme 对象）487
RemoveAll 方法 （Variables 集合）505
RemoveByID 方法 （Features 集合）283
Replace 方法 （Features 集合）283
Replace 方法 （Selection 集合）449
RequestData 事件 542
Reset 方法 （State 对象）458
ResolveDataBind 事件 543
ResolveDataBindConstants 572
ResolveDataBindEx 事件 543
ResolveObject 对象 440
ResolveObjects 集合 440
Restore 方法 （DataSets 集合）261
Restore 方法 （State 对象）459
Right$( ) 函数 610
RotateX、 RotateY、 RotateZ 属性 （Datum 对象）268
Rotation 属性 （Map 对象）415
Row 属性 （SourceRow 对象）454
RowCount 属性 （Dataset 对象）260
Round( ) 函数 610
RoundBy 属性 （ThemeProperties 对象）497
RoundRanges 属性 （ThemeProperties 对象）497
RowValue 对象 442
RowValues 集合 442
RowValues property (Dataset object) 260
RTrim$( ) 函数 611
日期比较 579
日期子句 580

S

Save 方法 （Map 对象）460
SaveMapAsGeoset 方法 （Map 对象）415
ScaleAdjust 属性 （Datum 对象）268
ScaleFactor 属性 （CoordSys 对象）254
Score 属性 （FindMatch 对象）317
SDO_GEOMETRY 161
Search 方法 （Find 对象）314
Search 方法 （Layer 对象）357
SearchAtPoint 方法 （Layer 对象）358
SearchEx 方法 （Find 对象）312
SearchPath 属性 （Map 对象）416
SearchResultTypeConstants 572
SearchTypeConstants 572
SearchWithinDistance 方法 （Layer 对象）359
SearchWithinFeature 方法 （Layer 对象）359
SearchWithinRectangle 方法 （Layer 对象）360
Secondary Geofield 参数 90
SecondaryGeoField 属性 （Dataset 对象）262
Selectable 属性 （Layer 对象）361
SelectAll 方法 （Selection 集合）449
SelectByID 方法 （Selection 集合）449
SelectByPoint 方法 （Selection 集合）450
SelectByRadius 方法 （Selection 集合）451
SelectByRectangle 方法 （Selection 集合）451
SelectByRegion 方法 （Selection 集合）452
Selected 属性 （Label 对象）333
Selection 集合 446
    SelectionChanged 事件 109
    方法 108
    说明 106
    搜索类型常数 109
    在特定点创建 109
Selection 属性 （Layer 对象）361
SelectionChanged 事件 544
SelectionStyle 属性 （Map 对象）416
SelectionType 常数 572
SemiMajorAxis、 SemiMinorAxis 属性 （Datum 对象）
    268
Server 属性 （NotesQueryInfo 对象）423
Set 方法 （AffineTransform 对象）235
Set 方法 （CoordSys 对象）255
Set 方法 （Datum 对象）269
Set 方法 （Point 对象）430
Set 方法 （Rectangle 对象）438
SetSize 方法 （Map 对象）416
Sgn( ) 函数 612
Shapefile (.shp) 373
ShowCentroids （Layer 对象）361
ShowCount 属性 （Legend 对象）383
ShowEmptyRanges 属性 （Legend 对象）383
ShowLineDirection （Layer 对象）361
ShowNegativeValues 属性
    （ThemeProperties 对象）496
ShowNodes （Layer 对象）361
Sin( ) 函数 612
Size 属性 （ThemeProperties 对象）492
Smooth 属性 （Feature 对象）279
SnapTolerance (Map 对象）417

SnapToNodeSupport (Map 对象）417
SourceData 参数
    数据绑定 89
SourceMatch 属性 （ResolveObject 对象）440
SourceRow 对象 454
SourceRows 集合 454
SourceRows 属性 （Dataset 对象）262
Space$( ) 函数 613
SpatialWare
    对 MapX 应用程序进行故障排除 175
    将 MapInfo 表导入 169
SpreadBy 属性 （ThemeProperties 对象）492
SpreadByConstants 572
SqlQuery 属性 （ODBC QueryInfo 对象）426
Sqr( ) 函数 613
SRID 161
StandardParallelOne、 StandardParallelTwo 属性
    （CoordSys 对象）256
State 对象 456
Str$( ) 函数 616
Stream 属性 （State 对象）461
StringCompare( ) 函数 614
StringCompareIntl( ) 函数 614
StringToDate( ) 函数 615
Style 对象 463
Style 属性 （Feature 对象）280
Style 属性 （Graphic 对象）324
Style 属性 （IndividualValueCategories 集合）327
Style 属性 （Label 对象）333
Style 属性 （LabelProperties 对象）339
Style 属性 （Layer 对象）362
Style 属性 （MultivarCategory 对象）420
Style 属性 （RangeCategories 集合）436
StyleUnitConstants 573
Substitute 属性 （FindResult 对象）321
SubTitle 属性 （Legend 对象）383
SubTitleStyle 属性 （Legend 对象）384
SupportsBitmapSymbols 属性 （Style 对象）474
SupportsPack 方法 （Layer 对象）362
SymbolBitmapColor 属性 （Style 对象）474
SymbolBitmapName 属性 （Style 对象）475
SymbolBitmapOverrideColor 属性 （Style 对象）475
SymbolBitmapSize 属性 （Style 对象）475
SymbolBitmapTransparent 属性 （Style 对象）476
SymbolCharacter 属性 （Style 对象）476
SymbolFont 属性 （Style 对象）476
SymbolFontBackColor 属性 （Style 对象）477
SymbolFontColor 属性 （Style 对象）477
SymbolFontHalo 属性 （Style 对象）477
SymbolFontOpaque 属性 （Style 对象）477
SymbolFontRotation 属性 （Style 对象）477
SymbolFontShadow 属性 （Style 对象）478
SymbolStyle 属性 （ThemeProperties 对象）492
SymbolType 属性 （Style 对象）480
SymbolTypeConstants 573
SymbolVectorColor 属性 （Style 对象）478
SymbolVectorSize 属性 （Style 对象）478
删除
    地图图元 110
事件 535
事件处理程序
    定义 522
示例 Geoset 623
示例应用程序
    Visual C++ 190
    其它程序 35
视图菜单
    使用 Geoset Manager 144, 672
视图 （NotesViewInfo 对象）424
使用 Dataset 对象 87
使用 Fields.Add 方法
    将列数据作为字段添加 92
术语表 136
术语词汇表 519–525
数据绑定
    BindLayer 92–95
    DataSets.Add 方法 88
    Dynamic 参数 91
    Fields 参数 91
    MapX 事件用于 96
    SourceData 参数 89
    不绑定数据集 99
    将数据加入地图 86–87
    使点图层成为永久的 95
    使用 Dataset 对象 87
    使用 Geodictionary 96–97
    数据绑定 42
    数据源类型 86, 97–98
    在 Visual C++ 中 198–200
    注册 MapInfo 表 527
数据访问对象记录集
    数据源类型 89, 97
数据合计
    Fields.Add 方法 92
    使用 Fields.Add 方法 92

数据集 42
Fields 集合 91
Name 参数 90
Themes 集合 112–114
Type 常数 89
数据绑定 49, 87
刷新 97
数据集支持 509
数据源
Fields.Add 方法 92
Geofield 参数 90
类型 89, 97–98
数据绑定 49
属性
Annotations 集合 43
BindLayer 对象 94
Feature 对象 108
Layer 对象 74
Layer 集合 68–70
Map 对象 38
Theme 对象 114
在 Visual C++ 中访问 191
属性数据
绑定 91
访问 165–166
属性页
修改地图 38–39
数学运算符 576
数字比较 578
刷新数据集 97
搜索类型常数
选择 109
缩放
图层 522
缩放级别
在 Geoset Manager 中设置 138–140, 667–668
缩放图层
标签 79
图层 76–78
图层设置 75

T

TableMatch 属性 （ResolveObject 对象）440
TableName 属性 （ResolveObject 对象）440
Tan( ) 函数 617
Text 属性 （LegendText 对象）386
TextFont 属性 （Style 对象）478
TextFontAllCaps 属性 （Style 对象）479
TextFontBackColor 属性 （Style 对象）479
TextFontColor 属性 （Style 对象）479
TextFontDblSpace 属性 （Style 对象）479
TextFontHalo 属性 （Style 对象）479
TextFontOpaque 属性 （Style 对象）479
TextFontRotation 属性 （Style 对象）480
TextFontShadow 属性 （Style 对象）480
TextStyle 属性 （Title 对象）501
Theme 对象 481
属性 114
ThemeDlg 方法 122
ThemeDlg 方法 （Theme 对象）484
ThemeModifyRequested 事件 544
ThemeProperties 对象 488
ThemeProperties 属性 （Theme 对象）484
Themes 集合 481
方法 112–114
ThemeProperties 对象 123
Themes 属性 （Dataset 对象）262
ThemeTypeConstants 573
TIF 文件
导出格式 218
Title 对象 500
Title 属性 （Legend 对象）384
Title 属性 （Map 对象）418
TitleText 属性 （Map 对象）418
ToolConstants 573
ToolFlagConstants 574
ToolTypeConstants 574
ToolUsed 事件 545
ToolUsedEvent 214
Top 属性 （Legend 对象）384
TrueType 字体 226
TrueType 字体符号的关键字 629
Type 属性 （Annotation 对象）239
Type 属性 （CoordSys 对象）256
Type 属性 （DataSet 对象）262
Type 属性 （Feature 对象）280
Type 属性 （Field 对象）304
Type 属性 （Layer 对象）362
Type 属性 （LayerInfo 对象）377
Type 属性 （Theme 对象）485
TypeEx 属性 （Field 对象）304
填充模式
定义 523
填充样式
远程空间表 175
添加
地图控件 55
将地图添加到 Visual Basic 表单 56–57
添加数据
数据绑定 86–87

统一资源定位符 (Uniform Resource Locator, URL)
定义 523
投影
定义 523
在 Geoset Manager 中设置 143, 671–672
图层
定义 523
集合中的绘图次序 75
生成标签 78–80
说明 66
缩放图层 76–78
用户绘制图层 82
在 Geodictionary 中注册 134, 531
“图层控制”对话框
调用 70
显示选项 71
图层控制
Geoset Manager 140–143, 669–671
图层说明 66
图像地图
定义 523
图形交换格式文件
导出格式 218
图元
类型 49

U

UCase$( ) 函数 618
Units 方法 （Affine Transform 对象）235
Units 属性 （CoordSys 对象）256
Unload 方法 （BitmapSymbol 集合）248
Update 方法 （Feature 对象）280
UpdateFeature 方法 （Layer 对象）363
UserName 属性 （Geoset 对象）322

V

Val( ) 函数 618
Value 方法 （DataSet 对象）262
Value 属性 （DataSet 对象）262
Value 属性 （IndividualValueCategories 集合）327
Value 属性 （RowValue 对象）443
Value 属性 （Variable 对象）503
ValuePerDot 属性 （ThemeProperties 对象）493
Variable 对象和 Variables 集合 502
Visible 属性 （Label 对象）333
Visible 属性 （LabelProperties 对象）339
Visible 属性 （Layer 对象）364
Visible 属性 （Legend 对象）384
Visible 属性 （LegendText 对象）387
Visible 属性 （Theme 对象）485
Visible 属性 （Title 对象）501
Visual Basic
4.0 版示例 57–59
6.0 版示例 59–60
将应用程序升级到 MapX 4.0 56
升级应用程序 55
添加地图 56–57
添加地图控件 55
Visual C++
MapX 属性和方法 191
MapX 异常错误 202–203
包括 MapX.cpp 文件 192
创建 MapX 控件 192
创建菜单项 194–195
创建地图 203–205
定制工具 197–198
内置帮助程序对话框 201–202
升级 MapX 应用程序 190
升级应用程序 56
示例应用程序 190
使用 C++ 的 MapX 事件 195
数据绑定 198–200
添加地图控件 55
添加快捷菜单 200
Visual C++ 中的 MapX 事件 195

W

WaitCursorEnabled 属性 （Map 对象）418
Web 服务器
定义 519
Web 浏览器
定义 520
Web 站点
定义 520
主页 524
Weekday( ) 函数 619
Version 属性 （Map 对象）418
Width 属性 （Legend 对象）385
Width 属性 （Rectangle 对象）438
Width 属性 （ThemeProperties 对象）493
Windows 元文件
导出格式 218
WMF 文件
导出格式 218
WriteToFile 方法 （State 对象）462
WWW 链接
定义 520
网格
定义 523

纬度
定义 521
纬线
定义 523
位图
定义 523
位图文件
导出格式 218
文本对象 80

X

X 方法 （Points 集合）432
X 属性 （Graphic 对象）325
X 属性 （Point 对象）430
X 属性 （Title 对象）501
XMax 属性 （Rectangle 对象）438
XMin 属性 （Rectangle 对象）438
XML
MapX XML 数据集驱动程序 99
miDatasetXML 99
X-Y 坐标
显示为点 94–95
在不同的 CoordSys 中指定 179–180
系统要求
MapX 52
Oracle8i 159
显示
标签 79
光栅图像 80
数据为点 92–95
向导
定义 524
项目链接
定义 524
像素
定义 524
小数度数
定义 524
修改
Features 集合 107
现有地图图元 110
主题 122–123
选择
使用 Visual Basic 6 中的查询 60
在 Visual Basic 6 中的距离 60
“选择投影”对话框
在运行时显示 180

Y

Y 方法 （Points 集合）433
Y 属性 （Graphic 对象）325
Y 属性 （Point 对象）430
Y 属性 （Title 对象）501
Year( ) 函数 620
YMax 属性 （Rectangle 对象）438
YMin 属性 （Rectangle 对象）439
样式
远程空间表 174
移除
标签 79
以制表符分隔的数据
数据源类型 89, 98
用户绘制图层 82
优化边界
图元搜索 126
邮政编码数据
显示为点 95
远程空间表
变成可以进行地图绘制 172–173
MapInfo 地图目录 168–173
指定样式 174–175
远程空间数据
访问 158
访问属性数据 165–166
缓存 166–168
LayerInfo 参数 162
ODBC 连接字符串 162–164
通过 X/Y 列进行地图绘制 159
性能问题 166
运算符 575

Z

Zoom 属性 （Map 对象）419
ZoomLayer 属性 （Layer 对象）364
ZoomMax 属性 （Layer 对象）364
ZoomMin 属性 （Layer 对象）364
ZoomTo 方法 （Map 对象）419
站点
说明 524
支持 Oracle8i 159–162
支持 Oracle8i v8.15 和 v8.16 159
支持格式
光栅图像 80
指定样式
标签 79
直线样式
远程空间表 174
质心
定义 524

逐层细化表
元数据关键字 152
逐层细化工具
创建 152
逐层细化图层
重置 154
术语和概念 149–150
说明 148–149
限制和要求 154
相关常数 154
相关方法和属性 154
逐层细化应用程序
过程小结 150
汇总工具 154
逐层细化表的要求 151–152
逐层细化工具 152
组件表的要求 150
注释 80
主题
插入 Visual Basic 示例 60
创建 112–114
类型常数 113
修改 122–123
主题地图
二元 122
类型 114–121
使用数据 112
主题图例 123
主页
定义 524
柱状图地图
主题地图类型 120–121
自定义工具
逐层细化工具 152
自动标记
图层设置 74
字符串比较 578
字符串运算符 578
字符串子句 580
字体 226
字体属性 110
子午线
定义 525
组件表
逐层细化图层 150
坐标系 525
坐标系统
CoordSys 对象 178
参考材料 187
解释类型数字 184
MAPINFOW.PRJ 文件 183–184
为地图指定 178–179
坐标系统参数 181